第37卷 第14期 计 算 机 工 程 2011年7月
Vol.37 No.14 Computer Engineering July 2011

·开发研究与设计技术· 文章编号：1000—3428(2011)14—0268—03 文献标识码：A 中图分类号：N945

递归随机分割算法在考试系统中的应用

陈青青

(湖南工学院网络中心，湖南 衡阳 421002)

摘 要：针对考试系统中的试卷生成问题，提出一种基于题型的递归随机分割算法。该算法根据输入的试卷生成参数，利用概率密度函数与矩阵方程计算参数，确定各难度级别中不同题型的试题抽取数量，从而生成试卷。通过对算法的分析和生成试卷的测评，证明该算法的效率较高且通用性强，能够较好地满足考试系统需要。

关键词：试题库；递归随机分割算法；考试系统；概率密度函数；矩阵方程

Application of Recursion Random Division Algorithm in Examination System

CHEN Qing-qing

(Network Center, Hunan Institute of Technology, Hengyang 421002, China)

【Abstract】Aiming at the test papers generating problems in examination system, a recursion random division algorithm is presented. The algorithm need input parameters for generating test papers. It uses probability density function and matrix equation to calculate parameters. It confirms the number of draw-out questions in different types and different difficulty. It uses the algorithm to generate test papers. The efficiency and adaptivity of the algorithm are demonstrated by analyzing the algorithm and evaluating the generated test papers, and the algorithm meets the requirements of examination system well.

【Key words】item pool; recursion random division algorithm; examination system; probability density function; matrix equation

DOI: 10.3969/j.issn.1000-3428.2011.14.091

1 概述

考试作为选拔人才最主要的手段之一，是教学过程中十分重要的环节。随着教育思想的更新和科学技术的进步，以纸和笔为主要工具的传统考试方式的诸多弊端显露出来[1]。为此，本文提出一种基于题型的递归随机分割算法，并将其应用于考试系统中，以生成考试试卷。

2 试卷生成算法分析

考试系统的试卷生成方式[2]是考试系统设计和开发的关键，常用的一般有随机选取算法、回溯试探算法、遗传算法(Genetic Algorithm, GA)等。

2.1 随机选取算法

随机选取算法是根据用户设置的试卷生成参数，由考试系统程序利用随机函数或随机量自动从试题库中抽取若干分量构成试题状态类型，再从试题库中随机选取一道试题加入试卷内容中，此过程不断重复，直到生成试卷完毕。

2.2 回溯试探算法

回溯试探算法是对随机选取算法的改进，它将试卷参数所设置的每种状态试题选取数量记录下来，当生成试卷失败时释放上次记录的状态类型所选取的试题，然后根据一定的规律变换出一种新的状态试题选取数量生成试卷，通过不断回溯试探直至生成试卷成功或失败。

2.3 遗传算法

遗传算法[3]是一种模拟自然界生物进化过程的随机优化算法，它首先根据要求选择一定规模的初始群体，然后使初始群体中的个体以一定的概率进行交叉与变异，实现个体结构的重组，再按预定的评价函数选择优秀个体，组成新的一代，如此循环迭代，最终找到满足寻优条件的全局最优解。

3 试卷生成方案

上述几种常用算法各有优缺点，而其本质上是根据用户设置的试卷生成参数，按照一定的试卷生成算法，从试题库中选择试题内容生成试卷，使试卷能满足考试要求，具有随机性、科学性和合理性。

在试卷生成过程中，一套试卷是否合理，主要由试卷的难易程度来体现，因此，准确确定试题的难度级别以及难度分布是否合理至关重要。一份好的试卷应该使考生成绩大致呈正态分布。

一般来说，用户在生成试卷时会提出多方面的要求，如难度、试题类型、总题量等，但所生成的试卷应满足如下基本原则：(1)试卷难度适中；(2)试卷知识点的覆盖面广；(3)试卷公平性要保证[4]。

