

如何出考题：分级回溯试探组卷算法

李川¹, 张少茹²

(1. 西安航空学院计算机学院, 西安 710077;

2. 西安交通大学医学部, 西安 710061)

摘要: 针对随机函数法、遗传算法、回溯试探法等当前常用组卷算法效率低、无法确保组卷质量等问题, 采用分级求解、早剪枝、瘦算法的思想, 提出一种分级回溯试探组卷算法。试验结果表明, 随着题库题目数量的增加, 分级回溯试探组卷算法在执行效率、组卷质量、试卷质量等方面均有显著提高。

关键词: 回溯试探; 分阶; 剪枝; 组卷

中图分类号: TP312

文献标识码: A

文章编号: 1674-4969(2019)01-0025-06

引言

测评是教育过程中考查学生知识能力水平的一种有效方式^[1], 随着教育信息化的发展, 传统的纸质试卷的考试模式暴露出了很多弊端^[2]。在线测评利用计算机网络简化了考试流程, 组卷、阅卷等很多工作可以使用计算机自动完成, 极大地减轻了教师的负担, 而且能够节约资源, 保证考试质量^[3]。

在线测评系统一般包括试题管理模块、组卷模块、考试管理模块等, 其中组卷模块属于技术难点, 组卷是按照给定的组卷约束条件从试题库中选取一定数量的试题组成若干套试卷的过程。组卷的关键是组卷策略(算法)的设计, 衡量组卷策略性能的依据是组卷质量、试卷质量、重复率及组卷效率等。

1 组卷算法研究

常见的组卷算法有随机抽选法、回溯试探法、

遗传算法等。

1) 随机抽取法

根据组卷筛选条件, 要求采用随机函数反复地在题库中选择一个符合要求并且没有抽取过的试题组卷, 直到组卷成功或无法满足组卷要求导致组卷过程停止^[4]。随机抽取法相对简单, 用户设计实现比较容易, 在现有的一些小型在线测评系统中应用较多^[5]。但由于随机选取法模式较为固定, 题库需要日益更新, 很容易被淘汰^[6], 并且组卷效率太低, 试题的覆盖率也没有办法得到保证。

2) 遗传算法

遗传算法从生物现象遗传规律演变而来, 经过了初始群体、演变、个体重组、优化下一代一系列过程, 通过重复该过程, 从而找到满足组卷要求的最优解^[7-8], 具体过程如图1所示。该算法组卷成功率高, 但算法实现困难。

收稿日期: 2018-08-08; 修回日期: 2019-01-14

基金项目: 国家自然科学基金《基于GIS的高校学生结核病防治管理体系及信息系统构建与应用研究》(71373203); 陕西省科技工业攻关项目《通用机载数据总线监控及测试技术研究》(2016GY-139); 西安航空学院校级科研基金项目《基于积差相关系数统计的OJ系统设计与研究》(2016JS3223)

