

摘 要

目前,随着计算机的广泛应用和互联网技术的高速发展,在全国高校中许多教学管理系统相继投入使用。然而,在实际项目的研发中,由于排课问题是一个NP完全问题,开发出符合要求的排课系统是一件难事。

针对排课系统研发和运行中存在的问题,尝试使用贪婪算法去研究和解决问题。贪婪算法是从问题的某一个初始解出发,通过一系列的贪婪选择——当前状态下的最优选择,逐步逼近给定的目标,以尽可能快的求得更好的解。当达到算法中的某一步不能再继续前进时,算法停止。在贪婪算法(greedy method)中采用逐步构造最优解的方法。在每个阶段,都作出一个看上去最优的决策(在一定的标准下)。

基于此,以广东工业大学计算机工程研发中心与深圳大学成人教育学院合作研发的排课系统为背景,在参照了排课算法的大量文献上,结合本系统的实现和现在的运行情况上的不足,根据项目中客户排课的实际需求,基于贪婪算法,以资源匹配为基础,用内存动态分区分配的最佳适应法为依托进行研发,最后设计和实现该排课系统,在实际运行测试中表明,该排课系统的响应时间和排课结果较满意,可以将此推广到其他教学管理系统的开发和实施上。

论文最后对论文本身做了总结,阐述了论文的工作,并指出进一步研究的方向。

关键词 贪婪算法; 教学管理; 排课

ABSTRACT

Nowadays, with widely use of the computer and the quick development of Internet, many teaching management systems have being used in succession in domestic universities. However, in the actual researching project, arranging course is a NP proplem. It is difficult to develop a good arranging course system for user.

Aimed at the problem of reseaching and running of arranging course system, I try to use greedy arithmetic to research and solve the problem. Greedy arithmetic gets going from an initial solution to problem, through a series of greeding chosen which is best chosen at currently status, and it gets the aim step by step, hope to get the better solution as possible as it can. When it reaches a step can not resumption, the arithmetic will stop. Greedy arithmetic want to get the best solution step by step, at every step, it will make a decision that seem to best.

Based that theory, aimed at arranging course system that developed by the center of computer engineering research and development in GuangDong university of technology, refer to the arranging course algorithmic of literature and the actual requirement of user, based the greedy arithmetic and resource matching method, use dynamic memory distributing best adapting method to develop. At last, we had desiged and implement the arranging course system. At the test of running, the response time and the result of system had reached satisfaction, it can spead to development and implement of other teaching management system.

At last, the thesis does the summary to itself, elaborate the work of the thesis, and point out the direction of further research.

Keywords: Greedy arithmetic; Teaching Management; Arranging Course

第一章 绪论

1.1 课题的来源

根据中国教育和科研计算机网统计全国已有七百多所高校接入了国际互联网。伴随着计算机技术在教育领域的广泛运用,高校大规模扩招,数字化校园进程深入发展,高校的传统教学管理模式也面临着重大变革,建立以计算机辅助管理为主要手段,利用现代信息技术和管理方法,对教学管理信息进行采集、分析、处理、存储、传播和反馈已成为各高校实现教学管理现代化工作的重点。

各大高校为了提高整个学校的教学管理水平,纷纷提出教学管理系统的研发需求。

在我们的实验室——广东工业大学计算机工程研究与发展中心,已经开发几个的教学管理系统,比如:广东工业大学成人教育学院学分制教学管理系统、广东工业大学教务处教学管理系统、嘉应大学成人教育学院教学管理系统、广西玉林师范学院成教院教学管理系统、广西师范大学成教院教学管理系统、深圳大学成人教育学院教学管理系统。这些系统在实际运行中取得良好的效果。本文作者从大学本科开始就参与到这些教学管理系统的开发实践中,经历了编码—设计—分析—项目管理等逐步上升的过程,对系统和开发工作越来越熟悉。

对这个课题的研究是因为广东工业大学计算机工程研发中心项目组要为深圳大学成人教育学院开发一个教学管理系统。该项目建设总的目标是:通过教学管理系统,改变原先各部门系统和数据独立的“信息孤岛”现状,将教学管理的各方面工作统一化、信息化,从招生开始,经历学籍管理、教学计划、排课、排考、成绩、毕业以及课室管理、教材管理、经费管理、教师管理等多方面工作,把管理提高到一个新的水平。本文作者紧跟整个项目的每一个环节,参与并负责了其中需求分析、设计和实施等大量实际工作,笔者是这个项目的项目经理和技术骨干,是整个项目的总负责人和系统总的设计者。

鉴于深圳大学成人教育学院排课需求方面的特殊性,如课室资源严重不足,全日制与夜大混全排课等,原先的排课系统不再适用。为了解决此问题,开始了

对排课模块的重新研究和实现。

作者和项目成员首先对项目的需求进行分析,从排课工作人员方面详细了解排课的业务流程以及需要考虑的各种因素,再基于对大量排课算法文献进行研究的基础上进行算法研究和设计,然后进行系统设计和数据库的分析设计,最后编码和系统的现场实施运行。由于排课算法设计和实施的难度较大,笔者是排课系统直接的具体负责人。

然而,排课由于其牵涉到的不同资源如上课的校区、大楼、教室、课程、教师、班级、学生及上课器材等,再加上学校本身的其他相关行政规定所构成的排课规定,可以看出要进行能同时满足这些因素和限制条件的排课业务是非常复杂和困难的。

问题主要集中在两个方面:一方面深圳大学成人教育学院地处繁华地段,课室资源严重不足,全日制与夜大混合排课,按原来的排课算法优化原则可能导致排出的课表不理想。另一方面当前排课算法达到较优解的收敛速度比较慢,用户修改相关资源条件后要等很久才能有结果,有时甚至不能收敛。本文紧密结合该项目的实际设计和运行情况,结合贪婪算法,根据原来系统的设计和运行时的不足之处,提出了基于贪婪算法的排课系统,以资源匹配为基础,用内存动态分区分配的最佳适应法为依托,不采用回溯,并不一味追求结果的最优化,而以实用为目标,解决该项目排课系统实现的难题。

1.2 排课系统发展现状

计算机排课系统的研究是当前各大高校数字化教学改革中面临的一个比较突出的问题。排课是各个大学的教学管理部门的常务性工作。每个学期都是教学管理工作人员最头痛的事情^[5]。现在的排课几乎都是采用人工排课,而人工排课存在效率低、容易产生疏忽、排课结果难以避免冲突、调课难度大等弊端。

在国外,上个世纪 50 年代末就有人研究课表的编排问题,如 1963 年 Gotlieb 曾提出一个课表问题的数学模型^[6]之后课表问题的研究一直都比较活跃,目前主要提出的研究方法有:模拟手工排课法、图论方法、拉格朗日法、二次分配型法等多种方法。国外的研究表明,解决大规模的排课问题单纯依靠数学方法是行不通的。从上个世纪 80 年代以来,国内的教学管理者和计算机工作者在

课表编排方面做了大量的工作,研究了许多实用算法,也取得了一些学术成果^[7]^[8]。具有代表性的计算机排课系统有:南京工学院的 UTSS(A University Timetable Scheduling System)系统、清华大学的 TISER(Timetable Scheduler)系统、大连理工大学的智能教学组织管理及课程调度系统等。但这些系统都模拟它们各自的手工排课,以“班”为单位,依赖于各个大学的传统学年制教学体制,不宜于进行大量推广以及适应现行学分制教学改革的要求^[9]。

1.3 本论文排课系统的解决方案

学分制改革是近年来我国高等学校教育教学改革的一项重要举措。推行学分制,体现了注重素质修养、加强通识教育、优化课程体系,培养复合型人才等特点,对扩大学生学习自主权,实行因材施教,激发教学工作的活力,深化教学改革,提高人才培养质量起着重要的作用。学分制的核心是选课制,选课制的实施使学生变被动学习为主动学习,为学生在选择所修课程、学习方式和方法、授课教师、授课时间及地点等诸多方面提供了广阔空间与自由。这种以人为本、因材施教的教学管理机制有助于调动学生学习的积极性和主动性。选课制的实现前提是与之相适应的合理排课。在高校的教学管理工作中,课表的编排是一项十分复杂、棘手的工作。

本论文为适应学分制的要求,在流程方面采取以下方案:教学计划实施任务——预排课——选课——最终排课。

预排课:对严格按教学计划时间安排进行学习的班级学生(大部分学生都是这样的)进行安排,每门课程的上课人数默认为班级注册人数。这里最关心的是上课时间。

选课:根据预排课的课表结果,学生上网进行选课,避免自身的上课时间冲突。这时学生除可以选自己教学计划那部分的课程,也可以根据自己的学习能力选择其他课程。

最终排课:根据选课的情况确定课程的上课人数然后进行安排,与预排课不同的是,此时课程已经有上课时间的要求,需要根据课程的选课人数适当调整课室,结果包括上课时间和上课地点都要保存到数据库。

如果此时为了达到最终排课结果的优化,采用一般的排课算法,则可能要求

对上课时间重新安排，这可能与在网上学生选课的时间安排存在冲突，会引起教学秩序的混乱。因此在算法方面本文采取以下方案：

(1) 融合Selim^{[1][2][3]}和Loo^{[1][4]}的思路，在对课程科目进行处理时，先对已经指定时间的科目进行安排，然后再对没有指定时间的科目进行处理。

(2) 为了提高排课速度，并不一味追求结果的最优化，而是基于贪婪算法，以资源匹配为基础，用内存动态分区分配的最佳适应法为依托，把教室的可用资源（综合教室可容纳人数以及可用的时间）按照从大到小排成一条链，把申请教室的课程班按照班级人数从多到少排成另一条链。而后取出课程班去申请教室资源，这样可形成大班优先排课的机制，保证大班一定有教室可上课。

1.4 课题研究内容与意义

1.4.1 课题研究内容

本课题的背景是广东工业大学计算机工程研发中心与深圳大学成人教育学院合作研发的教学管理系统，在教学管理系统项目研发和运行的过程中，发现排课问题是比较复杂的。

在开发和运行实践中，存在的问题主要表现在：

一是排课涉及到教室、教师、班级、课程、时间及其他人为要求等多个约束条件，排课方案的可能数目随问题规模的缓慢增长将急速上升，出现“组合爆炸”。

二是当前排课算法达到较优解的收敛速度比较慢，用户修改相关资源条件后要等很久才能有结果，有时甚至不能收敛。

三是深圳大学成人教育学院课室资源严重不足，全日制与夜大混合排课，按原来的排课算法优化原则可能导致排出的课表不理想。

在论文阶段，作者完成的工作是按照导师项目课题组的要求，完成排课系统的系统分析、算法研究和设计、实现和测试，主要内容包括：

1. 在参照了排课算法的大量文献上，根据项目中排课的实际需求，设计基于贪婪算法的排课方案、数据结构、主要算法步骤等。

2. 根据排课的需求，依据排课时的实际考虑因素，确定贪婪算法中的贪婪

准则。

3. 采用三层体系结构，将基于贪婪算法的排课核心程序写成线程的形式成为中间层，由排课主界面程序调用后在后台开辟线程进行运行，不影响其他功能的继续操作。

4. 为了提高排课速度，将数据库中的数据读取到内存中进行处理，算法应用了 C++ STL（标准模板库）中的 `multimap` 关联容器建立两条链表，一条是教室信息链，另一条是课程班信息链，然后进行基于资源匹配的贪婪准则的排课算法的运行。

5. 采用面向对象方法分析设计整个排课系统，用 C++Builder6.0 工具开发实现该排课系统，并在实践中验证。

1.4.2 课题研究意义

课表是教学管理工作开展的基础，排课是教学管理中复杂、庞大、工作量较大的工作任务，而排课问题是一个 NP 完全问题，研究开发一个实用的排课系统具有非常重要的意义：

1. 排课系统得出的课表更加合理，资源得到更充分的利用；
2. 排课系统的运行可以减轻工作人员的排课工作量，提高效率；
3. 排课方案改动时，容易马上产生一个新的课表。

1.4.3 特色之处

本论文的的目的是要实现一个实用的排课系统，让用户能够根据资源情况比较快地排课。具有以下特色：

- 1、利用贪婪算法有效抑制排课中的“组合爆炸”现象。
- 2、通过资源匹配方法实现贪婪策略，符合排课实际要求。
- 3、提高了排课速度。并不一味追求结果的最优化，而是以资源匹配为基础，用内存动态分区分配的最佳适应贪婪算法为依托，不采用回溯，所以排课速度得到提高。

1.5 论文内容组织

贪婪算法是一个应用得比较广泛的算法，本文根据排课系统的实现和现在的运行情况上的不足，提出基于贪婪算法的排课系统，并初步实现。

本文的组织如下：

第一章 绪论 首先阐述了课题的来源，接着分别阐述排课系统的发展现状和本论文排课系统的解决方案，最后指出了课题研究的内容、意义和研究成果。

第二章 教学管理系统介绍，介绍了论文的项目情况，包括项目的项目需求、网络拓扑，体系结构，模块功能，数据库设计和排课的概述，并指出排课系统存在的不足，从而引出本文的论题，为文章后续部分引出基于贪婪算法的排课系统研究和实现的描述作了一定的铺垫。。

第三章 排课系统的相关技术， 综述了排课问题的求解技术，介绍项目中用到的 `multimap` 关联容器技术，最后介绍了贪婪算法，包括主要思想、求解步骤、求解问题的性质和求解例子等。

第四章 基于贪婪算法的排课系统的研究与设计。首先阐述了本排课系统采用贪婪算法的原因，接着对排课的情况进行分析，然后对整个排课系统进行分析设计，包括用例图、体系结构、类、数据库和主要功能等。根据项目的实际情况定出排课中用到的贪婪准则，然后详细地描述排课算法中用到教室和课程班的数据结构，给出了主要的算法。。

第五章 基于贪婪算法的排课系统实现。对各部分功能，包括预排课、最终排课、手动排课、合并班级、删除排课结果和保存排课结果给出详细的实现，还将总控程序、排课、信息结点初始化、分裂教室结点的关键代码都给出来了。

第六章 运行情况分析 & 展望。介绍了运行情况和分析，以及展望和进一步的工作。

最后是论文的结论部分，总结完成的工作，并指出不足和进一步的研究。

第二章 教学管理系统介绍

2.1 项目需求

随着中国教育、科研计算机网络和各校校园网的建设和发展,大部分学校都具备较好的网络通讯环境,网上 MIS 建设已成为各校当务之急。10 多年的经验证明, MIS 建设工作的开展,促进了高校管理体制的改革,管理观念的更新。建立教学管理系统是为了更充分利用学校的人、财、物和信息资源,达到信息集成、高效管理的目标,提高学校的教学质量。各大高校为了提高整个学校的教学管理水平,纷纷提出教学管理系统的研发需求。

