

自适应遗传算法在智能组卷中的应用

黄宝玲

(柳州职业技术学院信息工程系, 广西 柳州 545006)

摘要: 传统遗传算法在组卷过程中存在收敛速度慢、迭代次数多, 以及采用固定遗传概率在遗传操作中极易破坏高适应性个体等问题。为此, 提出一种用于提高组卷效率的自适应遗传算法。根据组卷参数的不同, 采用有侧重的不完全随机搜索策略, 针对低适应性个体借助交叉、变异算子进行快速淘汰, 同时在迭代过程中增加最优个体保存机制, 以较小的运算代价获得较高的组卷效率。实验结果表明, 该算法在迭代次数、运行时间和组卷准确性方面均优于传统算法。

关键词: 自适应; 遗传算法; 智能组卷; 遗传操作; 试卷定制

Application of Adaptive Genetic Algorithm in Intelligent Test Paper Composition

HUANG Bao-ling

(Department of Information Engineering, Liuzhou Vocational & Technological College, Liuzhou 545006, China)

[Abstract] In order to overcome the slow convergence rate, too much number of iterations and the fixed heredity probability is extremely easy in the heredity operation to destroy questions and so on high compatible individual. According to the differences of the parameters of the generating papers, this paper uses incomplete random searching strategy. For the low adaptability of individual, it uses crossover, mutation operators for rapid elimination. As the meanwhile, it uses an iterative process to increase the best individual saving mechanisms and to decrease the cost of the high computing efficiency of the generating paper. Experimental result indicates that compared with traditional algorithm, the adaptive genetic algorithm has the characteristics such as fast convergence rate, high speed of running times and more accuracy of generating papers.

[Key words] adaptive; Genetic Algorithm(GA); intelligent test paper composition; genetic manipulation; test paper customization

DOI: 10.3969/j.issn.1000-3428.2011.14.053

1 概述

智能组卷就是根据用户提出的组卷要求, 系统自动地从题库中选择符合的试题形成试卷的过程。智能组卷的效率与质量取决于组卷算法的设计。

遗传算法是一种借鉴生物界自然选择和自然遗传机制的随机搜索算法。遗传算法的群体搜索性使其突破了一般启发式算法邻域搜索的限制, 可实现整个解空间上的分布式信息采集和搜索。近年来, 许多学者在遗传算法应用于组卷方面进行了有益的尝试。传统遗传算法在收敛速度和求全局最优解之间存在矛盾, 具有容易产生早熟现象、局部寻优能力较差等缺点^[1-2]。应用研究表明, 将遗传算法与问题特有的知识集成到一起所构成的混合遗传算法, 求解性能更好^[3]。

为解决智能组卷中多约束优化问题, 改善传统算法的不足, 本文提出一种采用分段实数编码、预设定制参数权重及自适应调整交叉、变异算子的改进遗传算法。

2 试卷属性与参数定义

组卷往往涉及到多个约束条件需同时满足的问题, 但约束条件过多也会增加组卷难度。鉴于多约束条件的重要性不同, 且它们之间存在内在联系, 有必要选择一些核心属性作为组卷的约束条件。

2.1 属性定义

组卷的数学模型: 在组卷中决定一道试题, 就要决定 n 项指标。本系统确定了试题的 6 个属性(约束): a_1 知识点, a_2 难度, a_3 区分度, a_4 分值, a_5 答题用时, a_6 题型等, a_i 相当于第 i 项指标。对于一套有 m 道试题的试卷, 实际上就是

决定一个 $m \times 6$ 矩阵。试题的指标属性矩阵如下:

$$S = \begin{Bmatrix} a_{11}a_{12}L & a_{16} \\ a_{21}a_{22}L & a_{26} \\ M & M \\ a_{m1}a_{m2}L & a_{m6} \end{Bmatrix}$$