作者简介: 李川(1980-), 男, 硕士, 副教授, 主要从事软件算法研究及大数据分析工作。E-mail: lch9964121@163.com

张少茹(1958-), 女, 博士, 教授, 博士生导师, 主要从事慢性病及传染性疾病防治管理工作。E-mail: shaoru@126.com

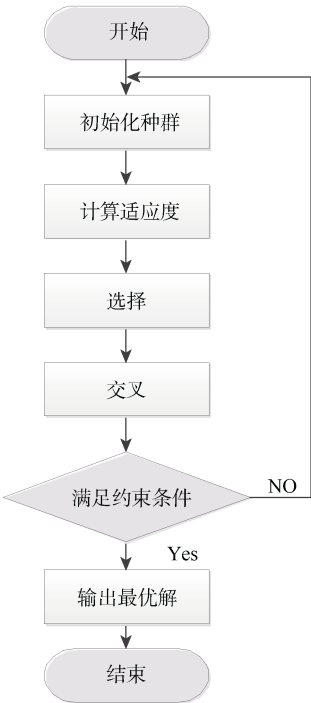


图 1 遗传算法流程图

3) 回溯试探法

回溯试探法是在随机抽取法的基础上保证了组卷成功率的一种算法^[9]。回溯试探法以随机函数抽取法为基础^[10],在选题过程中记录选题步骤,当判断所选题不能满足筛选条件且组卷过程未结束时,则回溯到上一个选题步骤重新抽题,通过反复地回溯试探抽题直到组卷成功^[11]。回溯试探法在组卷过程中终止了无效搜索,较随机函数选取法在性能方面有了明显提高。另外,回溯试探法相比遗传算法而言,算法的实现要相对简单很多。但回溯试探过程中需要记录每一步的状态,当组卷量较大时,需要记录大量信息,而反复回溯也会耗用大量时间,因此该算法组卷成功率高,但效率相对不高^[12]。

上述组卷算法各有优缺点,且有各自的适用情况。在题库规模大、约束条件多的情况下,随机抽取法效率低,而遗传算法在约束条件多时算法程序复杂,难以实现,因此,本文拟以回溯试探组卷法为基础,针对该情况下回溯法效率低的缺点,将回溯法进行改进,研究设计一种分级回

溯试探组卷算法模型^[13]。

2 基于回溯试探的分阶求解组卷算法设计

回溯试探算法是一种 DFT 选优搜索算法^[14],算法流程如图 2 所示。

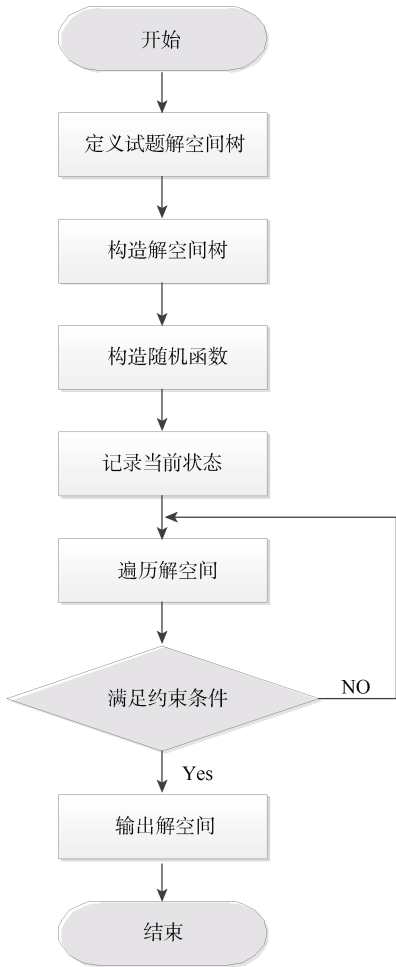


图 2 回溯试探算法流程图

从图 2 可以看出,回溯试探法在试题的状态空间树中,根据深度优先遍历从根结点按随机函数开始搜索解空间,每次搜索判断结点是否满足约束条件。如果满足,则记录状态,写入状态空间,继续遍历该节点子树;否则跳过该节点的子树,回溯到上一个状态,这样的操作称为“剪枝”^[15]。在该算法中,影响算法效率的主要因素是空间状态树遍历次数,如果能尽早剪掉无解的子树,即可避免大量无效的搜索。本文对此提出了一种分级回溯试探组卷算法,算法流程如图 3 所示。