为此广东工业大学计算机研发中心已经为多家高校开发了教学管理系统,通过网络的构建,系统的开发,优化了业务流程,将教学管理网络化、规划化,实现科学管理决策,优化信息资源。

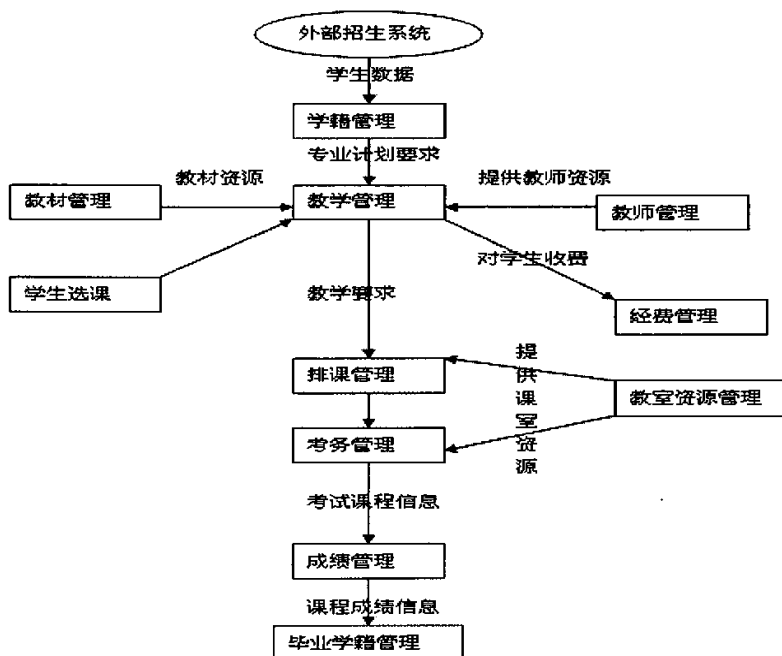


图 2-1 教学管理系统功能流转的业务流程图

Fig 2-1 the business flow diagram of function in teaching management system

图2-1是教学管理系统功能流转的业务流程图,高校的教学管理系统是比较

复杂、庞大的，要从招生数据开始，经历学籍管理、教学计划、选课、排课、排考、成绩、毕业审核等，还需要教材、经费、教室、教师等管理的辅助。

教学管理工作以学生为本，系统以学生为主线，从外部招生系统获得学生数据后，导入系统数据库中，进入学籍管理模块，在学籍管理模块中进行打印录取通知书、新建班级、为学生分班生成学号后进入系统的正式学籍库中。然后根据专业计划要求开展教学管理工作，制定教学计划，形成课程任务，根据课程情况订购教材和确定上课教师，在教材管理模块中开展教材的进销存管理，为广大师生提供便捷的教材服务，在教师管理模块中提供教师的相关信息以便聘用和让学生选课时进行选择。当确定教学计划实施任务后，将课程在网上公布，让学生进行选课。选课模块还提供给工作人员进行集体批量选课、补选课、重修选课等功能。选课结束后，根据实际上课人数和课程的资源要求进行排课，确定上课地点。排课管理模块主要功能是预排课、最终排课、合并班级、手动排课等，与选课结合使用。当学生学习完课程后就要进入考务工作，根据考试课程班和学生进行安排考试，通过导入考试数据、自动排考、手动排考、合并班级等功能，考虑解决各种冲突问题，然后将排考结果在网上公布，让学生及时了解考试安排。在这里排课和排考都需要教室资源管理提供支持，如教室基本信息维护、教室资源预分配、教室需求统计、教室使用情况统计、总课表打印等功能。当学生考试结束后就进入成绩管理模块，在网上开放成绩录入功能，由老师录入课程成绩，经过确认后打印出来交给管理部门，成绩管理模块还提供工作人员对学生成绩维护、审核、查询、统计等功能，学生可以在网上查看相应的成绩。最后，当学生修读完教学计划要求的课程后，就进入毕业审核阶段，根据毕业和学位的条件进行审核，将历年来的课程学习情况和相关的学籍奖惩信息进行审核，给出最后结果，上报毕业数据，导入毕业证书号和学位证书号，供相关部门在网上查询验证。整个教学管理系统紧紧围绕学生的培养，体现以学生为本的特点。

2.2 网络拓扑

系统的总体网络拓扑如图2-2。

在教学管理部门内部建立局域网，设立一个数据库服务器（兼文件服务器），将大量的公用数据进行集中组织、存储、管理、维护，并解决网上数据安全性问

题。同时，研发出在该网络环境下的客户端应用程序，实现数据在各职能科室之间流通的功能，达到共享，交流数据信息的目的。

将部门内部局域网连通到校园网，实现教学管理部门与各院系部等相关部门的交流与共享。通过JSP, java等编程技术，编制基于Web的应用程序，通过B/S/S(Brower/Server/Server)技术为全校提高Web数据库服务，完成相关的数据信息处理功能，如：教师、学生可通过浏览器登录到系统，在规定的时间内进行成绩录入、查询、选课和个人资料查询维护等交互操作；同时支持其他网络服务，如电子邮件、文件传输等。

由学校网络中心以Unix为网络操作系统的服务器直接与交换机相连，提供统一的internet接入功能，通过防火墙实现站点控制，按规则和权限实现资源共享。

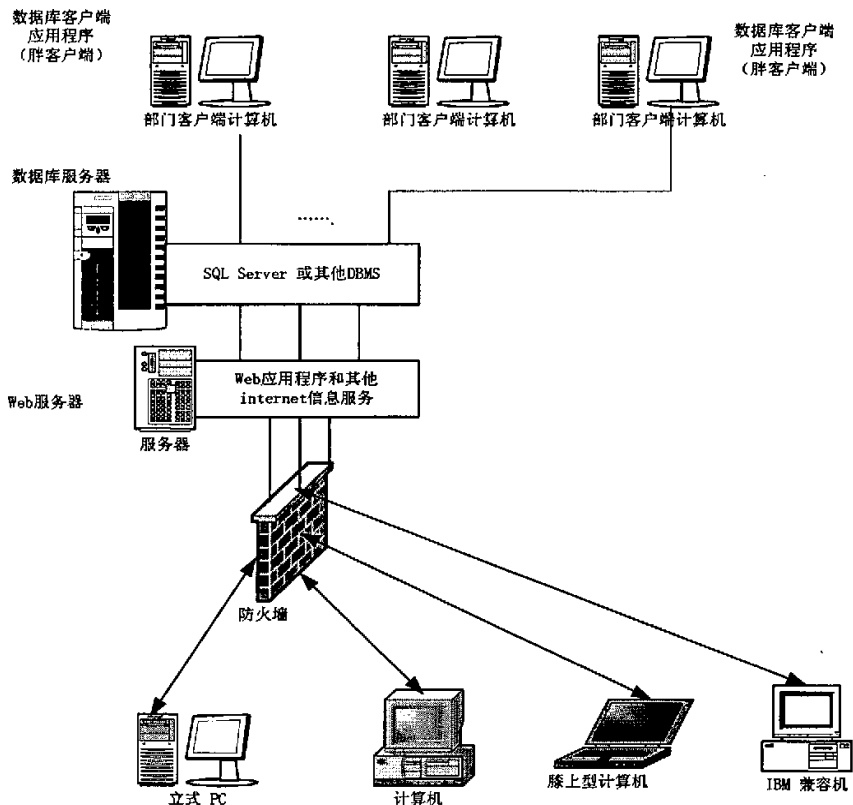


图 2-2 教学管理系统的总体网络拓扑图

Fig 2-2 the network Distribution of Teaching Management System

2.3 教学管理系统的体系结构

按照重在实用,易于扩展的原则,教学管理系统采用三层体系结构,即 C/S/S (客户端/服务器/服务器) 结构,如图 2-3 所示:

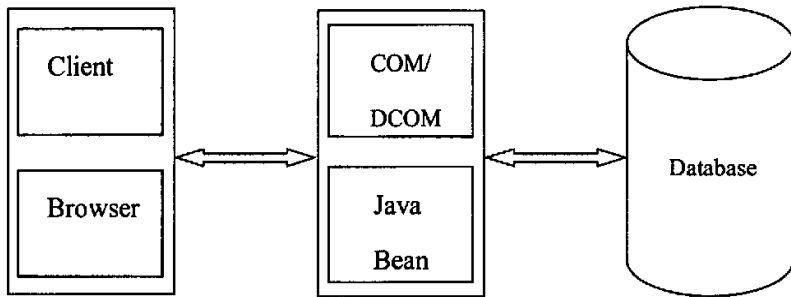


图 2-3 三层体系结构图

Fig 2-3 Three Layer system structure

三层结构的应用程序把业务逻辑独立出来,组成一层或多层。形成客户层界面、中间业务处理层(可由多层组成)和后台数据服务层,保证所有客户数据的一致性和数据的完整性,应用系统开发模式变成:

(1) 客户端人机界面的开发。开发大大简化,只注重人机界面的设计,不必关心业务逻辑和数据库的访问,可以是瘦客户机。提供给用户一个视觉上的界面,通过界面层,用户输入数据、获取数据。

(2) 中间业务逻辑层。逻辑层是界面层和数据层的桥梁,它响应界面层的用户请求,执行任务并从数据层抓取数据,并将必要的数据传送给界面层。

(3) 数据库服务层。数据层定义、维护数据的完整性、安全性,它响应逻辑层的请求,访问数据,被中间业务逻辑层调用完成业务逻辑。这一层通常由大型的数据库服务器实现,如 Oracle、Sybase、MS SQL Server 等。

三层或多层结构,可以将数据处理从客户端转移到应用服务器和数据库服务器上。这样,尽管客户端与应用服务器之间可能存在着多个甚至数百个的连接,但是应用服务器与数据库服务器之间的连接却只有少数几个,从而达到减少通信线路上传递的数据量的目标。这样的功能分配提供了很强的系统可伸缩性,使得在用户数量急剧增加时还能保持系统性能的稳定。使用传统的客户机/服务器模式根本无法胜任上千个客户机同时运行同时需要访问数据库的工作。即使在用户

数量很大的情况下，数据库仍能保持良好的工作负载，保持系统的快速的响应速度。三层结构中，层次的划分不是物理上的划分，而是结构逻辑上的划分，按应用目标划分。

2.4 系统功能模块

教学管理系统的功能结构图

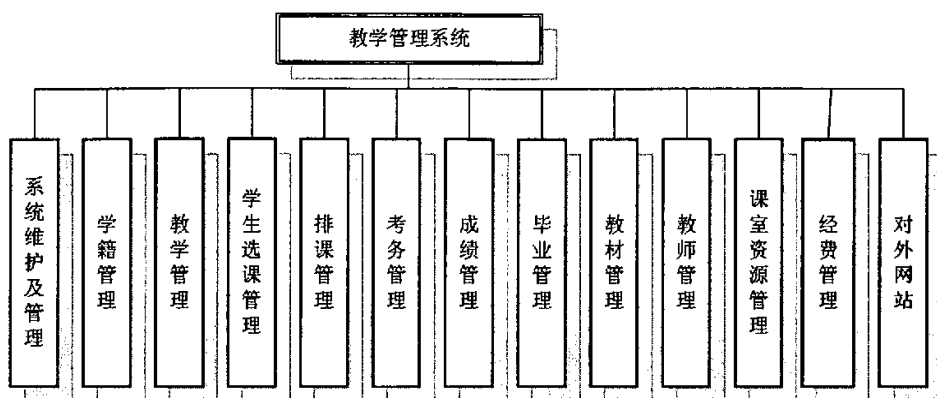


图 2-4 教学管理系统总体功能结构图

Fig 2-4 the chart of function in teaching management system

1 系统维护及管理

该模块是整个教学管理系统的数据库基础，主要功能有：初始化设置，如课程层次、课程性质、学习形式、考试性质、课程类别、课程分类、节次设置、考试类型；基本设置，如学校设置、院系设置、专业信息、课程设置。

2 学籍管理

该模块主要收集并处理学生学籍的原始信息，对学生在校过程中的报到、注册、学籍变动、奖惩等事务进行处理。主要功能包括：新生报到、注册；在校每学期注册登记；学籍变动；奖惩登记。

3 教学管理

该模块是整个系统比较核心的部分，通过该模块可实现教学计划基本信息录入和修改，教学任务的安排等操作。其主要功能包括：教学计划管理（根据教学

表制定教学计划)；教学任务管理。

4 学生选课管理

该模块是整个系统比较重要的部分，需要考虑学生选课时的各种冲突问题，包括学生集体批量选课、个人选课、补选课、重新选课、免修选课等功能。

5 排课管理

该模块是系统中开发和运行比较难的一个系统，需要考虑多方面的因素。该系统功能与选课结合运行，包括预排课、最终排课、手动排课、合并班级等功能。

6 考务管理

该模块主要完成课程的考试安排及相关的查询打印，其主要功能有：场次安排；考试数据的导入；排考；报表打印。

7 成绩管理

该模块主要实现学生成绩的录入、查询、修改、备份等，其功能有：期末成绩录入、查询、修改、统计分析；补考成绩录入、查询、修改、统计分析；成绩备份、导入与导出；报表打印。

8 毕业管理

该模块是教学管理系统流程的最后一阶段，将学生的学习结果进行审核，包括审核条件设置、毕业审核、学位审核、导出毕业数据、导入毕业和学位证书号等。

9 教材管理

该模块实现对教材的进销存管理，主要功能有：教材订购、教材发放、库存管理、经费管理。

10 教师管理

该模块包括教师资料管理和教师工作量统计。教师资料管理包括：院系资料管理、教师基本信息管理。教师工作量统计包括：教师授课信息管理，按系、按教师、按课程统计工作量。

11 课室管理

该模块对教学用的课室进行综合考虑维护，确保排课排考的成功，并对课室使用情况有及时清晰的掌握。主要功能有：课室基本信息维护，可以增删改、设置课室功能要求、教学中对课室的需求汇总打印、特殊情况申请、课室资源管理、课室使用情况查询、课程表打印。

12 经费管理

该模块实现对学生缴费的结算，包括如下功能：与财务接口（经费收取）；经费查询；经费结算。

13 对外网站

该模块完成教学基本信息的发布、查询，还提供其他的功能，如网上报名、与学生交互等。具体包括：网上选课、学生成绩录入（教师使用）、教学调查、通知、办证查询、考试、补考、统考等时间、地点查询、成绩查询、学籍情况查询、问题登记、经费查询。