其中各指标预设值为: (1)每门课程分 6 个知识点, 由有经验的教师划分, 取值 1~6; (2)难度分为 4 个等级: 较易, 中等难度, 较难, 难度较大, 分别赋值 0.3、0.5、0.7、1.0 以示区分; (3)区分度分 3 个等级: 低, 中, 高, 分别赋值 0.3、0.5、0.8 以示区分; (4)分值分 5 个档次: 2, 3, 4, 15, 20; (5)答题用时分 4 个等级: 2, 4, 10, 20; (6)题型分为单选题、填空题、多选题、改错题、综合题等, 分别赋值 1~5 以示区分。各题型分在不同段进行编码, 每种题型的编码间相互独立, 将多段编码合成即得一份试卷编码。

所有试题除试题内容、答案等字段外, 还要增加如上所述的 6 个属性字段保存数据表中。试题在数据表中的唯一标识就是试题记录号, 从 0 开始顺序编码。这样, 在抽到该试题时, 就容易提取该试题的各项属性值, 从而判别该试题是否符合组卷需求。当通过算法组卷获得一份试卷后, 可通过以下系列算式来计算该份试卷的各约束值^[4]:

基金项目: 教育部高职高专计算机类专业立项课题基金资助项目(jzw59010826)

作者简介: 黄宝玲(1976—), 女, 讲师、硕士, 主研方向: 智能化信息管理

收稿日期: 2011-01-12 **E-mail:** lzyhbl@163.com

批注 [U1]: 矩阵应该如右图。

$$(1) \text{ 试卷总分: } ZF = \sum_{i=1}^m a_{i4}$$

$$(2) \text{ 答题总时间: } ZS = \sum_{i=1}^m a_{i5}$$

$$(3) \text{ 试卷难度: } NDXS = \sum_{i=1}^m a_{i2} a_{i4} / \sum_{i=1}^m a_{i4}$$

$$(4) \text{ 试卷区分度: } QFD = \sum_{i=1}^m a_{i3} a_{i4} / \sum_{i=1}^m a_{i4}$$

$$(5) \text{ 第 } k \text{ 知识点的题分: } TF_k = \sum_{i=1}^m c_{ik} a_{i4}, \text{ 其中, 当 } a_{ii}=k \text{ 时,}$$

$C_{ii}=1$; 当 $a_{ii} \neq k$ 时, $C_{ii}=0$ 。

2.2 试卷参数

在定制试卷时, 用户需提供组卷参数, 如表 1 所示。算法将参数保存在数组或变量中, 组卷目标是生成一份最符合需求的试卷。

表 1 定制试卷参数说明

序号	数组或变量名	说明
1	Zsd[6]	6 个知识点占试卷百分比, 和为 100
2	T[5]	5 种题型的题量
3	W[5]	5 个约束指标所占权重, 和为 1
4	Znd	试卷总难度, 0~1 间的数, 一般取 1 位小数
5	Zqf	试卷总区分度, 0~1 间的数, 一般取 1 位小数
6	Zf	试卷总分
7	Zs	作答试卷总用时

3 遗传算法设计

3.1 染色体编码

遗传算法的编码就是用一种数字排列来表示问题的解的方法。传统的二进制编码会随着试题量的增多, 使染色体长度不断增长、计算复杂, 不利于遗传操作。为此, 本文将各种题型分开以实数来编码, 每种题型的编码作为一段, 各题型编码组合在一起形成遗传个体的编码:

(1) 18.9.52.84.63.22.11.2.27.43.34.18.6.25.47.17.4.26.16.4.
单选题 填空题 多选题 改错题 综合题

3.2 试题的适应度

确定一道试题要比对定制试卷的输入参数, 由于各参数间的联系很难确定, 因此无法给每个试题一个确定的、量化的适应度值。本文的试题适应度是先随机抽题, 再分别提取试题知识点属性与定制试卷的知识点参数比较, 不符合的就淘汰, 再随机抽取。使种群中的个体首先满足知识点目标要求。试卷其他约束的满足及最优解的判定要通过组卷的目标函数来衡量。