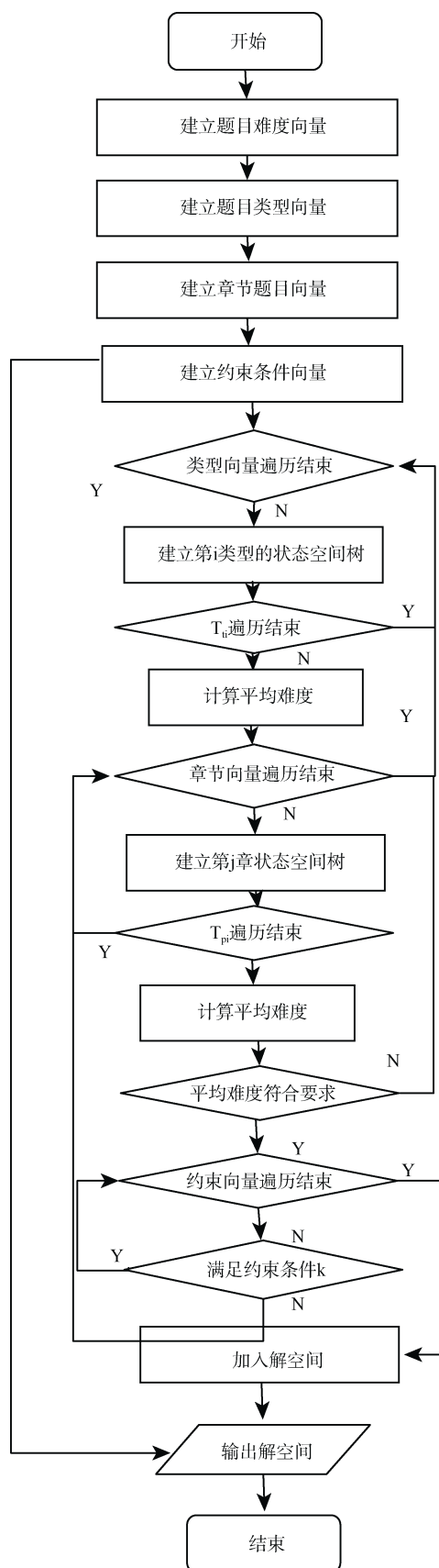


图3 分级回溯试探组卷算法流程

根据结构化程序设计思想, 一个复杂问题一般是由多个简单问题构成, 因此可以将解空间模块化为若干个简单解空间, 这些简单问题解空间构成一个决策序列, 所有可能的决策序列构成了这个问题的解空间。解空间中满足所有约束条件的解就是目标解, 达到目标的解也可能有多个。在多个目标解中, 一般还需要找到最优解, 最优解也可能是并列的多个。回溯试探分阶组卷算法过程中, 可以按照题目类型建立类型状态空间树, 并将每个类型对应的约束条件分解到每种类型的解空间上, 再进行遍历搜索, 对于不满足约束条件的结点直接进行剪枝。采用同样的方式再按章节对空间状态树进行剪枝, 还可以再按照难度对状态空间树进行剪枝等多阶段剪枝求解, 这样可以尽早地去掉无效的遍历, 而无需等到不满足所有条件后才剪枝, 因此提高了算法效率。

分级回溯试探组卷算法的具体流程如下。

1) 创建向量

(1) 建立 Q 个不同难度系数的试题向量 $\vec{u}D(d1, d2, d3, \dots, di, \dots, dq)$, 其中 di 表示第 i 类难度, 它是一个包括 3 个子元素的结点, $di1$ 表示难度系数, $di2$ 表示试题分值, $di3$ 表示试题数量, 且平均难度 $D_c = \sum_{i=1}^q di1 \times di3 / \sum_{i=1}^q di3$ 在 0.1~0.4 之间。

(2) 建立 N 个类型向量 $\vec{u}T(t1, t2, t3, \dots, ti, \dots, tn)$, 其中 ti 表示第 i 类试题, 它是一个包括 2 个子元素的结点, $ti1$ 表示试题数量, $ti2$ 表示试题分值, 且满足 $S_p = \sum_{i=1}^n ti1 \times ti2$, S_p 表示试卷总分。

(3) 建立 M 个章节向量 $\vec{u}P(p1, p2, p3, \dots, pj, \dots, pm)$, 其中 pj 表示第 j 章节, 它是一个包括 2 个子元素的结点, $pi1$ 表示试题数量, $pi2$ 表示试题比例, 且满足 $pj2 = \frac{pj}{\sum_{j=1}^m pj}$ 。

(4) 建立 L 个约束向量 $\vec{C}(c1, c2, L, ck, L, cl)$, 其中, ck 表示第 k 个约束条件。如果在约束条件 ck 下可以组卷成功, 记为 $f(ck)=1$, 否则记为 $f(ck)=0$ 。

2) 循环遍历类型向量 $\vec{T}(t1, t2, t3, L, ti, L, tn)$, 生成 N 个空间状态树的集合 $T_i\{t_{ti}, i \in N\}$ 。

