遗传算法在智能组卷系统中的设计与实现

魏德志 (集美大学诚毅学院, 福建 厦门 361021; 中国人民大学信息学院, 北京 100872)
林丽娜, 吴 旭 (集美大学诚毅学院, 福建 厦门 361021; 厦门大学软件学院, 福建 厦门 361021)

[摘要] 为了能更好地解决组卷质量和组卷速度之间的矛盾, 提出了一种基于分段整数编码的遗传算法。该算法在保证组卷预期效果的前提下, 不仅搜索速度快, 而且能够避免遗传算法中经常出现的"早熟现象", 具有很好的收敛性和实用性。实践结果表明, 该方法可以有效地解决智能组卷中的约束优化问题。

[关键词] 题库; 组卷; 遗传算法; 约束优化
[中图分类号] TP392 [文献标识码] A [文章编号] 1673-1409(2008)01-N247-03

自动组卷系统是计算机辅助教学系统(CAI)的重要组成部分, 其主要难题是如何保证生成的试卷能最大程度的满足用户的不同需要, 并具有随机性、科学性、合理性。目前的智能组卷算法主要有盲目随机选取法[1~3]、回溯法[4]、遗传算法[5~6]3种。文献[7]采用传统的简单遗传算法(SGA)来实现试题库的自动组卷, 取得了较好的实际效果。笔者对自动组卷数学模型进行分析, 并提出了一种改进的遗传算法。

1 自动组卷的数学模型

组卷目标包括: 试卷总分、章节分值、题型分值、试卷难度、考试时间、知识点满足、能力层次满足、试卷区分度、试卷形式。组卷的目标状态矩阵 $\boldsymbol{A}$:

$$\boldsymbol{A}=\begin{vmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,c} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,c} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,c} \end{vmatrix} \quad (1)$$

组卷中生成一份试卷, 就是得到一个 $n\times c$ 的矩阵。其中, n 是试卷中试题总数。每道试题的 c 个属性决定矩阵一行元素的值。组卷过程问题求解为: 根据用户输入或缺省设置的组卷目标要求, 在矩阵中寻找满足组卷目标要求的行组合, 即搜索确定矩阵 $\boldsymbol{A}$ 中全部行状态的多变量组合优化$(K_1, K_2, \cdots, K_n)$。这里 K 为第 i 行的状态变量(即第 i 道题), 若 $K_i=1$ 表示第 i 行被选中, 而 $K_i=0$ 表示第 i 行没有被选中。每个组卷目标要求都对应着一个约束条件, 则全部 c 项组卷目标要求分别对应着如下约束条件: ①试卷总分: $\sum_{i=1}^{n} k_i a_{i,1}=P$, P 是试卷总分、$a_{i,1}$ 为第 i 道试题分值; ②章节分数: $\sum_{i=1}^{n} k_i b_{i,s}=P_s$, 第 s 章节试题分数为 P_s, 当试题章节编号 $a_{i,2}=s$ 时有 $b_{i,1}=a_{i,1}$ 否则为零; ③题型分数: $\sum_{i=1}^{n} k_i g_{i,m}=w_m$, w_m 为第 m 种题型试题的分数。当试题题型编号 $a_{i,3}$ 属于第 m 种题型时有 $g_{i,m}=a_{i,1}$, 否则 $g_{i,m}$ 等于零。试题共分成7种题型(即选择填空、问答题、改错题、证明题、分析判断题、计算题、综合题); ④试卷难度: $\sum_{i=1}^{n} k_i a_{i,1} a_{i,4}=PD$, $a_{i,4}$ 为第 i 道试题的难度系数, D 为指定试卷的难度值; ⑤考试时间: $\sum_{i=1}^{n} k_i a_{i,5}=T$, T 为用户要求的考试时间, $a_{i,5}$ 为第 i 道题的预计答题时间; ⑥知识点满足: $\sum_{i=1}^{n} k_i u_{i,n}=Z_n$, 第 n 个知识点试题分值 Z_n。当试题知识点属性编号 $a_{i,6}$ 等于第 n 个知识点时有 $u_{i,n}=a_{i,1}$, 否则为零; ⑦能力层次满足: 教学内容掌握的层次要求分为一般了解、熟悉理解、熟练掌握和灵活运用4个层次, $a_{i,7}$ 为第 i 道试题的教学内容掌握层次要求编码。显然, 所

[收稿日期] 2007-12-05
[作者简介] 魏德志(1982-), 男, 2003年大学毕业, 助教, 硕士生, 现主要从事遗传算法、数据挖掘方面的研究工作。