3.3 组卷的目标函数

试卷在定制时用户给出了期望的试卷难度、区分度、答题时间、总分等参数, 假设分别用 D 、 Q 、 T 、 F 来表示, 生成的试卷个体各参数如前所述分别用 $NDXS$ 、 QFD 、 ZS 、 ZF 表示, 则要分别计算下式的值:

(1) $f_1 = |D - NDXS|$, 表示用户给出的试卷难度值 D 与试卷整体平均难度 $NDXS$ 之差的绝对值。

(2) $f_2 = |Q - QFD|$, 表示试卷个体的区分度 QFD 和用户所期望的区分度 Q 之差的绝对值。

(3) $f_3 = |T - ZS|$, 表示试卷个体的整卷时间 ZS 和用户所确定的答卷时间 T 之差的绝对值。

(4) $f_4 = |Z - ZF|$, 表示试卷个体的总分 ZF 和用户所确定的期望总分 Z 之差的绝对值。

其中, f_i 越小, 表示此个体的相应属性越接近期望值, 越满足用户要求。另外, 根据实际组卷经验, 对不同的约束

条件可给定不同的允许误差(0.01~0.05), 在容差范围内的误差可忽略, 从而加快搜索速度。组卷的目标函数为:

$$F = \omega_1 f_1 + \omega_2 f_2 + \omega_3 f_3 + \omega_4 f_4$$

其中, $\omega_1 \sim \omega_4$ 分别代表不同期望值的权重, 且 $\omega_1 + \omega_2 + \omega_3 + \omega_4 = 1$ 。在实际的应用中, 可根据 $f_1 \sim f_4$ 参数的重要程度给出对应的权值。例如组卷用户认为试卷的知识点分布须符合自己所给的数据, 偏差应最小, 那么知识点约束的权重可设大一些, 其他约束权重相应就要小一些。

3.4 自适应遗传算子

遗传算法的交叉概率 P_c 和变异概率 P_m 是影响遗传算法行为和性能的关键所在, 直接影响算法的收敛性。 P_c 和 P_m 越大, 算法产生新个体的能力就越强, 个体间的适应性波动就会比较大, 可能会使高适应度个体被破坏, 算法因而付出较大的运算代价; P_c 和 P_m 越小, 算法使个体趋于收敛的能力越强, 个体的平均适应度比较平稳, 也可能减少个体的多样性, 出现局部收敛。在本算法的运行过程中, 除预设 P_c 和 P_m 的值外, 还给 P_c 和 P_m 的自适应调整提供了一个机制, 使其随个体适应度的增加而减少, 随个体适应度的减少而增加, 机制的具体实现如下所述。

3.4.1 选择

由于选择算子是在满足知识点前提下随机抽题, 并在遗传操作过程中采用最优个体保存策略。父代群体中适应度最高的一个个体并不参与交叉和变异操作, 而是被直接保留到下一代, 替换掉本代群体经过交叉、变异等操作后所产生的适应度最低的一个个体。

3.4.2 交叉

采用个体排序后单点交叉算子。交叉的实现过程: 将当前群体中的各个体, 分各题型块以题号(试题在数据表中的序号)为关键字按从小到大排序, 并选择 2 条染色体, 按概率 P_c , 随机地对 2 条染色体的随机基因点进行交叉。设个体的基因长度为 N , 则产生一个 0~ $N-1$ 之间的随机数 T 。将 2 个个体在第 T 位基因位置分为左右 2 段, 并将之从第 T 基因位开始的右半段互换并重新连接得到 2 个新个体。这样交叉得到的新段可能是非法的, 因为存在题号重复的可能, 但最多只会在其中一个个体的交叉点所在题型段内出现重复题, 所以需要对此 2 个新个体进行调整^[5]。

例如, 以下 2 个编码长度为 9 的个体 a 、 b 进行交叉, 交叉位置为下划线位置:

$a[2,4,8 \quad 2,3,6 \quad 3,5,7]$

$b[2,3,4 \quad 1,2,3 \quad 3,8,9]$

交叉之后形成 2 个新个体 a' 、 b' 如下:

$a'[2,4,8 \quad 2,2,3 \quad 3,8,9]$

$b'[2,3,4 \quad 1,3,6 \quad 3,5,7]$

显然, 在这 2 个新个体中, 第一个个体 a' 出现了重复题号“2”, 而另一个新个体不可能出现重复题号, 因为题号是按从小到大顺序排列的。

确定交叉点后, 比较交叉点所在题型段的第一个题号, 题号小的个体交叉后新个体一定合法, 题号大的交叉后新个体有可能出现题号重复或次序混乱, 比如本例新个体 a' 。可能出现题号重复情况的只在交叉点所在题型段, 重新排序后, 在 0~ $L-1$ 之间(L 为题库中该题型的数量)取随机数, 该数要不同于已存在题号, 用其替换重复题, 即形成合法新个体。再计算子代的目标函数, 与父代比较, 即从这 4 个染色体中选择 2 个目标函数值最小的染色体保存下来, 从而保持群体规模 M 不变。

3.4.3 变异

本算法的变异算子将分2种情况:

(1)当交叉算子后某个个体出现重复题号,对其中一个重复基因执行变异操作,以去除重复题号。

(2)采用个体基因的单个变异,即只对某题型段上的某个基因进行变异,对某个染色体,随机生成一个 $[0, 1]$ 范围内的实数 r ,若 $r \leq P_m$,则对该染色体进行变异,否则不进行变异。

变异的操作如下:在 $[1, 5]$ 范围内随机生成一个题型段号 i ,设该段的段长为 L ,则在 $[0, L-1]$ 范围内随机生成一个变异位置 p ,以不完全随机的选择原则从题库与其对应的题型表中选择一个变异基因,直到产生不同于已有题号的随机数,将其插入排序队列内,形成合法新个体。

3.5 迭代终止条件

迭代终止条件分2种情况:

(1)在最理想的情况下是所组试卷的各项指标完全符合用户输入参数要求,这时迭代自然终止,算法找到了最优解。

(2)由于用户需求的多样性,组卷目标函数的约束条件较多,有时要精确满足所有约束条件会使迭代无休止地进行下去,考虑到本文采取了最优保存策略,历代中的最优个体一定非常接近于目标期望值,所以,规定一个最大迭代次数,以最后一代的最优个体为所求解,停止迭代。

4 算法实现

本文改进遗传算法的具体实现是通过 Visual Studio 2005 语言平台实现。

4.1 试卷定制及算法控制参数

试卷定制参数如表2所示。

表2 本次组卷实验所用参数

序号	数组或变量名	取值	说明
1	Zsd[6]	{20, 0, 30, 0, 50, 0}	第1、3、5知识试题分别占20%、30%、50%
2	TI[5]	{10, 5, 5, 5, 2}	单选题10道、填空题5道、多选题5道、改错题5道、综合题2道
3	WI[5]	{0, 0.5, 0.3, 0.1, 0.1}	各约束权重值:难度0.5,区分度0.3,分值0.1,答题用时0.1
4	Znd	0.6	希望试卷总难度控制在0.6左右
5	Zqf	0.5	希望试卷总区分度控制在0.5左右
6	Zf	100	试卷总分为100
7	Zs	120	希望试卷答题用时在120 min左右

算法控制参数如下: $M=10$,每迭代一次的群体大小,即群体中所含个体的数量; $Gen=100$,遗传算法的终止进化代数; $P_c=0.5$,交叉概率; $P_m=0.08$,变异概率。

4.2 最优解搜索

本文算法通过 Visual C# 2005 程序设计语言实现,实现流程如下:

(1)在试卷定制界面中,提供组卷参数。

(2)用5个2维数组存放数据库5种题型试题的知识点、难度、区分度、分值、答题用时的数据值。

(3)设置进化计数器 $Gen \leftarrow 0$ 。

(4)选择运算:得到初始群体 $P(Gen)$ 。对每个个体分题型按从小到大的顺序排序,将各题型段内有重复题号的基因去除,重新用随机函数抽题,直到种群中所有个体的各题型段