2.5 数据库设计

根据数据设计规范和实际应用的反规范化要求，将数据库表设计成两部分：一部分为码表，这部分的表中的数据是静态的，也就是说基本没有数据更新操作，一般只有查询操作，主要包括学习形式、课程层次、课程类别、课程特征、课程性质、课程科类、考试形式、考试类别、考核方式、考试情况、模块名称、办学地等，如图2-5所示：

学习形式 fForm_No fFName	课程层次 fCLevel_No fCLName	课程类别 fCForm_No fCFName
课程特征 fCCharacter_No fCCharacter	课程性质 fCKind_No fCKName	课程科类 fCSort_No fCSName
考试形式 fEKind_No fEKName	考试类别 fESort_No fESName	考核方式 fEMode_No fEMName
考试情况 fCheat_No fCheatName	模块名称 fModule_No fMName	办学地 fSchool_No fSName

图 2-5 部分主要静态数据表

Fig 2-5 some main static database table

另一部分为数据表，这部分的表中的数据是动态的，也就是说数据更新操作频繁，数据量也比较大，是数据库性能的核心部分，主要包括班级、学生基本信息、教学计划实施、教室信息、教室使用信息、选课结果等等，如图2-6所示。

学生基本信息	教学计划实施	教室信息
fStudentNum fExamNum fClassNum fAcademy_No fLForm_No fLLevel_No fName fIdCardNum fPass fSex fBirth fNation fAddress fTelep ... fPost	fId fClass_No fCourse_No fAcademy_No fForm_No fLevel_No fTermNo fTime fPlace fCombine fTotalStudent fGrade fAddress fIfAutoarrange ... fFunctionId	fId fSchool_No fBuildNo fAcademy_No fRoom_No fTotalSeat fTermNo fNote fRoomName fFloorNo fInternativeseats ... fFunctionId
班级	选课结果	教室使用信息
fClassNum fForm_No fLevel_No fAcademy_No fProfession_No fAcademy_No fSchool_No fGrade fBanZhuRen fBZRTel fYearBegin fYearEnd fYear fStudentCount ... fClassName	fId fStudentNum fCourse_No fAcademy_No fTermNo fESort_No fCKind_No fScore1 fScore2 fScore3 fCheat fSelectTime fScore1_ps_wjz ... fRePlan	fClassroomId fTermNo fWeek fAcademy_No fDayInfo fDateInfo fPeriod fArrangeItemId fArrangeTableId ... fArrange

图 2-6 部分主要动态数据表

Fig 2-6 some main dynamic database table

2.6 排课概述

排课开始于教学计划的编制。即教学管理部门必须在每学期前收集下学期的开课信息,然后统计下学期可用的教学资源,经过协调编制下学期的开课任务书。开课任务书应该包括全校每个班级的教学计划实施表,而每个教学计划实施表的记录都应该包含校区、班级、教师、课程、课程类型、开课学院、人数、需求教室资源类型、有特殊要求时的上课方式、教师期望时间等完备信息。

教学计划的实施的关键是课表的编排,在编排的过程中必须遵守多个原则:

- 1 在同一时间同一学生不能上两门不同的课程;
- 2 在同一时间同一教师不能给两门不同的课程上课;
- 3 在同一时间同一教室不能安排两门不同的课程;
- 4 每门课程的教室都有自己特定的类型;
- 5 教室必须足够大,能够容纳上课的学生;

- 6 教师、学生不同校区上课时要留一定的时间用于赶赴；
- 7 体育课需安排在特定的时间，且同一时段内之后不能再安排课程；
- 8 实验课、实习课等课程有自己的安排方式等。

而且一般还有以下多个目标：

- 1 一个班级时间安排在一周内尽量分布均匀；
- 2 尽量满足教师上课时间的期望；
- 3 教师对时间安排在课表上的密度有一定的喜好；
- 4 教师和班级相邻两次上课地点尽量接近。等等

现在大多数院校的排课方式是手工编排，主要通过人智能的判断和协调完成的。手工编排方式往往开始于一个学期数月前，在实际的安排过程中，教师数量成千，学生数目上万，教师跨院上课和班级交叉上课众多，……，而且在计划安排完毕之后，往往由于频繁的变动不得及时调整。诸如此类因素，使得排课工作不堪重负，工作结果也不尽如人意。

计算机排课，是把排课问题化为计算领域的有约束的时空组合优化问题进行求解的。它对课表上的时间进行了分片和编号处理，使每个时间分片和教室空间组合，构建了一个个大小不等的时空组合块，并根据求解规则，对每个开课计划进行时空组合块分配，而且分配的组合，即安排方案，必须在目标空间中表现良好的人为满意度。这种人为满意度往往不仅多个，而且是模糊的。虽然利用计算机来模拟手工排课，可以抽象问题中的各个要素，数学表达各种约束条件，并根据课表的组织形式和规律，缩减了问题空间的搜索范围，有效组织了排课知识，在一定程度上呈现智能化。但由于问题本身的求解规模过于庞大，各要素间的关联层出不穷，以及人们对课表评定的准则不同，使计算机在求解排课问题的过程中，面对难以穷尽的组合和多个模糊的目标的优化，也表现得无能为力。

就本质而言，排课问题是一个非线性的、有约束的、模糊多目标优化的、难解的、时空组合的数学问题。即在满足各种已知约束条件下找到一组较优的时空组合，同时它还受到客观物质条件、教学组织形式和求解目标等多种因素的相互影响，使这一问题在实际解决时呈现出受具体条件制约的特点。

根据深圳大学成人教育学院的排课要求，采用手动排课和自动排课相结合的排课方案，为了适应学分制的要求，先预排课，确定上课时间，然后在网上公布课程让学生进行选课，当选课完确定上课人数后再最终排课，调整上课地点，让人数多的在大课室上课。其基本流程如图 2-7。

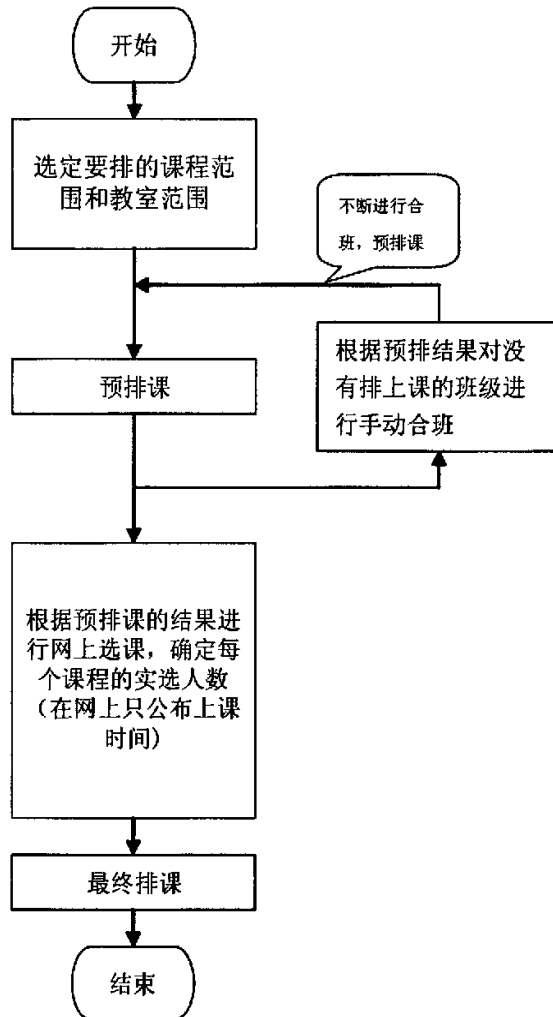


图 2-7 排课业务流程图

Fig 2-7 The operation flow diagram of arranging course

为了满足排课工作人员提出的灵活方便的要求, 本排课系统并不总是对全校的课程进行安排, 它可以根据条件查询出课程范围和教室范围, 然后在此范围内进行自动排课或手动排课。由于课室资源紧张的现状, 可以针对排课结果的情况进行合并, 将没有排上的课程班合并到其他课程班, 一直到相关课程班都有地方上课为止。

2.7 系统的运行情况和问题及论文研究对象的导出

教学管理系统已于2004年12月份投入正式的运行。系统是用C++BUILDER 6开发工具开发。所有子系统都是由图形化，形象化的友好界面组成。数据库有100多个数据表组成，大量使用了触发器和存储过程技术。各个子系统的数据交换通讯通过数据表来实现的。系统设计主要使用了面向对象的设计方式，由不同的类和对象来实现。教学管理以招生到学生毕业为主线，包括了学生注册、学籍管理、教学计划制定、选课、排课、排考，成绩管理、毕业审核等，为了辅助主线的运行，还有教室资源管理、教材管理、经费管理、教师管理等模块。每个系统的登陆都有严格的权限定义。

下面分别是几个程序的界面：

(1) 学籍管理程序

学号	班代码	班级名称	姓名	性别	政治面貌	学籍状况	是否锁库	毕
20031032121701014	200501091701201	2005专科土木工程(1)	江科技	男	待定	复学		
20031032121941041	2005010111941201	2005专科电子商务(1)	汪露	女		复学		
20031032121961021	200501091961201	2005专科计算机网络(1)	周勇	男		复学		
20031032123061084	200501023061202	2005专科国际贸易(2)	方利华	女		复学		
20031032123081518	200501993081205	2005专科会计学(5)	林丽萍	女	党员	复学		
20031032123082134	200501013131202	2005专科行政管理(2)	罗静	女		复学		
20031032123181031	200501996032201	2005专科外贸英语(脱产)(1)	刘敏楠	女		复学		
20031032123201128	200501023201202	2005专科物流管理(2)	谢冬竹	女		复学		
20031032123332074	200501993332202	2005专科工商管理(脱产)(2)	范文英	女		复学		
20031032125081010	200501055081201	2005专科涉外文秘与公共关系(1)	李咏梅	女		复学		
20031032129041188	200501039041203	2005专科英语(3)	戴莉莉	女		复学		
20031032133081093	200501993081303	2005专升本会计学(3)	张一玲	女		复学		
20031032133331070	200501993331302	2005专升本工商管理(2)	袁玲楠	女		复学		
20031032134021068	200501044021302	2005专升本法学(2)	赵彤	女		复学		
20031032139041080	200501999041304	2005专升本英语(4)	冯凌	女		复学		
20051032121091001	200501011091201	2005专科财务管理(1)	廖丽文	女		正常考入		

学生数: 4570

图 2-8 学籍管理程序

Fig 2-8 the program of student management

(2) 成绩录入程序

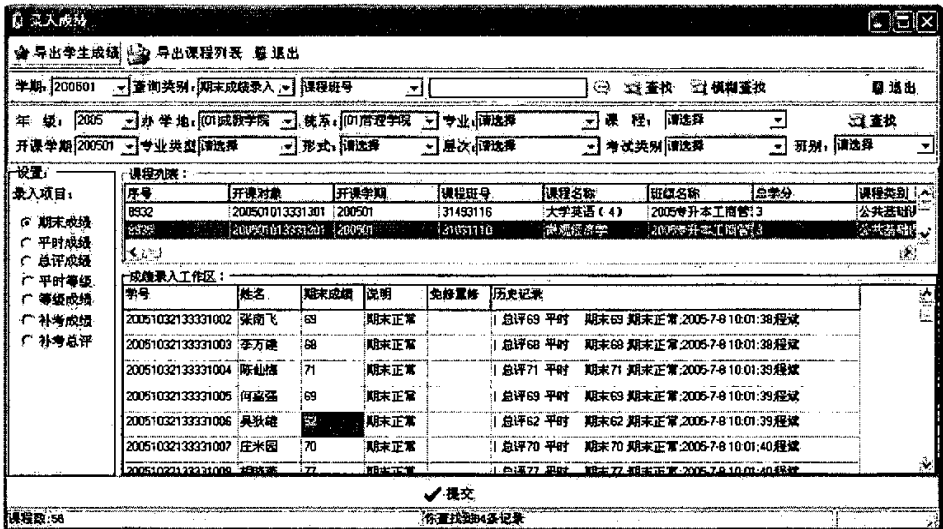


图 2-9 成绩录入程序

Fig 2-9 the program of inputing scores

(3) 安排教室程序

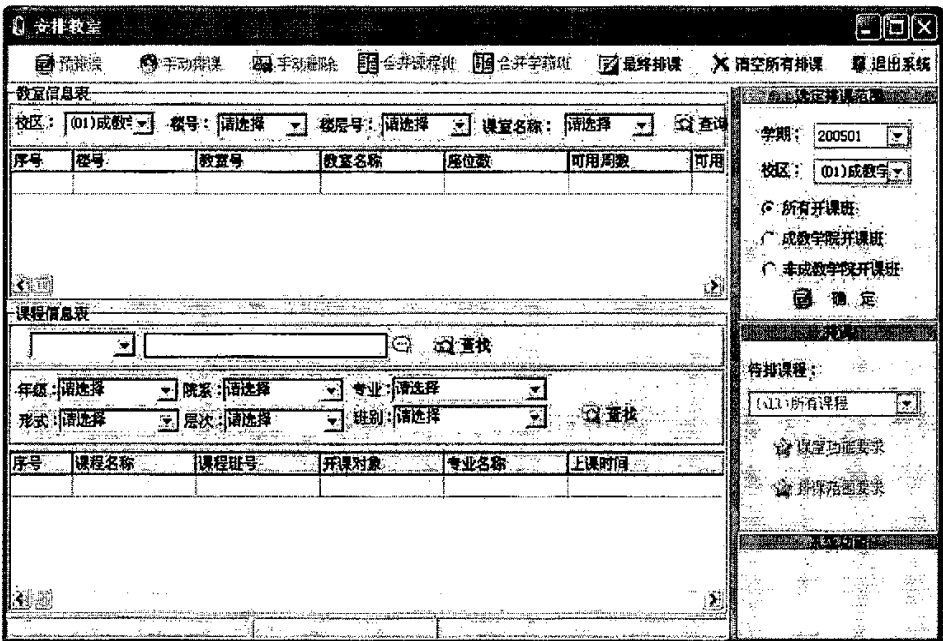


图 2-10 安排教室程序

Fig 2-10 the program of arranging classroom

该系统现在深圳大学成人教育学院正式运行中基本满足教学管理的要求，但排课系统仍然存在不足之处：

- 1 深圳大学成人教育学院课室资源严重不足，全日制与夜大混全排课，按原来的排课算法优化原则导致排出的课表不理想。
- 2 排课系统运行时算法收敛速度慢，要等很长一段时间才能得到结果，用户不满意这种速度。

