

高校自动排课算法的研究与设计^{*}

李志娟^{1), 2)} 王冠¹⁾
(北京工业大学¹⁾ 北京 100022) (北京联合大学生物化学工程学院²⁾ 北京 100023)

摘 要 描述排课的基本问题,排课系统的研究现状,以及设计排课系统时应该考虑的各种约束条件,在分析其他排课算法的同时结合北京联合大学生物化学工程学院的实际情况,设计出适合本校的排课算法,该算法编码简单,易于实现。测试结果表明,该算法能够满足排课的各种约束,排出的课表较为科学合理,具有一定的实用性。

关键词 自动排课 排课系统 算法 约束条件
中图分类号 TP31

Study and Design of the Algorithms of University Course Automatic Arranging

Li Zhijuan^{1), 2)} Wang Guan¹⁾
(Beijing University of Technology¹⁾, Beijing 100022)
(Biochemical Engineering College of Beijing Union University²⁾, Beijing 100023)

Abstract The paper specified the basic problem of university course automatic arranging, the status of university course automatic arranging system existed, and some restrictive condition of university course automatic arranging system. Analyzing the algorithms of university course automatic arranging else, the thesis designed a algorithm based on Biochemical Engineering College of Beijing Union University. The algorithm is easy realized. The test shows that the university course scheduler by the algorithm is scientific and applied.

Keyw ords course automatic arranging, course arranging system, algorithm, restrictive condition
Class Num ber TP31

1 引言

排课问题的本质是将课程、教师和学生合适的时间段内分配到合适的教室,考虑的因素比较多,既要避免教师、学生、教室在时间上的冲突,保证教学工作的正常开展,又要讲求科学性。利用计算机进行自动排课,不但能使教务人员从繁杂的排课任务中解脱出来,大大提高教务管理工作的效率,改善教学管理质量,合理高效地利用资源,而且对于优化学生的学习进程、评估每位教师对教学的贡献、领导的合理决策等都具有重要意义,可以间接提高教学质量,促进教学的良性循环。

排课问题早在上个世纪 70 年代就证明是一个 NP 完全问题,即算法的计算时间是呈指数增长的,这一论断确立了排课问题的理论深度。目前人们

对 NP 完全问题研究的主要思想是如何降低其计算复杂度,即利用一个近似算法来代替,力争使得解决问题的时间从指数增长化简到多项式增长。结合到课表问题就是建立一个合适的现实简约模型,利用该简约模型能够大大降低算法的复杂度,便于程序实现,这是解决排课问题一个很常见的思路。目前常见的排课算法有以下几种:

1)遗传算法

遗传算法是美国 Michigan 大学 J. Holland 教授于 1975 年首先提出来的,它是模拟达尔文生物进化论的计算模型,是一种具有鲁棒性、全局最优性、可并行处理性及高效性的搜索算法,用来对复杂系统进行优化,该算法应用到排课系统中的主要演算步骤如下:

(1)根据排课因素产生基因编码和染色体,并

^{*} 收稿日期: 2007 年 11 月 2 日,修回日期: 2008 年 3 月 11 日
作者简介: 李志娟,女,硕士研究生,研究方向:数据库技术,数据挖掘和智能信息系统。王冠,男,硕士研究生,副教授,研究方向:数据库技术,数据挖掘和智能信息系统。
© 1994-2014 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

随机产生一定数目的初始种群 (一定数目的班级课程表); (2)对个体 (班级课程表) 适应度进行评估, 如果个体的适应度符合优化准则, 则输出最佳个体及其代表的最优解, 并结束计算, 否则转向第 (3) 步; (3)依据适应度选择再生个体; (4)按照一定的交叉概率和交叉方法生成新的个体; (5)按照一定的变异概率和变异方法生成新的个体; (6)由交叉和变异生成新一代的种群, 然后返回第 (2) 步。最后进行冲突的检测与消除。

2) 蚂蚁算法

20 世纪 90 年代初意大利学者 Dorigo Maniezzo 依照蚂蚁觅食原理提出了第一个“蚂蚁算法 (ant colony algorithm) ”。用蚂蚁算法来解决高校排课问题首先要构造二部图和蚂蚁个体, 并赋予蚂蚁一定的周游“能力”, 采用信息素策略来解决问题, 即采用 MMAS 思想。使用蚂蚁算法来解决排课问题的效率不如使用现在比较成熟的算法来排课的效率, 但这为排课问题提供了一种新的解决思路, 扩大了蚂蚁算法的应用。