按此原则，对于任何一门课程的考试试卷，可以看成是由这样一些属性值共同决定的。分别是课程名称、考试日期、考试时间、试卷总分、试题的知识点、试题的题型、各题型的分值、各题型试题的个数、试卷的总体难度。其中，课程名称、考试日期、考试时间、试卷总分为确定值，只要能计算出试卷的其他属性，即可确定具体试卷。因此，试卷生成方案主要分为2步：

第1步 根据输入试卷参数中的试题类型、题型分值、题型试题个数和试卷整体难度，确定对应难度级别中不同试题类型的试题抽取数量。

作者简介：陈青青(1983－)，男，讲师、硕士，主研方向：网络管理，数据库应用

收稿日期：2011-03-15 **E-mail：**cqq@hnpu.edu.cn

第 2 步 使用基于试题类型的递归随机分割算法，在按题型分割的试题库中按难度级别随机从约束的知识点中选取试题内容，生成试卷。

4　试卷生成参数计算

试卷中难度高的题目比例越大，得低分的考生越多，反之，得高分的考生多，一般希望学生得分多集中在平均分周围。若用固定正态分布平移的方法表示学生得分情况，当平均分靠近 100 分时，会造成百分以上的概率无法忽略。所以，这里用 2 个正态分布：高分学生得分分布以及低分学生得分分布，合起来表示学生的得分情况，如图 1 所示，分界点为 p。

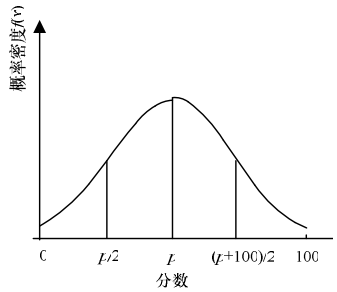


图 1　得分正态分布

在图 1 中，概率密度函数 $f(x)$可表示为正态分布函数(略)。所以数学期望，即总体平均分为：

$$E(x)=p+(\delta_2-\delta_1)/\sqrt{2\pi} \tag{1}$$

令学生得分在$[p,(100+p)/2]$、$[p/2,p]$之间的比例均为 40%。即可解得：$\delta_1=p/3.588$，$\delta_2=(100-p)/3.588$。代入式(1)得：$p=(9\times E(x)-100)/7$。

该结果表明：当给定一个平均分 $E(x)$，就能够确定分界点 p，也就求得了 δ_1、δ_2，即概率密度函数 $f_1(x)$、$f_2(x)$。

若规定得分区间在[0,25]试卷总体难度为难，在[25,50]为较难，在[50,75]为较易，在[75,100]为易。图 2(a)~图 2(d) 4 个概率密度分别为 p 在不同得分区间范围取值时，试卷中各难度等级试题所占百分比概率密度分布情况。