3) 循环检索 t_{ti} 。当 t_{ti} 有解时:

(1) 根据 $\vec{D}(d1, d2, d3, L, di, L, dq)$ 计算平均难度 $D_{ct} = \sum_{i=1}^N di1 \times di3 / \sum_{i=1}^N di3$;

(2) 遍历章节向量 $\vec{P}(p1, p2, p3, L, pj, L, pm)$, 生成 M 个空间状态树的集合 $T_p\{t_{pj}, j \in N\}$ 。

4) 循环检索 t_{pi} 。当 t_{pi} 有解时:

(1) 根据 $\vec{D}(d1, d2, d3, L, di, L, dq)$ 计算平均难度 $D_{cp} = \sum_{i=1}^M di1 \times di3 / \sum_{i=1}^M di3$;

(2) 计算 $D_c = (D_{cp} + D_{ct}) / 2$, 如果 $D_c \notin [0.1 \sim 0.4]$, 回溯到 (2)。

5) 循环遍历约束条件向量 $\vec{C}(c1, c2, L, ck, L, cl)$ 。如果函数 $f(ck)=1$, 继续遍历 $\vec{C}(c1, c2, L, ck, L, cl)$, 否则回溯到 (2)。

6) 如果满足 $k \geq l$, 将解 c_{jh} 写入集合 $Cj\{c_{jh}, h \in N\}$ 。

7) 如果满足 $j \geq m$, 将解集合 Cj 写入解集合 $Ci\{c_{ij}, j \in N\}$ 。

8) 如果满足 $i \geq n$, 输出 Ci 解集合。

利用分级回溯试探组卷算法在具体使用过程中也可以根据其他指标进行分级求解, 但一般空间树不宜太深, 否则可能会造成算法效率下降。

3 算法效率分析建模测试

3.1 算法效率分析建模

分级回溯试探组卷算法主要根据平均组卷时

长、试题重复率、试卷覆盖率指标建立组卷算法分析模型。

1) 设一次组卷时长 L 为:

$L = Te - Ts$ (Te 为完成时间, Ts 为开始时间)。

平均组卷时长 $\bar{L}$ 如下:

$$\bar{L} = \frac{\sum_{i=1}^n L_i}{n} \quad (1)$$

2) 试题重复率 R_r , 表示每套试卷中每一试题在所有试卷中出现的比率。

设 N 表示试卷总数, N_p 表示每套试卷题目总数, f_i 表示试题 i 在试卷中出现的频率, f_i 计算如下:

$$f_i = \sum_{i=1}^N \sum_{p=1}^{N_p} f(i) \quad (2)$$

其中 $f(i)$ 计算如下:

$$f(i) = \begin{cases} 1 & \text{题库中第 } i \text{ 题与某试卷中一题相同} \\ 0 & \text{题库中第 } i \text{ 题与某试卷中一题不相同} \end{cases} \quad (3)$$

R_r 计算如下:

$$R_r = \frac{\sum_{p=1}^{N_p} f_i / N}{N_p} \quad (4)$$

3) 试卷覆盖率 R_c , 表示一套试卷所包含的指标占题库指标总数的比率, 计算如下:

$$R_c = \frac{C_p}{C_d} \quad (5)$$

其中, C_p 表示一套试卷所包含指标数量, C_d 表示题库指标总数量。

平均覆盖率 $\bar{R}_c$ 计算如下:

$$\bar{R}_c = \frac{\sum_{i=1}^N R_c}{N} \quad (6)$$

覆盖率标准偏差 S 计算如下:

$$S = \sqrt{\frac{\sum_{i=1}^N (R_{ci} - \bar{R}_c)^2}{N - 1}} \quad (7)$$

3.2 试验及结果分析

将分级回溯试探组卷算法与随机函数抽取

法、遗传算法及传统回溯试探组卷算法分别用于 B/S 结构的 OJ 系统中，服务器基本配置为 CPU：2.4G×2；内存：16G。组卷试验分别针对三种规模的题库进行，采用相同的试题结构、试卷结构，如表 1 所示，题库包括四个类型题目：单项选择题、多项选择题、填空题、判断题。