3) 基于规则的算法和回溯算法

基于规则的排课算法主要是引入了关联规则对冲突严重的课程进行优先处理, 极大地减少冲突死锁的概率, 保证所有课程能够全部编排。

回溯算法是一种搜索一棵状态树的过程, 具体到排课算法就是按顺序处理某一课表中的各门课程, 当处理到某一课程时发现没有解, 即无法成功地分配时间片或者教室资源, 则修改上一课程的分配结果, 然后再重新对该课程进行资源的分配。为了使得到的课表符合硬性约束、较为科学合理并具有较高的适应度, 在分配的过程中必须引入评价和惩罚机制, 确保每一步都是比较优化的选择, 从而使整个排课结果也比较优化。实现评价机制的关键是编码和对应的适应度评价的函数。

4) 基于优先级的算法

基于优先级的算法主要是通过划分等价类、计算优先级等方法将整个问题分层处理, 从而使矛盾分散在各个子问题当中, 并通过逐层处理达到了降低算法复杂性、减少死锁的目的, 获得较快的处理速度和比较满意的处理结果, 实现了计算机自动排课功能。

除了以上算法之外还有基于图论的算法, 贪心算法等等, 这些算法都比较成熟, 但在系统开发上比较复杂, 有一定的难度。

近几年, 计算机自动排课问题已经成为软件公司的一个课题, 但每个高校的教学资源配置情况不同, 办学定位和办学层次决定了高校的教学模式以

及管理方法不同, 编排课程表时考虑的因素及权重也不同, 直接导致目前现有的计算机自动排课系统的通用性不强。所以, 对于大多数高校, 需要根据本校的具体情况来设计自动排课软件。

本文主要针对我校的具体情况设计一种简便、易于系统实现的一种算法, 在课表优化上有针对性, 考虑的较为全面。

2 排课的基本问题

排课的主要任务是对教师、教室、班级、时间、课程 5 个因素进行最优化组合配置, 保证充分发挥各资源优势和提高教学质量。课程表应该满足以下约束:

- 1) 同一教学班级的学生同一时间不能安排两门以上的课程;
- 2) 同一教师同一时间不能安排两门以上的课程;
- 3) 同一教室同一时间不能安排两门以上的课程;
- 4) 同一时间安排课程总数不能大于所能提供的教室总数;
- 5) 某一教学班的人数不能大于所安排教室的容量;

课表除了满足以上硬性约束之外, 排课时还应该遵循以下原则:

1) 有效原则。这是排课总的原则, 应该根据每类课程的特点, 把课程安排在上课效果最好的时间, 比如数学、物理、化学、理工类专业课、需要计算分析的课程等应尽量安排在上午, 文科类课程、体育课应该安排在下午等。

2) 交错原则。要交错安排特点不同的学科, 比如需要逻辑思维和形象思维的学科要交错安排, 脑力消耗大的和脑力消耗小的学科要交错安排, 体力消耗大的和脑力消耗大的要交错安排等。

3) 分散原则。首先, 一门课要尽量分散在一个星期中, 即某天上完某一门课后, 要隔一天以上再上这门课, 这样教师就有足够的时间来备课和批改作业, 学生也有足够的时间来复习消化。其次, 由于人的精力有限, 为了保证教师课堂授课效果, 一个教师的课不能排满一整天。学生课表中的上课时间不能过分集中, 应该避免一天课程很满而另外一天课很少或者没课的情况。

4) 优先原则。由于教师兼任行政职务或进修或其他特殊困难对排课有特定要求的要优先处理; 公共课, 合班专业课等涉及面较广、学时多的课程应该优先处理。

5) 相对固定原则。同一教师、同一课程应尽量选择相对固定的几个教室。

另外还应考虑学生相邻两堂课的教室距离不能太远, 要保证足够的时间作上课准备等。

3 高校自动排课算法设计

3 1 数据库的设计与实现

主要的表结构描述如下:

课程信息表 (课程编号, 课程名称, 课程性质, 上课学时, 周学时, 上课班级, 教师代码, 教师姓名, 是否已排, 优先级);

班级信息表 (班级名称, 人数);

教师信息表 (教师编号, 教师姓名, 性别, 年龄);