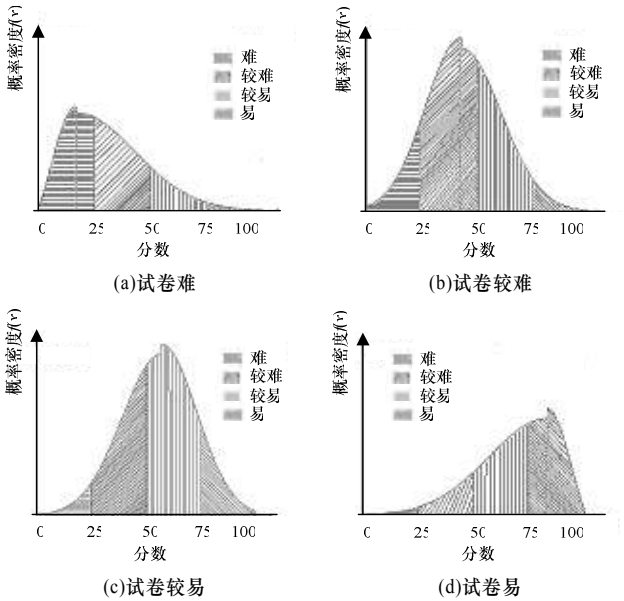


图 2　p 值在不同区间的各种概率密度

由此可得表 1 所示的各难度等级试题分值所占百分比计算结果。

表 1　各难度等级试题所占百分比计算结果

难度	$0x\leqslant p\leqslant 5$	$25<p\leqslant 50$	$50<p\leqslant 75$	$75<p\leqslant 100$
难	1-较难-较易-易	$\int_0^{25} f_1(x)\mathrm{d}x$	$\int_0^{25} f_1(x)\mathrm{d}x$	$\int_0^{25} f_1(x)\mathrm{d}x$
较难	$\int_{25}^{50} f_2(x)\mathrm{d}x$	1-难-较易-易	$\int_{25}^{50} f_1(x)\mathrm{d}x$	$\int_{25}^{50} f_1(x)\mathrm{d}x$
较易	$\int_{50}^{75} f_2(x)\mathrm{d}x$	$\int_{50}^{75} f_2(x)\mathrm{d}x$	1-难-较难-易	$\int_{50}^{75} f_1(x)\mathrm{d}x$
易	$\int_{75}^{100} f_2(x)\mathrm{d}x$	$\int_{75}^{100} f_2(x)\mathrm{d}x$	$\int_{75}^{100} f_2(x)\mathrm{d}x$	1-难-较难-较易

假定输入试卷参数对要生成的试卷选取了 n 种题型($1\leqslant n\leqslant$系统中题型总数)，分别记为 $T1,T2,\cdots,Tn$。各题型难度级别有易、较易、较难、难这 4 种，分别记作 1,2,3,4。可构成如下各难度级别试题数矩阵 $\boldsymbol{Q}_4\times n$：

$$\begin{array}{c} \\ \text{难度级别} \end{array}\begin{array}{c} \text{题型} \\ \begin{array}{cccc} T1 & T2 & & Tn \end{array} \end{array}\begin{array}{c} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} \end{array}\begin{bmatrix} q_{11} & q_{12} & \mathrm{L} & q_{1n} \\ q_{21} & q_{22} & \mathrm{L} & q_{2n} \\ q_{31} & q_{32} & \mathrm{L} & q_{3n} \\ q_{41} & q_{42} & \mathrm{L} & q_{4n} \end{bmatrix}$$

其中，q_{ij} 是各类题型不同难度级别的题数，有 $q_{ij}\geqslant 0$。

令由表 1 所确定的各难度级别所占分值向量为 $\boldsymbol{S}=(x,y,z,r)$，其中，x、y、z、r 分别是易、较易、较难、难题所占的分值。满足：$x+y+z+r=$卷面总分。再设 n 种题型的单位分值列向量为 $\boldsymbol{L}=(t1,t2,\cdots,tn)$，其中，$t1,t2,\cdots,tn$ 分别是 n 种题型的单位分值。于是可得：$\boldsymbol{Q}\times\boldsymbol{L}=\boldsymbol{S}$。从而可得 $\boldsymbol{Q}$ 矩阵，即求得各题型不同难度级别的题数 q_{ij}。

需要说明的是，对矩阵方程的求解，其结果可能会出现正数、负数或小数的情况，可约束结果为正整数，但一般还是会出现多个解；对各难度等级试题所占百分比按表 1 计算，但结果可能会出现小数，这样很可能造成难度等级分值与试题分值不对应而无法生成试卷。对于矩阵方程求解，可以采用回溯法将方程的解全求出，用户选择其中一个满意的 $\boldsymbol{Q}$；对于各难度等级试题分布所占百分比，计算后的结果给出近似值后由用户手动调整好。