选择的试题应满足指定的能力层次要求; ⑧试卷区分度: $\sum_{i=1}^{n} k_i a_{i,1}\ a_{i,8} = F$, $a_{i,8}$ 为第 i 道题的区分度; ⑨试卷形式: $a_{i,9} \subset$ (用户指定试卷形式), $a_{i,9}$ 为每道题的形式属性代码。分为客观形式和主观形式两种, 前者支持客观标准做答, 后者支持过程主观输入做答(计算题需人工阅卷)。

2 改进的遗传算法设计

改进的遗传算法实现过程如下:

1) *编码方法的确定* 常用组卷遗传算法中, 随着题库试题量的增多, 染色体长度不断增长, 组卷计算的时间也加长。针对此弊端, 提出一种新的编码方法——分段整数编码。

染色体按题型分段编码, 各题型的代码段相对独立。设题库共有 H 道 m 种类型的试题, 则有 $\sum_{i=1}^{m} H_i = H$ 道试题, H 为试题库中每一道试题按顺序赋以惟一代号 (不同题型试题的代号可相同), 代号由一个 t 位的实数串构成, t 由题库中该题型试题量的大小确定, 各题型的 t 值可不同。若某题型试题共有 800 道, 则 t 取 3; 若有 8000 道, 则可取 4。编码时需保证同一题型中试题的代号串的位数相同。染色体代码由各题型段选中试题的代号排列组成, 如: $(J_{1,1}, J_{1,2}, \cdots, J_{1,h1}), \cdots, (J_{m,1}, J_{m,2}, \cdots, J_{m,hm})$。其中 h $(i = 1, 2, \cdots, m)$ 为第 i 种题型选中的试题数, 染色体代码的长度由试卷所需试题数确定。

2) *适应度函数设计* 构造染色体满足组卷目标的误差函数为:

$$E = \sum_{i=1}^{9} a_k 10^{e_k} \tag{2}$$

其中, $e_k = 0 \sim 1$ 为反映染色体满足第 k 项组卷要求程度的归一化相对误差; $0 < a_k < 1$ 为相应的误差加权系数, 且 $\sum_{k=1}^{9} a_k = 1$。E 的最小值为 1, 此时 e_k 均为 0, E 的最大理论值为无穷大。染色体的适应度函数[8]定义为:

$$Fit = E^{-1} \tag{3}$$

由式 (3) 知: 染色体适应度为 0~1, 最大值为 1。算法收敛时允许的最大综合误差参数均为 5%, 此时有 $e \leqslant 0.05$, $E \leqslant 1.12$, $Fit \geqslant 0.891$。

3) *初始种群生成* 根据组卷目标中各种题型的分值要求, 在初始种群产生时, 对于每个独立代码段采用随机产生初始群体的办法, 这样就使得整个初始种群基本满足了题型分布与总分值的要求。因题型分布、总分值已基本满足, 相当于减少了总约束项, 这对于加快进化速度有利。在生成初始群体的个体时, 每产生 1 个满足条件的随机数时, 都要把题库中对应试题的选题标记置 1, 同时把题库中具有相同知识点的试题的选题标记置 1, 这样就可以避免相同知识点试题在同一试卷生成时的重复出现, 从而优化了初始群体。

4) *交叉算子设计* 采用分段单点交叉策略: 对随机选择的两个染色体, 在每个分编码段内同时采用随机产生交叉位置的单点交叉, 且判断交叉后该段内的所有试题代号不一样。若段内试题代号一样, 则重新选择交叉点 (至多两次选择机会)。交叉概率采用改进式自适应方式产生, 改进式自适应交叉概率由下式确定:

$$p_c = \begin{cases} p_{c1} - \dfrac{(p_{c1} - p_{c2})(F_c - F_{avg})}{F_{max} - F_{avr}} & F_c \geqslant F_{avr} \\ p_{c1} & F_c \leqslant F_{avr} \end{cases} \tag{4}$$

式中, F_c 为要交叉的两个串中较大的适应值; F_{max}、F_{avr} 为上代群体中个体的最大适应值及群体的平均适应值; $p_{c1} = 0.9$, $P_{c2} = 0.6$。

5) *变异算子设计* 变异采用段内单位置替换式变异, 即在个体的各个不同题型段内随机选取一位进行替换, 且规定该位被替换后的题号不能与该个体的其他位题号重复。变异概率采用改进式自适应变异概率, 其具体表达式为:

$$p_m = \begin{cases} p_{m1} - \dfrac{(p_{m1} - p_{m2})(F_m - F_{avg})}{F_{max} - F_{avr}} & F_m \geqslant F_{avr} \\ p_{m1} & F_m \leqslant F_{avr} \end{cases} \tag{5}$$