针对排课系统的不足，作者再次深入调研和对大量文献的研究，为了满足用户排课的需求，采用基于贪婪算法的排课系统解决方案。

针对现有系统存在的问题，本文将从以下方面进行深入的研究与开发：

- 1 基于考虑课室资源不足，全日制与夜大混全排课又有各自的要求，将采用贪婪算法进行资源匹配的排课算法设计。

- 2 针对排课系统运行速度慢问题，将数据库中的数据读取到内存中进行处理，算法应用了 C++ STL（标准模板库）中的 `multimap` 关联容器建立链表保存和处理算法过程中数据。

- 3 在文章的第四部分和第五部分，针对排课系统复杂的需求，本文设计了基于贪婪算法的排课程序，用面向对象方法设计实现该排课系统

本论文排课系统并不一味追求结果的最优化，而是基于贪婪算法，以资源匹配为基础，用内存动态分区分配的最佳适应法为依托，不采用回溯，大大提高排课速度，同时使代码达到最大程度的重用，使深圳大学成人教育学院的排课系统面对复杂的业务过程能做到灵活性强，同时也利于日后维护和升级。

2.8 本章小结

本章比较全面地介绍了教学管理系统，包括项目需求、网络拓扑、体系结构、功能模块等，最后针对目前系统的运行情况，指出了排课系统上的不足之处，引出本文的论题，提出利用基于贪婪算法的资源匹配的内存动态分区分配的最佳适应法对原来系统给出改进的方案。

第三章 排课系统的相关技术

上一章根据系统运行情况和问题,导出了论文的研究对象,这一章将综述排课问题的现有求解技术,并介绍将在本论文排课系统用到的相关技术。

3.1 排课问题的求解技术综述

从 20 世纪 50 年代末,国外就有人研究课表的编排问题。1963 年 Gotlieb 在他的文章^[10]中提出排课问题的数学模型,它标志着排课问题的研究正式跨入了庄严的科学殿堂。但是后来在实践中遇到困难,人们对排课问题的解是否存在产生了疑问。1976 年, S.Even 在论文^[11]中, 1995 年, Cooper 等人在^[12]中,证明了排课问题是 NP 完全的,这回答了排课在实践中遇到困难的原因。但是计算机难解性的理论研究指出,现代计算机尚未找到解决 NP 完全问题的多项式算法,相当于计算机解决排课问题无法实现。此后对这一问题的研究大多离开理论研讨的轨道而转向经验方式,这使 20 世纪 80 年代的许多排课系统缺乏普适性。

S.Even 的论证方式正式确立了排课问题的学术地位,把人们对课表编排复杂性的认识提高到理论的高度。

从 1963 年 Gotlieb 提出排课问题的数学模型^[10]之后,学者们又对排课问题的算法作了许多探索,但是由于排课问题是 NP 完全问题,并且受实际问题边界的影响,大多数求解结果都不理想。

Ferland 等人^[13]和吴金荣^[14]把排课问题化成整数规划来解决,但计算量很大,其仅仅适用于规模很小的课表编排。还有何永太^[15]和胡顺仁^[16]等人试图用图论中的染色问题来求解排课问题,可惜图的染色问题本身也是 NP 完全问题。由于问题的复杂性,许多文章^{[17][18][19][20][21][22][23][24][25]}利用启发式函数来解决排课问题,大多数启发方法都是模拟手工排课来实现的。由于实际的排课问题存在各种各样的限制条件与特殊要求,对这些因素处理的好坏就显得很重要。

20 世纪 90 年代时,国外对排课问题的研究仍然活跃。如印度的 Vastapur 大学管理学院的 Arabinda Tripathy、加拿大 Montreal 大学的 Jean Aubin 和 Jacques A. Ferland 以及 Charles Fleurent 等^[26]。Arabinda Tripathy 的工作是

针对以“人”为单位进行课表安排的。他运用拉格朗日松弛法和分支定界技术求解, 这种方法的缺点是为了减少变量的个数, 人为造成科目间的冲突。Arabinda Tripathy 还研究了研究生课表的编排问题, 他采用多重课组的方法来处理冲突 (根据学生选课的情况, 将人数多的课程在一星期内开多次)。Jacques A. Ferland 等人则把问题分为两个子问题: 时间表问题和分组问题。在时间表问题中, 根据学生、教师和教室的资源情况形成一个时间表。对于选课人数较多的课程, 一个星期要分成几个时间段来上, 分组问题就是将学生分给时间段。两个问题相关联, 通过惩罚因子来构造启发函数。它们研制的 SAPHIR 课程调度决策支持系统分为数据处理、自动优化、交互优化等几个模块。该系统解决矛盾问题的主要方法也是采用多重课组, 这与西方的教学管理体制是不可分的。

另外一批学者还将模拟退火法应用到排课问题的研究中^{[17][18][27][28]}。模拟退火法 (Simulated Annealing) 是 Kirkpatrick 等人于 1983 年首先提出的^[29], 它是人们从自然界固体退火过程中得到启发并从中抽象出来的一种随机优化算法。模拟退火法用于求解优化问题的出发点是基于物理中固体物质的退火过程与一般优化问题的相似性。在对固体物质进行退火处理时, 常先将它加热使其粒子可自由运动, 以后随着温度的下降, 粒子逐渐形成低能态晶格。若在凝点附近的温度下降速率足够慢, 则固体物质会形成最低能量的基态, 优化问题也存在类似过程。模拟退火法被用来解决许多实际应用中优化的问题, 取得了不错的效果, 但用来解决排课问题, 现在还处于模型试验阶段, 还有许多问题需要研究解决。

基于遗传算法的排课^{[30][31][32][33][34][35][36]}, 借鉴于生物界进化思想和遗传机制, 从潜在的解集中的一个种群 (population) 开始, 在每一代中, 根据问题域中个体的适应度 (fitness) 大小挑选个体, 经过基因 (Gene) 编码 (Coding) 的一定数目的个体 (Individual) 进行组合交叉 (Crossover) 和变异 (Mutation), 产生代表新的解集的种群, 末代种群中最优个体经过解码 (Decoding), 可以作为问题的近似解。

基于时间位图迭加匹配的算法^[19]定义了教学过程中的时间位图、课时模式和时间匹配等概念, 将排课问题转化为基于时间位图迭加匹配的课时模式查找问题, 给出了相应的抽象具体排课过程, 并对与排课过程中可能出现的冲突, 探讨了三种回溯消除冲突方式。

基于优先级自动排课算法^[20]利用了运筹学中分层规划的思想, 把排课问题在

数学上看作一个在时间、教师、学生和教室的四维空间，以教学计划和各种特殊要求为约束条件的组合规划问题，采用了化整为零的思想及提出了优先级的概念，有效地抽象了实际排课情况，缩小了求解问题的空间。

基于专家系统的求解算法^{[37][38][39]}将专家系统知识引入排课问题的求解中，有效组织排课过程中的知识，使各种排课逻辑从程序中解放出来，能够便于各种排课经验的积累，使排课结果更加符合实际情况。

更多算法^{[21][22][23][24][25][40]}已经提出，它们都是一定程度上启发式搜索求解方法，对后继者有一定的借鉴意义。

3.2 multimap 关联容器

在排课过程中，为了提高排课速度，数据库中的数据要读取到内存中进行处理，因此需要建立一定的数据结构来存储这些数据。本排课算法会建立两条链表，一条是教室信息链，另一条是课程班信息链。需要说明的是，在建立链表时，本算法应用了 C++ STL（标准模板库）中的 multimap 关联容器。

在 multimap 中，我们提供一个“键/值”对：键用来索引 multimap，而值用作被存储和检索的数据，而且顾名思义，multi 的意思就是允许键重复。每个插入 multimap 的元素都将按照键的顺序排序，这个排序规则可以自定义，并作为第三个参数传给 multimap，本算法中的 CompareRoom 就是为这个目的而定义的一类。

插入

假设需要开发一个 DNS 后台程序（也就是 Windows 系统中的服务程序），该程序将 IP 地址映射匹配的 URL 串。知道在某些情况下，相同的 IP 地址要被关联到多个 URLs。这些 URLs 全都指向相同的站点。在这种情况下，你应该使用 multimap，而不是 map。例如：

```
#include <map>
```



```
#include <string>

multimap <string, string> DNS_daemon;
```

用 `insert()` 成员函数而不是下标操作符来插入元素。`insert()` 有一个 `pair` 类型的参数。可以象下面这样使用它：

```
DNS_daemon.insert(make_pair("213.108.96.7", "cppzone.com"));
```

在上面的 `insert()` 调用中, 串 “213.108.96.7” 是键, “cppzone.com” 是其关联的值。以后插入的是相同的键, 不同的关联值:

```
DNS_daemon.insert(make_pair("213.108.96.7", "cpluspluszone.com"))
```

因此, `DNS_daemon` 包含两个用相同键值的元素。注意 `multimap::insert()` 和 `map::insert()` 返回的值是不同的。

```
typedef pair <const Key, T> value_type;

iterator

insert(const value_type&); // #1 multimap

pair <iterator, bool>

insert(const value_type&); // #2 map
```

`multimap::insert()` 成员函数返回指向新插入元素的迭代指针, 也就是 `iterator` (`multimap::insert()` 总是能执行成功)。但是 `map::insert()` 返回 `pair<iterator, bool>`, 此处 `bool` 值表示插入操作是否成功。

查找单个值

与 `map` 类似, `multimap` 具备两个版本重载的 `find()` 成员函数:

```
iterator find(const key_type& k);

const_iterator find(const key_type& k). const;
```

`find(k)` 返回指向第一个与键 `k` 匹配的 `pair` 的迭代指针, 这就是说, 当你想要检查是否存在至少一个与该键关联的值时, 或者只需第一个匹配时, 这个

函数最有用。例如：

```
typedef multimap <string, string> mmss;

void func(const mmss & dns)
{
    mmss::const_iterator cit=dns.find("213.108.96.7");

    if (cit != dns.end())
        cout <<"213.108.96.7 found" <<endl;

    else
        cout <<"not found" <<endl;
}
```

处理多个关联值

count(k) 成员函数返回与给定键关联的值得数量。下面的例子报告了有多少个与键 “213.108.96.7” 关联的值：

```
cout<<dns.count("213.108.96.7") //output: 2
<<" elements associated"<<endl;
```

为了存取 multimap 中的多个值，使用 equal_range()、lower_bound() 和 upper_bound() 成员函数：

equal_range(k)：该函数查找所有与 k 关联的值。返回迭代指针的 pair，它标记开始和结束范围。下面的例子显示所有与键 “213.108.96.7” 关联的值：

```
typedef multimap <string, string>::const_iterator CIT;
typedef pair<CIT, CIT> Range;

Range range=dns.equal_range("213.108.96.7");

for(CIT i=range.first; i!=range.second; ++i)

cout << i->second << endl; //output: cpluspluszone.com
```

```
// cppzone.com
```

`lower_bound()` 和 `upper_bound()`: `lower_bound(k)` 查找第一个与键 `k` 关联的值, 而 `upper_bound(k)` 是查找第一个键值比 `k` 大的元素。下面的例子示范用 `upper_bound()` 来定位第一个其键值大于“213.108.96.7”的元素。通常, 当键是一个字符串时, 会有一个词典编纂比较:

```
dns.insert(make_pair("219.108.96.70", "pythonzone.com"));
CIT cit=dns.upper_bound("213.108.96.7");
if (cit!=dns.end()) //found anything?
    cout<<cit->second<<endl; //display: pythonzone.com
```

如果你想显示其后所有的值, 可以用下面这样的循环:

```
// 插入有相同键的多个值
```

```
dns.insert(make_pair("219.108.96.70", "pythonzone.com"));
dns.insert(make_pair("219.108.96.70", "python-zone.com"));
// 获得第一个值的迭代指针
CIT cit=dns.upper_bound("213.108.96.7");
// 输出: pythonzone.com, python-zone.com
while(cit!=dns.end())
{
    cout<<cit->second<<endl;
    ++cit;
}
```

有关 `multimap` 更详细的资料可在 MSDN 上找到。

3.3 贪婪算法介绍

在本论文排课系统中，根据实际情况采用贪婪算法。贪婪法不追求最优解，不要回溯，只希望得到较为满意的解。虽然贪婪法不是对所有问题都能得到整体最优解，但对范围相当广泛的求最优解问题来说，它是一种最直接的算法设计技术，通过一系列局部最优的选择，即贪婪选择可以产生整体最优解。具体地说，通常所求问题的一个整体最优解，是从贪婪选择开始的，而且每作一步贪婪选择后，原问题可简化为一个规模更小的类似子问题，然后通过多步贪婪选择，最终可得到问题的一个整体最优解。

3.3.1 贪婪算法主要思想

从问题的某一个初始解出发，通过一系列的贪婪选择——当前状态下的最优选择，逐步逼近给定的目标，以尽可能快的求得更好的解。当达到算法中的某一步不能再继续前进时，算法停止。

在贪婪算法（greedy method）中采用逐步构造最优解的方法。在每个阶段，都作出一个看上去最优的决策（在一定的标准下）。决策一旦作出，就不可再更改。作出贪婪决策的依据称为贪婪准则（greedy criterion）。

3.3.2 贪婪算法的求解步骤

从问题的某一初始解出发；
while 依据贪婪策略朝给定目标前进一步 do
 求出可行解的一个解元素；
由所有解元素组合成问题的一个可行解。

3.3.3 贪婪算法求解问题性质

贪婪选择性质：可通过局部最优（贪婪）选择达到全局最优解；

- 通常以自顶向下的方式进行，每次选择后将问题转化为规模更小的子

问题:

— 该性质是贪婪法使用成功的保障, 否则得到的是近优解;

最优子结构性质: 问题的最优解包含它的子问题的最优解;

— 并不是具有最优子结构性质的问题都可以采用贪婪策略;

— 往往利用最优子结构性质来证明贪婪选择性质;

3.3.4 贪婪算法的求解例子

这里以最优装载为例子说明贪婪算法,

```
void ContainerLoading(int x[], float w[], float c, int n)
{ //x[i]=1 当且仅当货箱 i 被装载, 对重量按间接寻址方式排序
  int *t = new int [n+1]; //t 是间接寻址表
  IndirectSort(w, t, n); //此时, w[t[i]] ≤ w[t[i+1]], 1 ≤ i < n
  for(int i = 1; i <= n; i++) //初始化 x
    x[i] = 0;
  for(i = 1; i <= n && w[t[i]] <= c; i++) { //按重量次序选择物品
    x[t[i]] = 1;
    c -= w[t[i]];
  } // 剩余容量
  delete t[];
}
```