比如现在要按此方法生成 C 语言考试试卷，设置试卷参数：卷面总分为 100 分，试卷难度为较易(选择各难度级别所占分值容易题目 15 分、较易题目 45 分、较难题目 30 分、难题目 10 分)，试题类型有选择题 2 分/个×25 个共 50 分，填空题 3 分/个×10 个共 30 分，简答题 10 分/个×2 个共 20 分；知识点有：常量，变量，数据类型等 14 个。3 种题型分别记为 T1(选择题)、T2(填空题)、T3(简答题)，计算出 $\boldsymbol{Q}$ 矩阵结果可以为：$\boldsymbol{Q}1=\begin{bmatrix}3&3&0\\13&3&1\\9&4&0\\0&0&1\end{bmatrix}$，$\boldsymbol{Q}2=\begin{bmatrix}3&3&0\\13&3&1\\7&2&1\\2&2&0\end{bmatrix}$ 等。

选择其中一个合适的 $\boldsymbol{Q}$，作为下一步递归随机分割算法的入口参数去控制试题的抽取。

对于 $\boldsymbol{Q}$ 的求解，系统采用回溯法求出全部解，不过这些都是在生成试卷之前独立完成，而且对于同一门考试课程来说，要生成多套试卷，其 $\boldsymbol{Q}$ 矩阵和各难度级别所占百分比均只需计算一次，所以并不会对整个试卷生成算法性能造成影响。

5　基于题型的递归随机分割算法

按照上述思路，在确定各题型不同难度级别的试题数量后，在 j 种题型中，共有题目数为 q_j，不同难度级别试题抽取数分别为 $q_{1j},q_{2j},\cdots,q_{ij}$，且 $q_{1j}+q_{2j}+\cdots+q_{ij}=q_j$，其关系如图 3 所示。

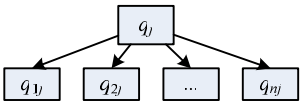


图 3 j 种题型中各难度题数间的关系

可以看出，不同难度级别的题数是相互独立的，因此，若能求出 q_{1j} 所选择的试题内容，循环 n 次，其他的就都能求出，则试卷生成完成。

5.1 基于题型的递归随机分割算法的基本思想

首先将同一难度、同一类型试题 q_{1j} 按随机比例划分为 q_{1j1} 和 q_{1j2} 2 个部分，然后将 2 个部分的试题进行随机分割，依次递归找出不同试题类型的试题直至生成试卷完成为止。如图 4 所示。

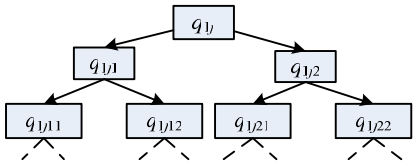


图 4 随机比例分割

则满足：

$q_{1j1}+q_{1j2}=q_{1j}$(第 1 次划分)

$q_{1j11}+q_{1j12}=q_{1j1}$(第 2 次前半部分划分)

$q_{1j11}+q_{1j12}+q_{1j21}+q_{1j22}=q_{1j}$(第 2 次全划分)

5.2 基于题型的递归随机分割算法

基于题型的递归随机分割算法，采用递归结构，算法的伪代码如下：

```
Sub Ranlocate(idL As Integer, idH As Integer)
If idL<idH Then
        Randomize
        id = Int((idH–idL + 1) * Rnd()) + idL
        '选出变量 id 中指出的试题,将其内容添加到试卷中并设置分值
        Count = Count + 1
        l1 = idL: r1 = id - 1: l2 = id + 1: r2 = idH
        If Count = qij Then End
          If id mod 2=0 then
          Call Ranlocate(l1,r1) '递归左子树选题
          Call Ranlocate(l2,r2) '递归右子树选题
          else
          Call Ranlocate(l2,r2) '递归右子树选题
          Call Ranlocate(l1,r1) '递归左子树选题
          End If
      End If
      If idL=idH Then '只有一题时递归终止
        id = idL
        '选出变量 id 中指出的试题,将其内容添加到试卷中并设置分值
        Count = Count + 1
        If Count = qij Then End
End If
End Sub
```