表 1 实验组卷要求

序号	要求
1	满分 100 分
2	单项选择 40 道，每题 1 分；多选题 20 道，每题 2 分；填空题 10 道，每题 1 分；判断题 10 道，每题 1 分
3	平均难度 0.2
4	试题分布：第 1 章占 6%；2-3 章各占 12%；4-7 章各占 15%；第 8 章占 10%
5	概念题占 20%，综合应用题占 60%，计算题占 20%
6	试卷份数 150 套

首先使用《程序设计基础》题库，题库中单选 3000 道，多选 2000 道，填空及判断各 1000 道，共 7000 题，试验结果如表 2 所示。

表 2 四种算法在《程序设计基础》题库中的组卷结果

算法 指标	随机 抽取	遗传 算法	传统 回溯	分级回溯 试探
$\bar{L}$	1.35	0.73	1.15	0.75
Rr	4.24%	2.36%	3.16%	2.85%
$\overline{Rc}$ (章节)	91%	99%	95%	100%
S (章节)	0.041	0.003	0.029	0
$\overline{Rc}$ (难度)	96.8%	99.2%	99.2%	98%
S (难度)	0.033	0.014	0.021	0.012
$\overline{Rc}$ (知识类型)	97.9%	99.6%	99.3%	98.1%
S (难度)	0.025	0.013	0.018	0.011

试验结果表明，分级回溯试探组卷算法在平均组卷时间、重复率方面与传统回溯算法及随机抽取算法相比都有了明显降低，在平均覆盖率方面相差无几（有高有低），但覆盖率标准偏差上有明显提高，遗传算法在平均组卷时间、重复率方面表现较好，但覆盖率偏差与分级回溯试探组卷算法相比较，说明遗传算法缺乏稳定性。

第二次试验使用《计算机基础》题库，题库中单选 8000 道，多选 4000 道，填空及判断各 2000 道，共 16000 题，试验结果如表 3 所示。

表 3 四种算法在《计算机基础》题库中的组卷结果

算法 指标	随机 抽取	遗传 算法	传统 回溯	分级回溯 试探
$\bar{L}$	1.98	0.97	1.75	0.9
Rr	3.45%	1.85%	2.64%	1.92%
$\overline{Rc}$ (章节)	93%	99.6%	96.7%	100%
S (章节)	0.034	0.002	0.021	0
$\overline{Rc}$ (难度)	97.6%	99.6%	99.4%	99%
S (难度)	0.029	0.011	0.018	0.009
$\overline{Rc}$ (知识类型)	98.3%	99.8%	99.4%	99%
S (难度)	0.021	0.01	0.013	0.007

第二次试验结果表明，随着题库规模增大，四种组卷算法的组卷时间都有所增加，组卷效率和组卷质量都有明显提高，但分级回溯试探组卷算法组卷时间增加幅度较小，组卷效率和组卷质量提高的幅度更大。

第三次试验使用《高等数学》题库，题库中单选 20000 道，多选 10000 道，填空及判断各 4000 道，共 38000 题，试验结果如表 4 所示。

表 4 四种算法在《高等数学》题库中的组卷结果

算法 指标	随机 抽取	遗传 算法	传统 回溯	分级回溯 试探
$\bar{L}$	2.58	1.17	2.25	1.03
Rr	2.48%	1.25%	2.03%	1.14%
$\overline{Rc}$ (章节)	94%	99.6%	97.1%	100%
S (章节)	0.031	0.002	0.017	0
$\overline{Rc}$ (难度)	97.9%	99.8%	99.3%	99.7%
S (难度)	0.023	0.008	0.015	0.006
$\overline{Rc}$ (知识类型)	98.9%	99.9%	99.3%	99.7%
S (难度)	0.019	0.009	0.012	0.005

第三次试验结果同样表明，当题目规模继续增大时，分级回溯试探组卷算法各方面表现依然较好。

4 结语

本文通过研究随机函数抽取法、遗传算法、

回溯试探法等几种常见组卷方法的算法机制, 针对这些算法的优缺点, 以回溯试探法为基础, 按照分级求解、早剪枝、瘦算法的思想, 采用题型、章节、难度等指标, 建立了分级回溯试探组卷算法模型。试验结果可以看出, 随着题库题目数量的增加, 分级回溯试探组卷算法组卷效率高, 试题重复率低、覆盖率高, 组卷质量较高。