3.4 本章小结

该章节讨论了排课问题的求解技术, 介绍了排课系统中用到的 multimap 关联容器和贪婪算法。

第四章 基于贪婪算法的排课系统的研究与设计

4.1 本排课系统采用贪婪算法的原因

上一章所述针对排课问题的求解技术对后继者有一定的借鉴意义,但它存在着不足之处:

1 对于启发式搜索来说,其启发信息依赖于实际情况,排课问题的求解只能针对个别的实际问题,也没有形成一种通用的有效排课方法。

2 对于遗传算法来说,问题的编码往往不能很好地反映针对更多实际的信息,而且操作很不方便;对求解问题的边界定义时往往考虑排课的一般情况,忽略更多的约束条件,求解规模普遍偏小,实用程度较低,很难在学校中开展使用。

3 引入专家系统技术的排课方法,虽然可以对排课规则知识有效组织,但由于排课过程中各要素关联规则难以提取,并且实际求解效果也不理想。

对于排课系统的研发,林漳希和林尧瑞 1984 年在^[43]上发表了该课题上的实验性研究成果。成形的系统有大连理工大学 1998 年推出的教学调度系统 3.00 版本^[44]和清华大学计算机与信息管理中心开发的综合教务管理系统^[45],但是,

1 国内的排课软件很少,涉及到自动排课算法的系统更少,大部分局限于辅助人工排课。

2 自动排课系统往往由于在随机求解过程中出现太多的未被安排课程而很难在实际中使用。

综上所述可知,目前针对排课问题还没有非常成熟的算法和系统,在排课系统实际的开发和运行中,为了取得较满意的效果,可以考虑采用贪婪算法。

贪婪法不追求最优解,不要回溯,只希望得到较为满意的解。虽然贪婪法不是对所有问题都能得到整体最优解,但对范围相当广泛的求最优解问题来说,它是一种最直接的算法设计技术,通过一系列局部最优的选择,即贪婪选择可以产生整体最优解。具体地说,通常所求问题的一个整体最优解,是从贪婪选择开始的,而且每作一步贪婪选择后,原问题可简化为一个规模更小的类似子问题,

然后通过多步贪婪选择, 最终可得到问题的一个整体最优解。

4.2 排课情况分析

4.2.1 排课的要素

从排课过程可能引起潜在冲突的角度, 结合深圳大学成人教育学院的实际情况, 可以将排课问题涉及到的因素考虑如下:

时间:在排课问题中涉及时间的概念有学年、学期、周、天、时段。上课的时间一般是按周来计算的(指周课表), 从开学第一周到第 N 周, 每周要上的课一般来说是一样的, 但也有可能存在不一样的周, 如单双周或者不是从第一周开始的情况。每周上课时间为 M 天($M \leq 7$), 每天可分三个时间段(上午、下午、晚上), 每个时间段可分为 P_i 节, 用户中实际要求 $P_i = 3$ 。一节就是一个课时, 是上课的最小单位。根据客户实际情况, 一般一门课程是按时段的形式来组织的。在一周内开多次的课程应选择隔天。

课程:每个课程都有自己的编号、名称、开课学院和学时要求等。每门课程可以有多个班级合并上课, 合班班级往往是同专业或相似专业, 但有时候是跨院系的。每门课程都有指定的课室功能类型, 如普通课室、多媒体课室、语音室、机房等。某些课程由于上课班级较多难以协调或照顾教师要求等, 可以预先给定时间或教室。

教室:每个教室都有编号、门牌号码和名称等, 每个教室在同一时间只能接纳一门课程的授课, 并且教室容量应该大于等于上课的人数。当上课人数远远小于教室容量也是不合适的。

班级:每个班级都有编号和名称。每个班级同一时间只能上一门课程。

教师:每个教师都有编号和名称。每个教师同一时间只能上一门课。每个教师可以有自己期望的授课时间。

4.2.2 排课过程的约束条件

排课是将教师和学生时间和空间上根据不同的约束条件进行排列组合,

以使教学正常进行。这里的约束条件主要为避免冲突,冲突所包含的内容很广泛,几乎发生在所有两个或多个排课因素之间。而避免冲突是排课中要解决的核心问题。只有在满足全部约束条件和避免所有冲突的基础上,才能保证整个教学计划合理正常进行。而对教师、学生、教室和时间等几部份资源进行最优化组合配置,才能保证充分发挥各类资源的优势和提高教学质量。

可以将排课过程的约束条件分为三类:基本硬约束、硬约束和软约束。其中基本硬约束是指教师、学生和教室在时空概念上发生了不可能发生的事情,它是排课过程中最基本的约束条件;硬约束是由学校的实际情况,排课时必须遵循的原则,否则将会导致排课结果无意义。软约束是指排课过程中满足更佳,但不满足也无妨的约束条件,它们的违反与否往往与排课实际情况相关。在三类约束中,前两者是衡量排课方案是否切实可行的标准,软约束是衡量排课方案优劣的准则。

可以把排课过程中常见的约束条件分类如下,这些约束条件比较符合排课过程的实际情况。

表 4-1 排课约束条件

Table 4-1 the restriction of arranging course

基本硬约束 (Base hard restriction)	
B1	同一学生不能同时上两门或两门以上的不同课程
B2	同一教师不能同时给两门或两门以上的不同课程授课
B3	同一教室不能同时安排两门或两门以上的不同课程
硬约束 (Hard restriction)	
H1	教室必须足够大,能容纳上课的学生
H2	每门课程都需要特定功能类型的教室
H3	某些课程可以先行手工排定上课时间和教室
H4	上课按时段的形式进行组织
H5	课程的学时分配有一定的规律
H6	每门课程都要求安排在一定的预分配区域内
软约束 (Soft restriction)	
S1	班级课程表在一周内尽量分布均匀

S2	教师对上课时间存在一定的偏好
S3	全日制学生的课程安排在白天，夜大学生的课程安排在晚上，星期六、日尽量不排课

4.2.3 排课目标

排课问题实质上是一个多目标的组合规划问题，对组合规划问题如果想找到最优解，必须有足够的约束条件来实现，可是对于部分属于人文范畴的排课问题，不可能找到充足的约束条件。而且，在组合规划问题中，方案的可能数目随问题规模的缓慢增长将急速上升，这种空间向量稍有增加，就引起课表编排方案数量急剧增加的现象，称为“组合爆炸”。对于此类问题的求解，往往在理论上是可行的，但即使放在当今最先进的计算机上，也要运行上百年，而此类问题的求解方法对时间情况而言，将毫无意义。因此，如何从实际情况出发，采取适当的措施，抑制“组合爆炸”，缩小搜索空间，成为排课问题中的一个关键问题。

稍加思考，一张空的课表是最容易安排的，因为我们可以立即按照最理想的方案将待排课程填入。可是如果按照组合规划模型来处理此类事情将非常复杂，这就不能不引人深思了。

因此，放弃了寻求“单个绝对最优”方案的企图。在课表安排中，除了满足所有硬件约束条件外，还能够满足人工排课中的“合理、实用”的要求，就认为被安排的课表是可行的并且相当较优的。

根据以上的分析和人工排课的过程，可以确定用计算机来编排课表的目标有：

1 课表中没有任何硬性冲突。它表示课表必须符合基本硬约束条件，没有办法执行的冲突。如：同一教师不能同时给两门或两门以上的不同课程授课；同一教室不能同时安排两门或两门以上的不同课程等。它也表示课表还遵守硬约束条件，这些约束条件的违反虽然不会导致任何冲突，但是也是现实中衡量课表是否可行的标准之一。

2 课表具有较高的质量。一套高质量的课表，在教师、学生、教室、时间、课程等很多方面都应该做到科学安排，并且尽量有人性化的考虑。

3 课表安排过程中应该不能出现太多的未安排课程。虽然未被安排的课程

的出现,可能是没有足够的教学资源,或者安排中没有充分考虑后继课程的要求而导致的,但是因为在一套完整的课表中,调整某门课程的安排,往往会牵连到更多课程的安排。

通过以上几个排课目标的分析,也可以看出排课问题的难点在于:保证课表在时间分配上符合一切共性和个性要求,质量过关,没有违反规则的地方。在此基础上,所有的课程都能够安排合适的时间和课室,使排课方案在各个目标上尽量达到全局最优。

4.3 排课系统分析

4.3.1 排课系统用例图

下图是整个系统的用例图:

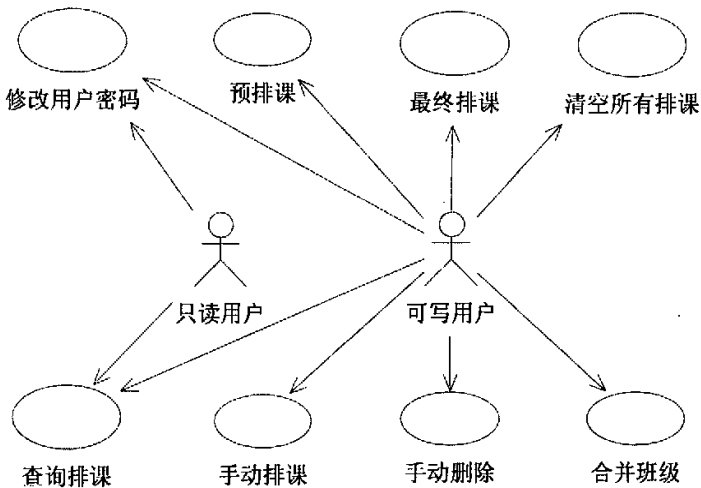


图 4-1 排课系统用例图

Fig 4-1 the use case chart of arranging course system

角色解释:

只读用户:此用户没有对排课数据的增、删、改权限,只能查看排课结果和

修改自己的用户密码。







可写用户：此用户拥有排课管理系统的全部操作权限，包括查询和增、删、改。



4.3.2 排课系统用例说明

下表给出了各个用例的说明：

表 4-2 排课系统用例说明

Table 4-2 the explain of use case in arranging course system

用例图	说明
 修改用户密码	用户进入系统后可修改自己的密码
 查询排课	可按学期、校区、课程来查看排课结果，同时查询结果也可作为待排课的课程范围
 预排课	对选定的课程进行排课，但结果在保存到数据库时只保存上课时间而不保存上课地点
 最终排课	对选定的课程进行排课，与预排课不同的是，结果包括上课时间和上课地点都要保存到数据库
 手动排课	对单独的特定课程进行手动调整，提高排课的灵活性
 手动删除	对特定的课程排课结果不满意，可单独删除其排课结果

 清空所有排课	删除选定的所有课程的排课结果。
 合并班级	把两个班级合到一起上课

下面对主要用例进行详细说明。

(1) 预排课

根据深圳大学成人教育学院的要求，课程应该先有个上课时间，然后发布出来，让学生可以根据自己的时间安排选择课程。这样在学生网上选课之后，还可以根据每门课程的选课人数的多少来决定是否开课，是否需要合并班级。

(2) 最终排课

在经过预排课和学生网上选课之后，就可以进行最终的排课了，它在算法实现上与预排课完全相同，不同的只是最后保存排课结果时保存上课地点，而且它不修改预排课已经排出来的上课时间。

(3) 手动排课

对一门课程的排课结果涉及到两种情况，一种是这门课已经有上课时间了，那么它只是去寻找在此定好的时段可用的教室；另一种是课程还没有定上课时间，那么系统将根据课程的面授周数、总学时、总共人数、形式（全日制或夜大）几个条件来为此课程生成上课时间和上课地点。

4.4 系统的体系结构

参考教学管理系统的体系结构，在结合排课系统的运行需求设计出如图 4-2 排课系统的体系结构：

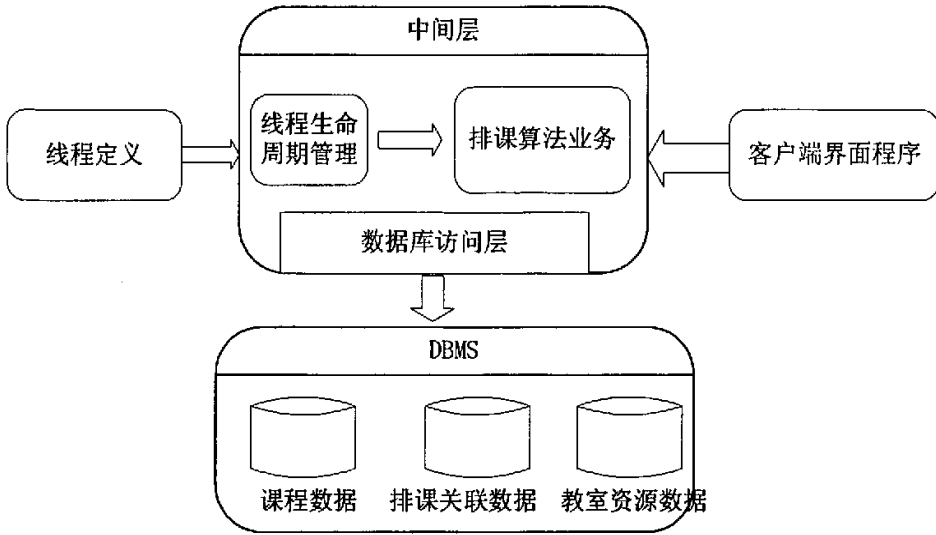


图 4-2 排课系统的体系结构

Fig 4-2 the System Structure of arranging course

客户界面端：是排课系统的直接于用户接口的部分，它负责用户于应用间的对话功能。系统的数据输入和输出。当需求变化需要更改用户接口是，只是改写显示控制和数据访问程序，而不影响其他两层。

中间业务处理层：是排课系统的一个中间件（Middleware），是整个排课的调度中心，自动排课时是通过线程来实现的。它包括对排课过程数据结构链表的初始化、排课算法的运行、排课结果的保存等操作。提供用于激活外部应用程序或者访问排课数据的接口。