式中, F_m 为要变异个体的适应值; $p_{m1} = 0.1$, $P_{m2} = 0.01$。

6) 选择算子设计　运用 hereboy 算法的思想, 搜索概率的计算公式为:

$$\text{Search Probability} = p * b \tag{6}$$

$$b = (\text{MaxScore} - \text{MaxCurrentScore}) / \text{MaxScore} \tag{7}$$

其中, p 是用户定义的最大搜索概率, 为定值, 由用户给定, 经验值取 0.01~ 0.05, 笔者取 0.02; MaxScore 为最大适应度值, 即算法收敛的适应度值; MaxCurrentScore 为当前代的适应度值。

7) 搜索结束条件　搜索结束条件的判断: 一是最大染色体适应度达到 $Fit = 0.95 \sim 1$; 二是种群中最大个体适应度 $Fit > 0.95$, 且在连续 5 代内最大适应度值改善小于 0.01 时; 三是已进化到规定的最大代数(现设为 1000) 时, 终止进化, 若此时最大个体适应度值 $Fit < 0.891$, 则认为本次问题求解搜索失败。

3　实验分析

为验证上述遗传算法的可行性和有效性, 针对《计算机技术基础》二级题库, 按照上述思想用 asp 语言编制了程序, 进行组卷试验。本次实验题库共有 1200 道题, 包括 8 个二级考查点、三类题型、4 种难度。组卷要求为: 整卷难度系数为 0.4, 允许误差为 ±0.05; 3 种题型, 各题型所占分数比例为 3∶ 4∶ 3; 8 个二级考查点所占分数比例为 2∶ 2∶ 2∶ 5∶ 5∶ 1∶ 2∶ 1。本算法同简单遗传算法的仿真实验结果比较如表 1 所示。

表 1　传统遗传算法与改进遗传算法的比较

实验序号	传统遗传算法		改进遗传算法	
	进化代数	所用时间/s	进化代数	所用时间/s
1	234	75	52	30
2	217	64	53	31
3	280	86	60	35
4	275	80	63	34
5	263	75	58	33

通过表 1 的实验数据, 分别将两种算法的数据制作成折线图进行比较, 可以更清楚的看出两种算法的区别。

由表 1, 图 1 可见, 改进的遗传算法在收敛速度上有了明显提高, 也比较稳定, 能有效地解决试题库中的智能组卷问题, 与传统遗传算法相比, 能较快地找到满足条件的解。

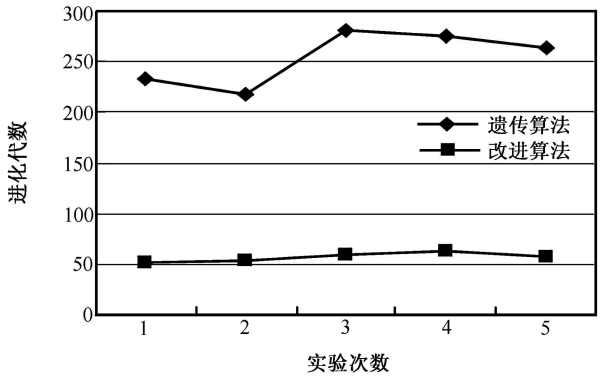

图 1　两种算法进化代数比较

[参考文献]

[1] 杨朝群. 谈试卷编制与题库建设 [J]. 云南高教研究, 1985, (1): 75~ 77.
[2] 戴亚飞. 计算机自动组卷算法分析 [J]. 小型微型计算机系统, 1995, (9): 128~ 130.
[3] 李晔. 计算机基础与语言课试卷自动生成系统 [J]. 陕西教育学院学报, 2000, 16 (2): 86~ 87.
[4] 胡维华等. 多目标选题策略研究与应用 [J]. 杭州电子工业学院学报, 1999, (2): 72~ 74.
[5] Goldberg D E Genetic Algorithms in Search, Optimization and Machine Learning [J]. Reading, MA: Addison-Wesley, 1989.
[6] 陈国良. 遗传算法及其应用 [M]. 北京: 人民邮电出版社, 1999.
[7] 全惠云. 基于遗传算法的试题库智能组卷系统研究 [J]. 武汉大学学报, 1999, (5): 156~ 157.
[8] 于洋, 查建中, 唐晓君. 基于学习的遗传算法及其在布局中的应用 [J]. 计算机学报, 2001, 24 (12): 1242~ 1249.

[编辑]　洪云飞