内均无重复题号为止。

(5)交叉运算。

(6)变异运算。

(7)将得到的新一代种群的各个体,根据用户组卷定制界面提出的各约束指标的权重值来计算每个个体的目标函数值,并将之升序排列,执行最优个体保存策略,即记忆第一个(即目标函数值最小的,与组卷要求误差最小的)个体,该个体是遗传操作运行到迄今为止的最满意解。

(8)进行终止条件判断:1)解的目标函数与用户需求吻合;2)遗传操作的迭代已经达到一定次数。若不满足终止条件,则 $Gen \leftarrow Gen+1$,转到第(4)步;若满足终止条件,则输出迄今为止的最高适应性个体,算法结束。

实验结果如图1所示,每行为一套试卷的试题号(各题型分布见3.1节的试卷染色体编码),共10行,即本算法遗传种群数为10,然后依据该实数从各题型数据表中抽题,生成相应格式的试卷即可。

7	15	53	60	63	72	73	78	80	85	3	12	14	19	37	6	18	21	41	44	2	15	28	38	42	12	15
3	31	44	62	63	64	72	63	90	91	0	15	25	40	43	16	19	31	38	39	16	29	37	44	46	12	13
4	27	31	54	58	64	69	75	83	95	12	14	17	21	29	4	15	21	39	44	6	15	17	29	37	4	8
53	59	62	63	68	73	75	79	85	95	3	12	19	31	46	3	25	30	37	40	15	19	29	37	40	8	19
13	14	17	22	65	69	72	74	80	95	17	19	40	42	44	0	13	17	29	44	3	4	12	17	46	4	12
0	2	4	12	15	17	32	44	64	85	13	17	21	25	44	21	27	30	40	43	5	19	27	39	40	4	12
4	7	9	12	13	54	59	61	62	64	14	17	40	43	44	16	18	25	29	40	6	15	16	21	40	11	15
0	2	13	14	56	65	69	75	79	95	0	18	40	42	43	28	30	37	43	46	18	19	27	30	46	10	15
4	14	27	54	59	62	66	68	79	83	0	5	13	14	25	6	15	17	18	30	19	21	29	41	46	8	12
27	41	60	66	68	74	75	79	90	91	4	12	29	30	43	12	16	31	43	46	19	21	25	28	44	8	18

图1 本文算法的实验结果

本次实验结果是迭代100次得到的最后种群,其中,第8行个体为最优解,该个体在知识点和各题型题量约束上100%符合需求,试卷总分的符合度为98%,试卷答题总用时的符合度为99%,总难度值为0.56667,符合度97%,总区分度值为0.5074,符合度为99.3%。算法的结果很好地满足了组卷的要求,且效率和成功率较高。由此可见,将本文改进的遗传算法用于组卷是可行的。

5 结束语

传统遗传算法的运行机制并不都有益于组卷问题的求解,有些甚至会阻碍优良个体的产生。本文算法通过自适应调整遗传算子,使整个种群保持在最有可能获得成功状态,加速了算法向全局最优值的逼近速度,提高了组卷效率。本文提出的算法可供开发组卷系统的学者参考。下一步工作将将在自主研发的网络教学测试平台中运用本文提出的组卷策略,同时将对遗传算法在其他方面的应用做进一步的研究。

参考文献

- [1] 江建. 精英自适应混合遗传算法及其实现[J]. 计算机工程与应用, 2009, 45(27): 34-35.
- [2] 初文科. 基于遗传算法的试题库设计与组卷方法研究[D]. 济南: 山东师范大学, 2008.
- [3] 周文举. 一种基于知识点的遗传算法组卷的改进应用[J]. 山东师范大学学报: 自然科学版, 2006, 21(3): 39-42.
- [4] 刘贝贝, 肖明, 马晓敏. 基于推理的组卷数学建模及其应用[J]. 计算机工程, 2010, 36(4): 71-72.
- [5] 范明虎, 孙斌. 通用试题库管理系统的设计与实现[J]. 计算机工程与设计, 2007, 28(9): 61-62.

编辑 金胡考