数据库服务器：对数据库的排课数据的读写。本设计中关系数据库采用 SQL Sever2000。

本系统结构是基于关系的轻量级的结构，该体系结构严格按照传统意义上的基于 C/S 的三层结构进行设计。

4.5 类和数据库的设计

4.4.1 类设计

经过对系统的分析,抽象出图 4-3 所示的类。

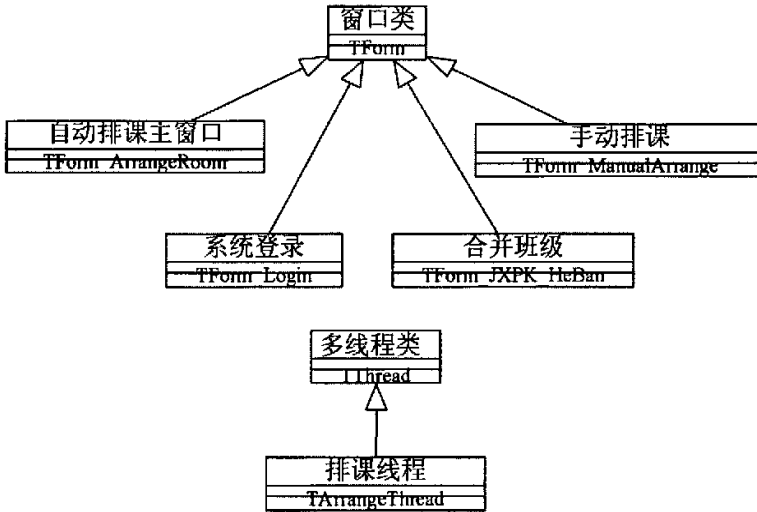


图 4-3 排课系统的类图

Fig 4-3 the class chart of the arranging course system

(1) TForm

Borland C++ Builder 提供的窗口基类, 自己在定制窗口时都可继承于它。

(2) TForm_ArrangeRoom

排课系统主窗口类, 预排课与最终排课所使用的属性与方法与包含在此类中。

(3) Form_ManualArrange 手动排课类

(4) TForm_Login 系统登录类

(5) TForm_JXPK_HeBan 合并班级类

(6) TThread 与 TArrangeThread

TThread是Borland C++ Builder提供的一个多线程基类, 如果程序中要用

到多线程，一般可自定义一个继承于TThread的线程类。因为在自动排课过程中涉及大量数据处理，程序需要运行一段时间，而这段时间界面线程会因为得不到响应而陷入假死状态，给用户带来不便，为了防止此情况发生，特建立TArrangeThread线程类来解决这个问题。

4.4.2 数据库设计

排课系统涉及教学管理系统的很多基本数据表，在这里只列出关键数据表。

以下是要申请教室资源的主要课程班表：

表 4-3 教学计划实施表

Table 4-3 T_jxTeachImplement

字段名	数据类型(长度)	含义	键值	备注
fld	DECIMAL (9)	序号	主键	自增
fClass_No	VARCHAR (13)	开课对象	外键	T_jxClassNum
fCourse_No	VARCHAR (8)	课程班号		见说明
fForm_No	VARCHAR (1)	形式代码	外键	T_jxLearnForm
fLevel_No	VARCHAR (1)	层次代码	外键	T_jxCourseLevel
fProfession_No	VARCHAR (3)	专业代码	外键	T_jxProfessionInfo
fProfessionInfoType	VARCHAR (1)	专业类型	外键	T_jxProfessionInfoType
fSchool_No	VARCHAR (2)	办学地	外键	T_jxSchoolArea
fCForm_No	VARCHAR (1)	课程类别	外键	T_jxCourseForm
fCKind_No	VARCHAR (1)	课程性质	外键	T_jxCourseKind
fEKind_No	VARCHAR (1)	考试形式	外键	T_jxExamKind
fESort_No	VARCHAR (1)	考试类别	外键	T_jxExamSort
fEMode_No	VARCHAR (1)	考核方式	外键	T_jxExamMode
fCCharacter_No	VARCHAR (1)	课程特征	外键	T_jxCharacter
fCSort_No	VARCHAR (1)	课程科类	外键	T_jxCourseSort

fAcademy_No	VARCHAR (2)	开课院系	外键	T_jxAcademy
fTeacher_No1	VARCHAR (4)	主教师 1	外键	T_jxTeacherInfo
fTeacher_No2	VARCHAR (4)	主教师 2	外键	T_jxTeacherInfo
fTTeacher_No1	VARCHAR (4)	实教师 1	外键	T_jxTeacherInfo
fTTeacher_No2	VARCHAR (4)	实教师 2	外键	T_jxTeacherInfo
fTermNo	VARCHAR (6)	开课学期		200201
fCTime	VARCHAR (100)	上课时间		见说明
fCPlace	VARCHAR (200)	上课地点		见说明
fRemark	VARCHAR (200)	备注		
fTeacheTime	INT (4)	讲授学时		
fTestTime	INT (4)	实验学时		
fOperTime	INT (4)	上机学时		
fOtherTime	INT (4)	自学学时		
fTotalTime	INT (4)	总学时		
fCredit	FLOAT (8)	总学分		
fPriorCourses	VARCHAR (50)	先修课程		见说明
fTwinCourse	VARCHAR (50)	同选课		
fCsymbol	VARCHAR (1)	开班标识		0—不开, 2—开班
fCombine	VARCHAR (7)	合班代码		合到那个班
fExamTime	VARCHAR (30)	考试时间		
fMaxStudent	INT (4)	上限人数		默认 0
fMinStudent	INT (4)	下限人数		默认 0
fTotalStudent	INT (4)	总共人数		默认 0
fLoginStudent	INT (4)	注册人数		默认 0
fGrade	VARCHAR (4)	年级		
fYear	VARCHAR (10)	年限		
fTeacheDay	INT (4)	面授天数		
fCTime2	VARCHAR (100)	上课时间		见说明
fCPlace2	VARCHAR (200)	上课地点		见说明

fifWebInput	VARCHAR (1)	能否网站 录入成绩	默认 0 (0 为可以, 1 为 不行)
-------------	-------------	--------------	-------------------------

说明:

上课时间: 0109020*1220631#(01:第 1 周, 09:第 9 周, 0:星期天, 2:下午, 0:全
周)*(12:第 12 周, 20:第 20 周, 6:星期六, 3:晚上, 1:单周), 9999999#:时间未定。
以*分隔, #结束

上课地点: (0109020)7-201*(1220631)五山机房#

以下是有关教室资源的表:

表 4-4 教室信息表

Table 4-4 T_jxClassroomList

字段名	数据类型(长度)	可空	含义	键值	备注
fId	DECIMAL (9)		序号	主键	自增
fSchool_No	VARCHAR (2)		校区代码	外键	T_jxSchoolArea
fBuildNo	VARCHAR (3)		楼号	外键	Building
fRoom_No	VARCHAR (5)		教室号		
fTotalSeat	VARCHAR (14)		座位数		
fPriority	VARCHAR (4)	√	优先级		
fNote	VARCHAR (10)	√	备注		
fRoomName	VARCHAR (20)	√	教室名称		
fTermNo	VARCHAR (6)		学期		200401
fFloorNo	DECIMAL (9)	√	楼层号	外键	Floor
fInternativeseats	VARCHAR (14)	√	隔位数		
fFunctionId	DECIMAL (9)		课室功能代号	外键	RoomFunction

表 4-5 课室使用信息表

Table 4-5 T_ksClassroomUseInfo

字段名	数据类型(长度)	可空	含义	键值	备注
-----	----------	----	----	----	----

fClassroomId	DECIMAL (9)		课室 id	外键	t_jxClassroomList
fTermNo	VARCHAR (6)		学期号		
fWeek	Int(4)	√	周数		
fDayInfo	Int(4)	√	星期几		
fDateInfo	Date (8)	√	日期		
fPeriod	Int(4)		节次		
fArrangeItemId	Char (2)	√	安排给哪个项目 ArrangeItem	外键	

4.6 系统主要功能

通过系统的分析可以看出来，该系统按照功能划分可以分为这么五个功能：预排课、最终排课、手动排课、合并班级、删除排课结果，如下图所示，接下来就分别对这五个功能设计进行阐述。