参考文献

- [1] Chao Yi, Xiao Yudu, Su Feng, et al. The Design of Course-Oriented Online Judge[J]. Advanced Materials Research, 2012, 1566(433): 4736-4740.
- [2] 李欣然, 樊永生. 改进量子行为粒子群算法智能组卷策略研究[J]. 计算机科学, 2013, 40(4): 236-239.
- [3] 王建忠, 张 萍, 吴 倩, 等. 考试系统中题库量与试题量的关系研究[J]. 计算机应用研究, 2010, 27(2): 611-613.
- [4] Lei Liu, Hui Shi, Rui Zhai. The Design and Implementation of Online Exam System[J]. Applied Mechanics and Materials, 2014, 3634(687): 2506-2509.
- [5] 江卓军, 陈 璇, 苏晓兰, 等. 一种复杂约束下的采购评审专家自动抽取算法[J]. 现代电子技术, 2018, 41(3): 75-79.
- [6] Li Yan, Li Shuhong, Li Xiurong. Test Paper Generating Method Based on Genetic Algorithm[J]. AASRI Procedia, 2012, 1: 549-553.
- [7] 杨素锦, 陈 莹. 基于改进遗传算法的自动组卷问题研究[J]. 煤炭技术, 2012, 31(12): 217-219.
- [8] 刘怀兰, 牛 辉, 王 佳. 基于改进遗传算法的智能组卷模型优化[J]. 华中科技大学学报(自然科学版), 2013, 41(5): 82-85.
- [9] 王 东. 基于改进随机抽取策略的智能组卷算法[J]. 山东理工大学学报(自然科学版), 2017, 31(4): 19-23, 29.
- [10] 陈青青. 递归随机分割算法在考试系统中的应用[J]. 计算机工程, 2011, 37(14): 268-270, 273.
- [11] 方 敏, 李 浩. 基于状态回溯代价分析的启发式 Q 学习[J]. 模式识别与人工智能, 2013, 26(9): 838-844.
- [12] Zou Feng, Chen Debao, Li Suwen, et al. Community Detection in Complex Networks: Multi-objective Discrete Backtracking Search Optimization Algorithm with Decomposition[J]. Applied Soft Computing, 2017, 53: 285-295.
- [13] 鲁 萍, 王玉英. 多约束分级寻优结合预测计算的智能组卷策略[J]. 计算机应用, 2013, 33(2): 342-345.
- [14] Chen Debao, Lu Renquan, Zou Feng, et al. A learning and niching based backtracking search optimisation algorithm and its applications in global optimisation and ANN training[J]. Neurocomputing, 2017, 266: 579-594.
- [15] 高欢欢. 分类分段回溯试探算法在计算机基础在线考试系统中的应用[J]. 软件导刊, 2016(5): 157-159.

How to Generate the Examination Paper: A Novel Hierarchical Backtracking-Based Algorithm

Li Chuan¹, Zhang Shaoru²

(1. College of Computing, Xi'an Aeronautical University, Xi'an 710077, China;

2. Health Science Center, Xi'an Jiaotong University, Xi'an 710061, China)

Abstract: The random function method, genetic algorithm, and backtracking search algorithm are widely used to generate paper. However, typical paper-generating algorithms are inefficient to the point that they cannot ensure generate high-quality paper. In this study, we focus on this problem and devise a novel hierarchical backtracking paper-generational algorithm (HBPA). The HBPA optimizes the traditional paper-generating algorithm by using a hierarchical structure for pruning to solve the problems of ineffectiveness and inefficiency in the paper generation process. In addition, to measure the efficiency of the algorithm, an algorithm analytical model is established. Evaluation results reveal the effectiveness of the algorithm's execution, whereby the quality of paper generation improves significantly. In addition, the number of test questions is increased.

Key words: backtracking; hierarchical; pruning; examination paper generating