在上述代码中，在递归调用时一定要随机选择递归调用的方向，否则在试题数量足够多的情况下，会出现始终递归调用左(右)子树中的试题生成试卷，导致右(左)子树中的试题永远无法被选取。

5.3 基于题型的递归随机分割算法分析

从图 4 可以看出，采用递归随机分割算法，根据满二叉树性质，当 q_{ij} 为 N 时，该分割下试题数量大于 2^N-1 时，其最好时间开销为 $O(N)$，最坏时间开销 $2\cdot O(N)$，且时间开销会随着该分割下题量的增加而趋向于 $O(N)$。

从上述算法中可以看出，用这种递归随机分割算法，不为选取的试题设置选中标志，也不会出现重复选取试题的情况；题型数量 q_{ij} 的有限性保证了递归调用深度；该算法不但结构简单、高效、易行、且每道试题被选中的概率分布均匀，通用性强。

6 生成试卷质量测评

应用基于题型的递归随机分割算法生成的试卷是否能真实地反映出教师的教学水平以及学生对知识的掌握程度，需要对试卷质量进行质量测评。测量试卷的主要检验指标有：试卷难度和知识点覆盖度。

6.1 试卷难度

试卷的难度与考生的水平是有一定关系的。试卷难度计算的方法，都要在测试或正式考试之后才能进行分析，这样对考生成绩的结果进行预测或者对试卷难度的把握程度可能会有很大出入。因为影响试卷难度的主要因素是多方面的：包括考生对知识的掌握程度、试题自身属性的正确性、试卷的题型结构、试题内容本身是否有迷惑性等。因此，可以在试卷生成后对试卷难度进行粗略估算。

设试卷中题目总数量为 $\boldsymbol{Q}$，4 种难度级别题目数量分别为 q_1,q_2,q_3,q_4；试卷总分为 M，4 种难度级别所占分值分别为 m_1,m_2,m_3,m_4，已抽取的试卷中各试题难度级别分别为 b_{ij}，则各难度级别的平均难度为各试题难度级别的和除以试题数量；试卷难度为各难度级别平均难度乘以分值所占百分比之和。

6.2 知识点覆盖度

反映试卷对知识点覆盖广度的重要指标，即试题应尽量反映该门课的知识点。这里用生成试卷中所选试题占知识点个数表示。

通过该算法生成的 C 语言试卷，试卷参数按前述要求设置，选择矩阵 $\boldsymbol{Q}$**1** 生成试卷。在生成时不检测试卷质量，一次性生成的 60 份试卷中，生成后随机抽取 6 份，应用这 2 个主要测评指标对试卷内容统计，结果如表 2 所示。

表 2 试卷测评结果

试卷号	试卷难度	知识点覆盖个数
1	0.456	14
2	0.433	11
3	0.454	12
4	0.479	13
5	0.440	12
6	0.471	11

由测评指标统计结果可知，采用基于题型的递归随机分割算法生成的这些试卷基本达到了试卷参数所设置的要求。但在生成试卷的过程中，由于试卷中试题总个数 $\boldsymbol{Q}$ 为 37，而要求覆盖的知识点数量为 14 个，因此随机生成的试卷难以达到全覆盖的目的。由于试题为随机抽取，在特殊情况下，也可能会有达不到预期要求的试卷出现。主要有两方面：

(1)在生成的多份试卷中，试卷总体难度间的差异可能会较大，如果按要求从题库中抽取的所有试题难度级别都偏低或偏高，在极值情况下，试卷难度最低为 0.348，最高为 0.587。这对于同时参加考试的学生来说，是不公平的，需要进行调整。具体方法为：可以按试卷生成参数计算所生成试卷总体难度的均值为(0.348+0.587)/2 等于 0.467，对于所生成的多份试卷，

(下转第 273 页)