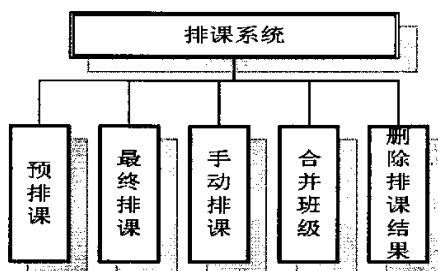


图 4-4 排课系统功能结构图

Fig4-4 the structure of function in arranging course system

4.6.1 预排课

图 4-5 是预排课的时序图，图 4-6 是预排课的活动图：

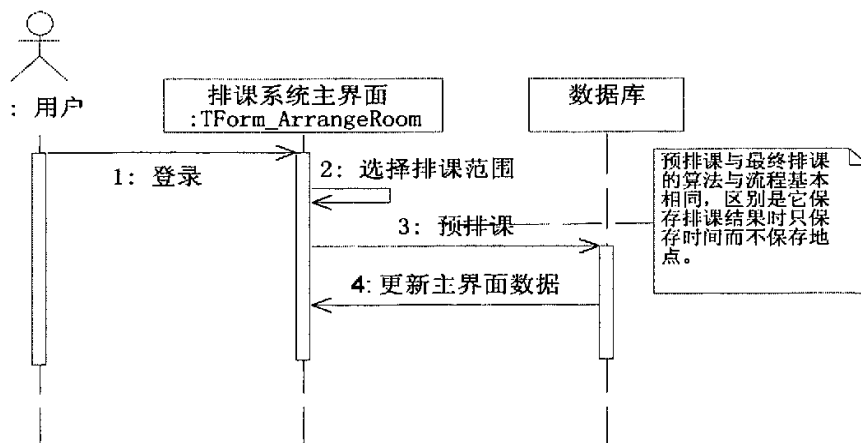


图 4-5 预排课顺序图

Fig 4-5 the sequence diagram of preview arranging course

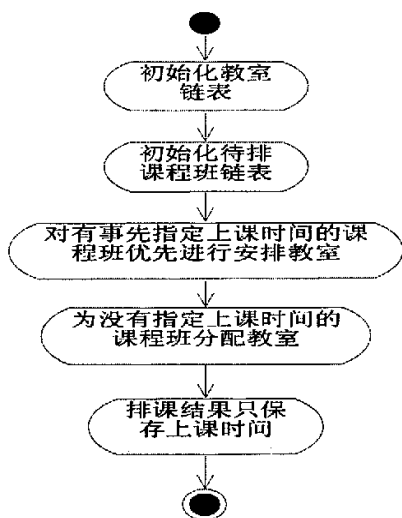


图 4-6 预排课活动图

Fig 4-6 the activity diagram of preview arranging course

4.6.2 最终排课

图 4-7 是最终排课的顺序图，图 4-8 是最终排课的活动图

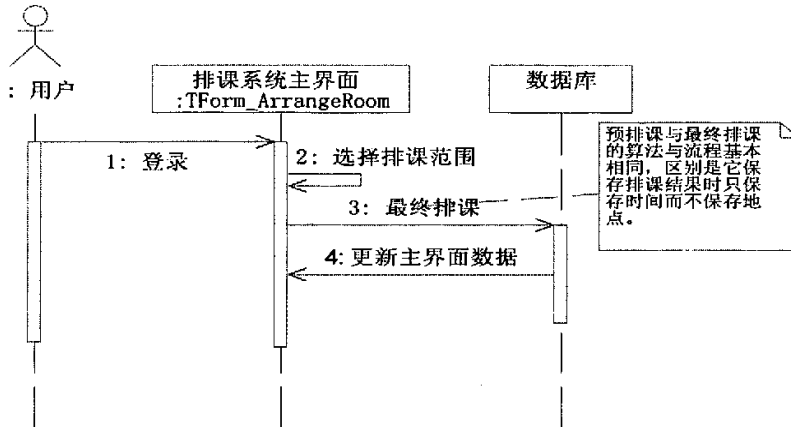


图 4-7 最终排课顺序图

Fig 4-7 the sequence diagram of final arranging course

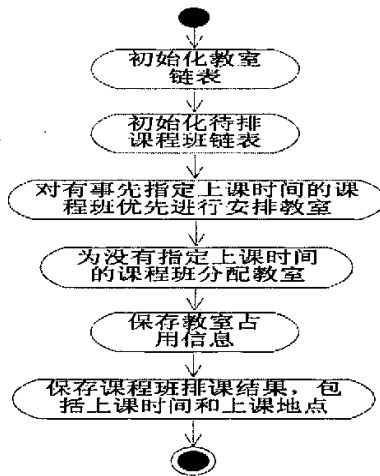


图 4-8 最终排课活动图

Fig 4-8 the activity diagram of final arranging course

4.6.3 手动排课

图 4-9 是手动排课的顺序图，图 4-10 是手动排课的活动图：

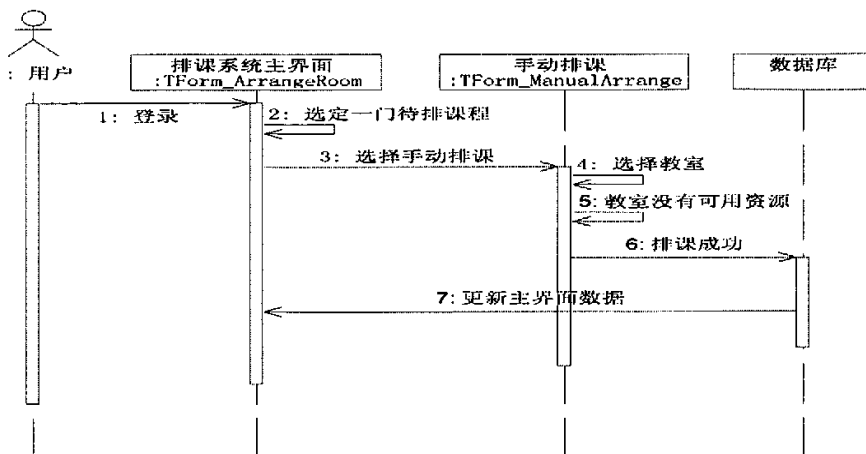


图 4-9 手动排课顺序图

Fig 4-9 the sequence diagram of ManualArrange course

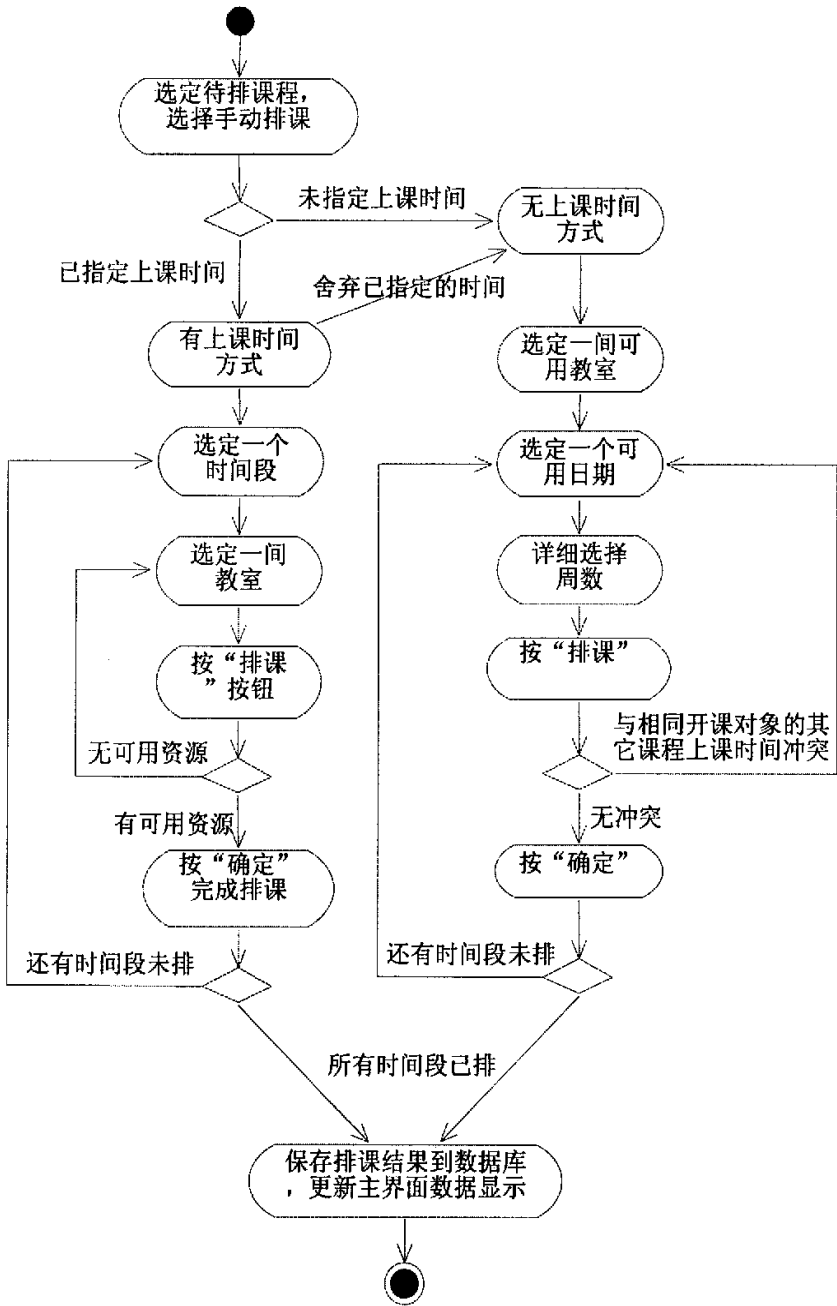


图 4-10 手动排课活动图

Fig 4-10 the activity diagram of Manual Arranging course

4.6.4 合并班级

图 4-11 是合并班级的顺序图，图 4-12 是合并班级的活动图

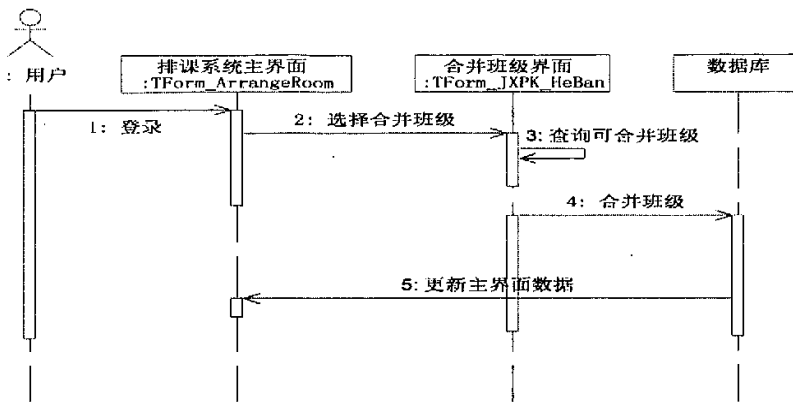


图 4-11 合并班级顺序图

Fig 4-11 the sequence diagram of unite class

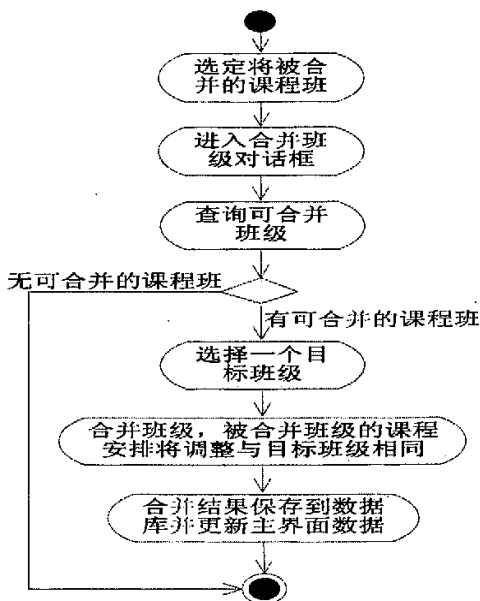


图 4-12 合并班级活动图

Fig 4-12 the activity diagram of unite class

4.6.5 删除排课结果

图 4-13 删除课程班排课结果顺序图, 图 4-14 是删除课程班排课结果的活动图,

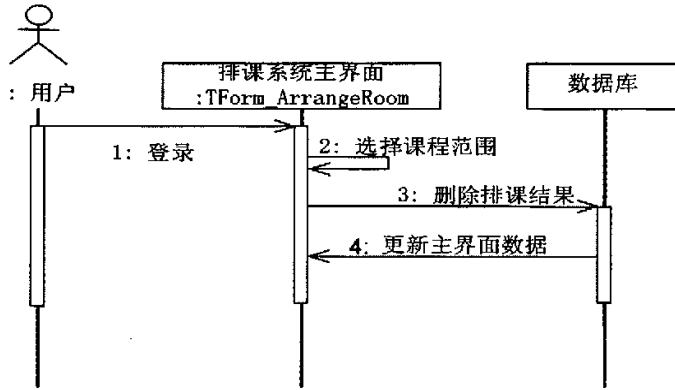


图 4-13 删除课程班排课结果顺序图

Fig 4-13 the sequence diagram of Deleting the result of arranging course

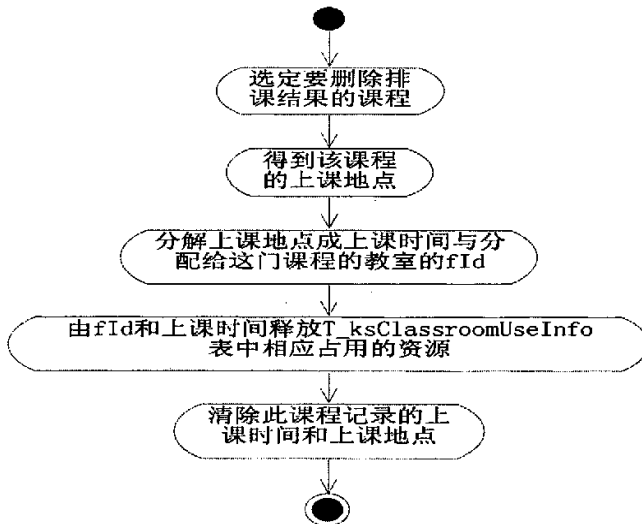


图 4-14 手动删除活动图

Fig 4-14 the activity diagram of Manual Deleting result of arranging course

4.7 排课系统中的贪婪准则

在贪婪算法 (greedy method) 中采用逐步构造最优解的方法。在每个阶段, 都作出一个看上去最优的决策 (在一定的标准下)。决策一旦作出, 就不可再更改。作出贪婪决策的依据称为贪婪准则 (greedy criterion)。

在排课算法运行时, 主要任务是为在课程班信息链表中的课程班从教室信息链表中找到合适的教室资源。在实际运行中, 需要分两步, 一是为有上课时间要求的课程安排教室, 二是为无上课时间要求的课程安排教室。这两者的贪婪准则有稍微的不同。为有上课时间的课程安排教室时, 其贪婪准则中是一定要满足时间要求的, 在为无上课时间的课程安排教室时, 其贪婪准则中是根据课程的学时要求、隔天上课要求和课室情况而确定上课时间的。

4.7.1 为有上课时间要求的课程安排教室的贪婪准则

为有上课时间的课程安排教室时, 主要是满足课程的要求和尽量充分利用教学资源, 其贪婪准则如下:

- 1 教室功能符合课程要求;
- 2 教室的可用周数跨度满足课程的面授周跨度;
- 3 教室的可用时间满足课程的单双周要求;
- 4 教室的可用节次满足课程的上课节次要求;
- 5 课室的预分配项目满足课程要求
- 6 课室的容纳人数与课程的上课人数分别在两个链表中降序排列的位置的距离尽量接近, 即保证上课人数多的课程在大课室上课。

4.7.2 为无上课时间要求的课程安排教室的贪婪准则

为无课时间的课程安排教室时, 主要是满足课程的要求、考虑班级其他课程上课时间、尽量充分利用教学资源, 其贪婪准则如下:

- 1 教室功能符合课程的需求;
- 2 教室的可用周数满足课程的面授周数要求;

- 3 课室的预分配项目满足课程要求;
- 4 生成的时间与课程同一开课对象的上课时间没有冲突;
- 5 生成的时间与课程同一开课对象的其他上课时间符合间隔要求;
- 6 课室的容纳人数与课程的上课人数分别在两个链表中降序排列的位置的距离尽量接近,即保证上课人数多的课程在大课室上课。

4.8 数据结构

在算法的设计过程中,主要使用了教室信息数据结构和课程班信息数据结构。对于教师授课时间冲突的处理,已经通过客户教学管理的实际运作而消除了,教师或者在排课前提出课程的授课时间要求,这归入课程班信息数据结构中,或者在排课后自己去选择所上的课程,这时课程已经有上课时间,教师需要自己避免授课时间冲突。这里教室信息数据结构和课程班信息数据结构都采用结构体形式,在建立链表时,本算法应用了 C++ STL (标准模板库) 中的 `multimap` 关联容器。

4.8.1 教室信息数据结构及函数、链表

教室信息数据结构主要用来存储从数据库中读取的教室信息,如主键、座位数、课室功能、教室的用途、可用周次、可用节次等。这里定义了结构体 `RoomKey` (教室的键,用来索引 `multimap`,存储教室的基本信息,如主键 `fid`,课室功能号,座位数等)和 `Roomvalue` (教室的值,用作被存储和检索的数据,如可用周,节次等),教室的排序函数对象 `CompareRoom` (排序函数),用于找到第一个大于或等于 `TotalStudent` 教室的对象 `CFindFirstOf` (查找函数),最后还定义了教室信息链表类型 `RoomMap` (教室信息链表)。

```
//教室信息数据结构
```

```
typedef struct
```

```
{
```

```
    int fTotalSeat; //座位数
```

```
    int fFunctionId; // 课室功能号
```

```

int fld;          //T_jxClassroomList 的主键 fld,
                 //设置此项是为了数据库访问的方便
}RoomKey; (教室的键)

typedef struct
{
    AnsiString fArrangeItemId;//记录教室的用途是全日制还是夜大,2为全日制,3为夜大
    AnsiString fDays;          //可用周 (01200: 1至20周全周)
    AnsiString fPeriod;       //可用节次 (01: 星期日早上)
}RoomValue; (教室的值)

//教室的排序函数对象(按座位数从大到小排)
class CompareRoom (教室的排序函数对象)
{
public:
    bool operator()(const RoomKey& room1, const RoomKey& room2)
    {
        return (room1.fTotalSeat>room2.fTotalSeat
                || (room1.fTotalSeat==room2.fTotalSeat
                    && room1.fld<room2.fld));
    }
};

//用于找到第一个大于或等于 TotalStudent 教室
class CFindFirstOf (查找函数)
{
private:
    int fTotalStudent;

public:
    CFindFirstOf(int TotalStudent) : fTotalStudent(TotalStudent){}

    bool operator()(const std::pair<RoomKey,Room Value>& p)
    {
        std::pair<RoomKey,Room Value> g;

```

```

        return p.first.fTotalSeat>=fTotalStudent;
    }
};

//教室信息链表类型
typedef std::multimap<RoomKey,RoomValue,CompareRoom> RoomMap; (教室信息链表)

```

4.8.2 课程班信息数据结构及函数、链表

课程信息数据结构主要用来存储从数据库中读取的课程信息，如主键、总共上课人数、需要的课室功能、开课对象、面授周数、每周上课次数、课程班是全日制还是夜大、上课时间、上课地点。这里定义了结构体 CourseKey（教室的键，用来索引 multimap，存储课程班的基本信息，如主键 fid，要求的课室功能号，总共人数等）和 CourseValue（课程的值，用作被存储和检索的数据，如课程班号，开课对象，面授周数、每周上课次数、上课时间地点等），教室的排序函数对象 CompareCourse（排序函数），最后还定义了课程信息链表类型 CourseMap（课程信息链表）。

```

//课程班信息数据结构
typedef struct
{
    int fTotalStudent; //总共人数
    int fFunctionId; // 课室功能号
    int fid;          //T_jxTeachImplement 表的主键 fid
}CourseKey; (课程班的键)

typedef struct
{
    AnsiString fCourse_No; //课程班号
    AnsiString fClassNo; // 开课对象
    AnsiString fTeachWeeks; //面授周数
    AnsiString CountPerWeek; // 每周上课次数
    AnsiString fArrangeItemId; //记录课程班是全日制还是夜大,1 为全日制, 2 为夜大

```

```

        AnsiString fCTime;    //上课时间

        AnsiString fCPlace;  //上课地点,未排课之前为空字符串
    }CourseValue; (课程班的值)

//课程班的排序函数对象 (按总共人数从大到小排)

class CompareCourse (排序函数)
{
public:

    bool operator() (const CourseKey& Course1, const CourseKey& Course2)
    {

        return (Course1.fTotalStudent>Course2.fTotalStudent

            || (Course1.fTotalStudent==Course2.fTotalStudent

                && Course1.fId<Course2.fId));

    }

};

//课程班信息链表类型

typedef std::multimap<CourseKey,CourseValue,CompareCourse> CourseMap; (课程班信息链表)

```

4.9 排课主要算法

排课时, 首先初始化教室信息链表和课程班信息链表, 然后由课程班信息链表开始循环求解, 针对课程班信息链表的每一个结点, 即一个课程班, 根据贪婪准则在教室信息链表中进行循环查找, 找到一个看上去最优的决策的一个结点, 将其上课时间和上课地点分配给该课程班结点, 同时分裂该教室结点, 删除原教室结点。其详细算法如下:

4.9.1 排课主算法 (0层图)