教室信息表 (教室代码, 房间号, 教室类型, 可容纳人数, 是否已排)。

以上四个表中的大部分数据可从《教务管理系统》数据库中通过 SQL 语句提取, 其中每门课任课教师由教务人员根据每学期教学任务书输入。

时间模式表 (时间代码, 节次, 具体时间段)。

这里需要特别说明的是, 根据本校的实际情况笔者把每周的上课时间分为 15 个时间片 (每天 3 个时间片, 每周按 5 天计算), 如表 1 所示。

表 1 时间模式表		
时间代码	节次	具体时间
01	周一 1- 2 节	8 00- 9 35
02	周一 3- 5 节	9 45- 12 10
03	周一 6- 8 节	13 20- 15 45
04	周二 1- 2 节	8 00- 9 35
05	周二 3- 5 节	9 45- 12 10
06	周二 6- 8 节	13 20- 15 45
07	周三 1- 2 节	8 00- 9 35
08	周三 3- 5 节	9 45- 12 10
09	周三 6- 8 节	13 20- 15 45
10	周四 1- 2 节	8 00- 9 35
11	周四 3- 5 节	9 45- 12 10
12	周四 6- 8 节	13 20- 15 45
13	周五 1- 2 节	8 00- 9 35
14	周五 3- 5 节	9 45- 12 10
15	周五 6- 8 节	13 20- 15 45

3 2 排课的优先级原则

- 1) 任课教师由于特殊情况对上课时间有具体要求的, 先手工固定上课时间。
- 2) 其次编排公共基础课, 因为公共基础课面向全院, 一般都为合班课, 涉及到的班级较多。
- 3) 编排其他的合班课。
- 4) 对剩下的单班课, 先排重头课教师的课, 再排单头课教师的课。

按以上原则给待排课程排序, 后依次编排。

3 3 课程表的优化目标

课程表的优化目标有 5 个, 分别描述如下:

1) 不同类型的课程在不同的时间段学习的效率不同, 排课时要尽量使学习效果最佳。

根据学习心理学, 将课程划分为以下四大类:

(1) 逻辑性强的课程: 如数学、物理、信号与系统、电路分析等;

(2) 记忆性强的课程: 如与语文、外语等;

(3) 综合类课程: 即既有理解记忆的部分, 又有逻辑推理, 如某些专业课, 会计类课程等;

(4) 体育类课程。

另外还有一类课程就是操作类的, 比如各种实验, 上机实训, 课程设计等, 我校对这类课程单独安排, 实验和上机找学生空闲的时间, 有可能在下午四点以后或者晚上; 实训和课程设计属于集中实践环节, 一般在学期末的时候集中几周时间来进行, 所以不存在排课的问题。

依据学习心理曲线, 给出学习效率系数如表 2 所示 (满分为 10 分)。

表 2 学习效率系数表				
	逻辑	记忆	综合	体育
1- 2 节	9	10	9	1
3- 4 (或 3- 5) 节	7	6	6	4
6- 7 (或 6- 8) 节	4	4	4	10

2) 一周内上两次或两次以上的课程 (一般为两次或三次), 两次间隔均匀, 这样便于教师备课、批改作业, 利于学生预习功课, 复习巩固所学的知

表 3 周三次课程间隔合理性分值表				
时间	一、三、五	一、二、四 一、三、四	一、二、五 一、四、五	一、二、三 二、三、四 三、四、五
得分	10	6	7	4

表 4 周二次课程间隔合理性分值表				
时间	一、四 二、五 一、五	一、三 三、五 二、四	一、二 三、四	二、三 四、五
得分	10	8	4	

3) 教师少来校, 在一天小于等于五节课的前提下, 每周来校的次数越少分值越高, 表 5 给出具体分值: (满分为 10 分)。

表 5 教师少来校分值表					
来校次数	1 次	2 次	3 次	3 次	5 次
得分	10	8	6	4	2

4) 一周中每天的课时平均, 避免某天全天课而某天没课或只有两节课的情况, 若某天的课时大于等于天平均课时的 + 30%, 则该项系数为 - 10。

5) 资源利用率高, 在这里主要指教室资源的利用率, 考虑上课人数与教室资源的比值, 即人数 / 教室容量, 比值越大越好, 最大值为 1。

以上五个优化原则的重要性排序为:

原则 2>原则 1>原则 4>原则 5>原则 3
我校聘请专家给出以上五个原则的权重系数分别为:

$w_2 = 2 \ 8 \ w_1 = 2 \ 6 \ w_4 = 1 \ 9 \ w_5 = 1 \ 7 \ w_3 = 1$
目标函数值得计算公式:
 $d = r_1 * w_1 + r_2 * w_2 + r_3 * w_3 + r_4 * w_4 + r_5 * w_5$
其中 r_1, r_2, r_3, r_4, r_5 为原则 1, 2, 3, 4, 5 的得分值

目标函数值越大则课表编排越优化, 目标函数最大时对应的课表为最终课表。

3.4 算法描述

算法描述, 即排课的基本步骤:

第 1 步: 根据优先原则, 计算每门课的优先次序。有特殊情况者优先级为 1; 公共基础课优先级为 2; 其他合班课的优先级为 3; 对剩下的单班课, 以任课教师任教门数多少为序, 任教门数最多的教师所任的课优先级为 4; 其余依次类推, 优先级为 5, 6, ……。

将计算所得的优先级填入课程信息数据表中, 等待按此优先级来编排。

第 2 步: 排课。设本校可用来上课的教室为 r_1, r_2, \dots, r_m , 共 m 个教室, 周一到周五的上课时间片有 15 个, 则构造 $m * 15$ 矩阵, 矩阵中的元素 $d_{ij} (i = 1, 2, \dots, m; j = 1, 2, \dots, 15)$ 表示教室 r_i 在时间 j 时安排的课程, d_{ij} 为课程代码 + 教师代码的组合字符串, 若教室 r_i 在时间 j 时未安排课程, 则 d_{ij} 为 0。排课发生冲突时, 则采用回溯的方法来解决。

课表编排完毕后, 计算目标函数值, 并在数据库中对课表和目标函数值进行存储。

第 3 步: 按照优先次序再次编排课表, 编排完毕后再次计算目标函数值, 如果该值比第 2 步中所得目标函数值大, 则用 3 中的课表和目标函数值来代替第 2 步的结果; 否则放弃 3 的操作结果。

第 4 步: 执行操作 3, 直到连续三次编排课表所

得的目标函数值都不比原目标函数值大时, 终止操作, 最大目标函数值所对应的课表即为最终课表。

第 5 步: 输出班级、教师、教室课程表。

3.5 算法实现

本算法的后台数据库可采用 Sql Server2000 Access 等, 前台开发工具 VB, PB 等都可以, 在编程实现上比较容易。

4 结语

采用本算法开发高校自动排课系统并进行排课实验, 实验结果表明, 使用该算法编排的课表是成功的, 各种约束条件都得到了满足, 而且时间分布比较均匀, 课表比较科学合理, 可以满足北京联大生化学院的实际需要, 编码简单, 容易实现。可以在情况相似的院校进行推广。

参考文献

[1] 陈谊. 基于优先级自动排课算法 PCSA 的设计与实现方案 [J]. 北京工商大学学报 (自然科学版), 2002 (6): 32~35
[2] 苏仰娜. 基于遗传算法的优化排课系统 [J]. 河南大学学报 (自然科学版), 2005 (01): 75~78
[3] 蒲保兴. 基于遗传算法的排课算法 [J]. 中央民族大学学报 (自然科学版), 2006 2 15(1): 83~87
[4] 张林. 基于蚂蚁算法的学校排课系统 [J]. 安徽电子信息职业技术学院学报, 2004 (21): 90~91
[5] 唐洪英, 周敏. 基于分层分次、贪心算法的排课系统的设计与实现 [J]. 微计算机信息, 2006 (03): 237~240
[6] Shaikh M U, Al-Bastaki Y. Design of an expert system for IT college course timetabling at the University of Bahrain by using a knowledge base process modeling [C]. Information and Communication Technologies From Theory to Applications, 2004 Proceedings. 2004 International Conference 2004: 139~140

(上接第 195 页)

[3] S. Singhal, M. Zyda. Networked Virtual Environments Design and Implementation. Addison Wesley, 1999
[4] J Falby, M. Zyda, D. Pratt, R. Mackey. NPSNET: Hierarchical Data Structures for Real-Time Three-Dimensional Visual Simulation [J]. Computers & Graphics, 1993 17 (1): 65~69
[5] M. Macedonia, M. Zyda, D. Pratt, D. Brutman, and P. Barham. Exploiting Reality with Multicast Groups: A Network Architecture for Large-scale Virtual Environments [C]. Proc. IEEE VRAIS, 1995, 38~45

[6] J. Leigh, A. Johnson, C. Vasilakis, and T. DeFanti. Multi-perspective Collaborative Design in Persistent Networked Virtual Environments [C]. Proc. IEEE VRAIS, 1996, 25~260
[7] C. Greenhalgh and S. Benford. MASSIVE: A Distributed Virtual Reality System Incorporating Spatial Trading [C]. Proc. IDCOS, 1995, 27~34
[8] T. Das, G. Singh, A. Mitchell, P. Kumar, and K. McGhee. NeEffect: A Network Architecture for Large-scale Multi-user Virtual World [C]. Proc. ACM VRST, 1997, 157~163