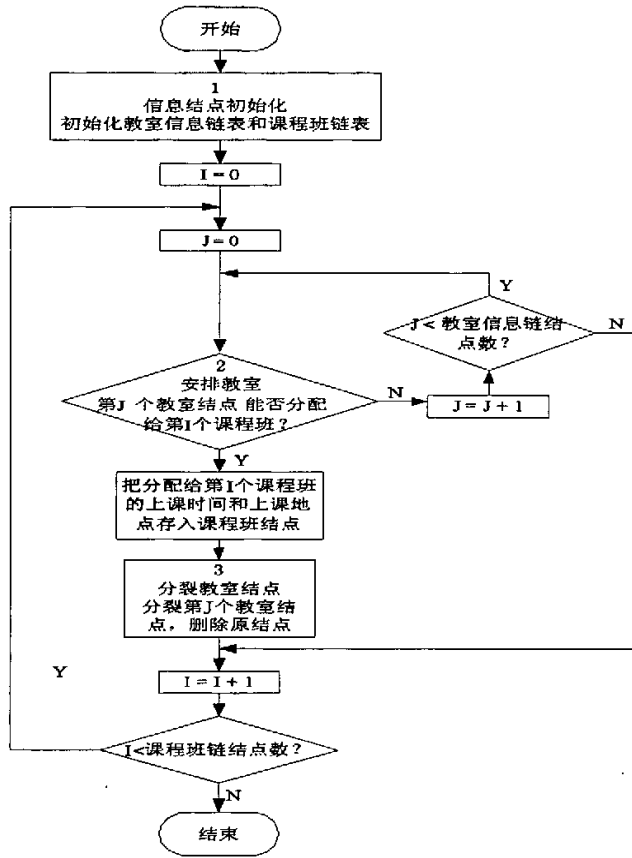


图 4-15 排课主算法

Fig 4-15 the main arithmetic of auto arranging course

在排课主算法中，初始化教室和课程班两条信息链表后，采用双循环程序结构，从 $I=0$ 到 $I=$ 课程班链结点数，对每个课程班信息结点，在教室信息链表中查找合适的教室，从 $J=0$ 开始，直到 $J=$ 教室信息链结点数为止，如果找到合适的教室，则跳出搜索教室结点的循环，把上课时间地点分配给该课程班，并分裂教室结点。由于教室和课程班两天信息链表都进行了排序，所以这种循环也保证了在符合其他贪婪原则的条件下，也符合课室的容纳人数与课程的上课人数分别在两个链表中降序排列的位置的距离尽量接近的贪婪原则，即保证人数多的课程班安排在大课室上课。

4.9.2 信息结点初始化（加工 1 子图）

每间教室第 1 学期第 m 周星期 n 第 p 节是否可用的状态是作为一条记录放在 T_ksClassroomUseInfo（课室使用信息表）中的，而要在内存中处理这些数据，就要把它们收集起来放在教室信息结点中，先从数据库中搜索出数据，然后程序将数据读取到信息结点中，图 5-20 是初始化教室信息结点的流程图：

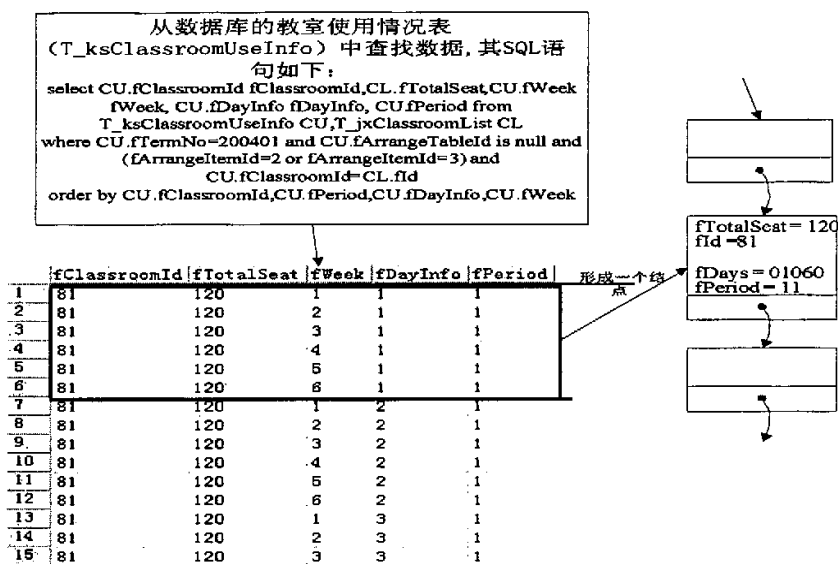


图 4-16 教室信息结点初始化流程

Fig 4-16 the flow diagram of initializing the node of classroom information

课程班信息结点的初始化比较简单，只需要将教学计划实施表中的课程班信息读取到课程班数据结构中即可，这里不详细描述。

4.9.3 安排教室（加工 2 子图）

为课程安排教室算法主要是应用贪婪准则，判断教室结点是否可以分配给指定的课程结点，这里主要写出为无上课时间要求的课程安排教室的算法，为有上课时间要求的算法可按此思路得到。为无上课时间要求的课程安排教室时经贪婪准则依次判断是否满足上课周数要求、生成的时间与课程同一开课对象的上课

时间是否冲突、生成的时间与课程同一开课对象的其他上课时间符合间隔要求等条件。其详细算法如下：

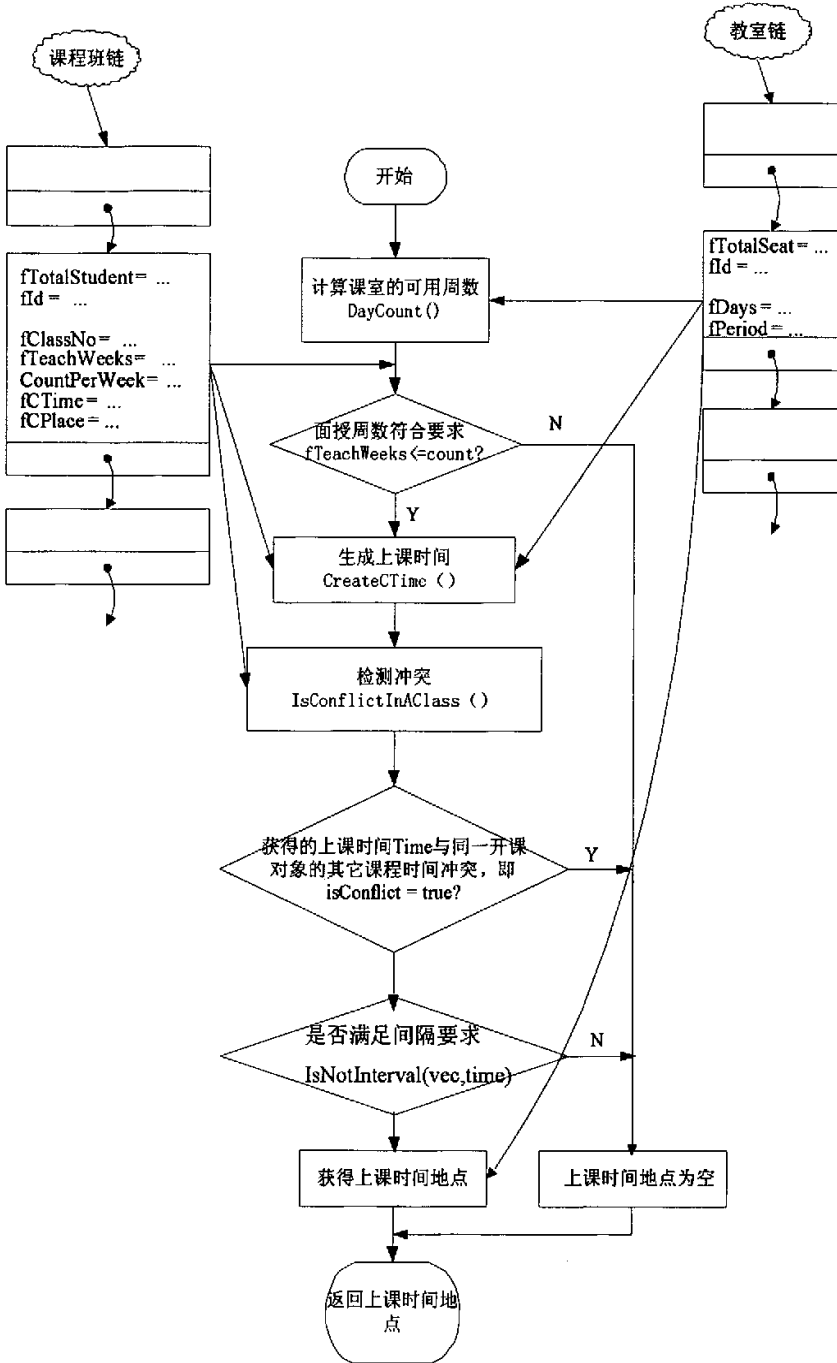


图 4-17 安排教室的算法

Fig 4-17 The arithmetic of arranging classroom

4.9.4 分裂教室结点（加工 3 子图）

当一个教室结点分配给课程班链表中的一个结点后，如果该教室还有可用资源，则需要分裂产生新的结点，删除掉原来的结点。如某教室结点的 $fDays$ 为 01200, $fPeriod$ 为 01, 即该课室从第 1 周到第 20 周的星期天的第一时段都可以使用，而某课程班的需求为授课周数是 16 ($fTeachWeek=16$), 每周上课次数为 1 次。经过分配后，该教室结点的 $fDays$ 为 17200, 即剩下从 17 周到 20 周的星期天的第一时段是可用的。其详细流程如下：

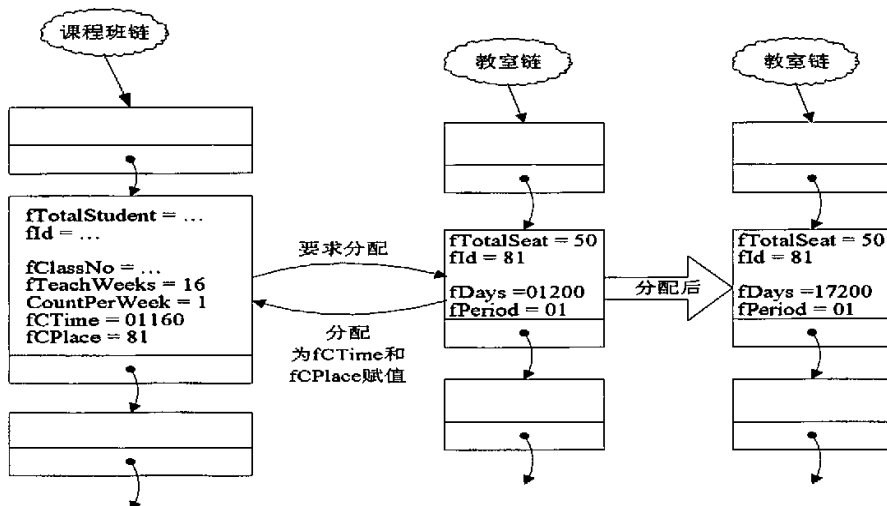


图 4-18 分裂教室结点的算法

Fig 4-18 The arithmetic of disparting classroom node

4.10 本章小结

本章首先阐述了本排课系统采用贪婪算法的原因，接着对排课的情况进行分析，然后对整个排课系统进行分析设计，包括用例图、体系结构、类、数据库和主要功能等。根据项目的实际情况定出排课中用到的贪婪准则，然后详细地描述排课算法中用到教室和课程班的数据结构，给出了主要的算法。

第五章 基于贪婪算法的排课系统实现

设计阶段以比较抽象概括的方式提出了解决问题的办法,进入实现阶段就是要把解法具体化,建立工程蓝图,确立实现模块功能的流程和所需要的算法代码。

5.1 预排课

预排课的实现主要是调用基于贪婪算法的排课程序,先初始化教室信息链表和课程班信息链表,然后按贪婪准则安排,最后将数据保存到数据库中。

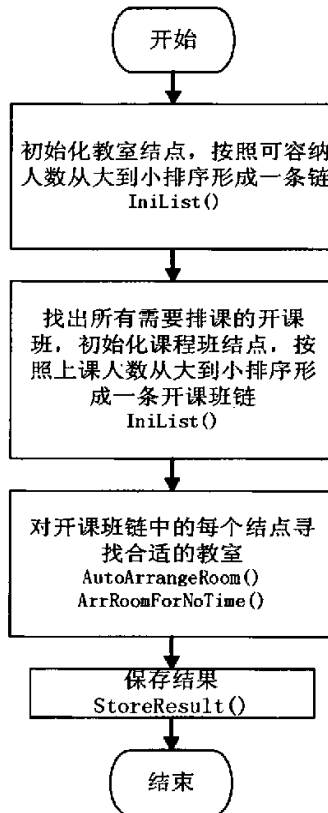


图 5-1 预排课流程图

Fig 5-1 the flow diagram of previous arranging course

5.2 最终排课

最终排课的实现也是通过调用基于贪婪算法的排课程序而实现的，到了该步骤时，学生网上选课已经结束，课程班的实际上课人数已经确定，这些上课人数跟预排课时预计的人数是有差别的，因此需要重新调整课室，保证人数多的课程班在大课室上课，其程序实现时的流程如下：

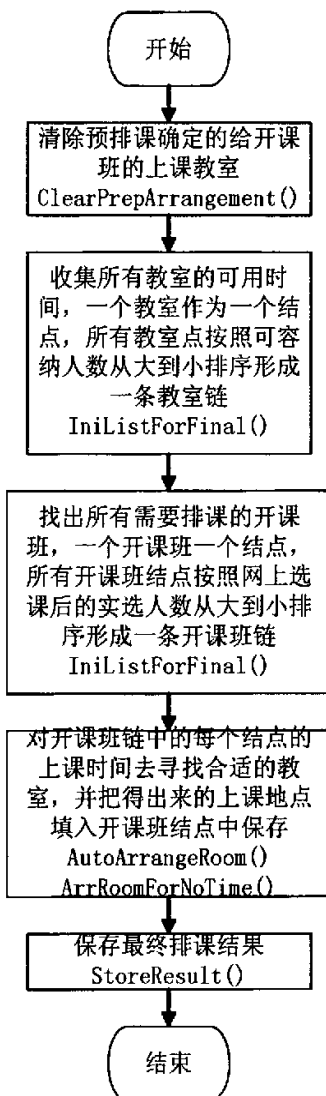


图 5-2 最终排课流程图

Fig 5-2 the flow diagram of final arranging course

5.3 手动排课

手动排课提供排课工作人员灵活选择安排课程，可为指定上课时间的课程安排教室，也可未指定上课时间的课程安排教室，安排教室的时候同样遵从排课要求，确保安排的结果可用，其程序实现时的流程如下：

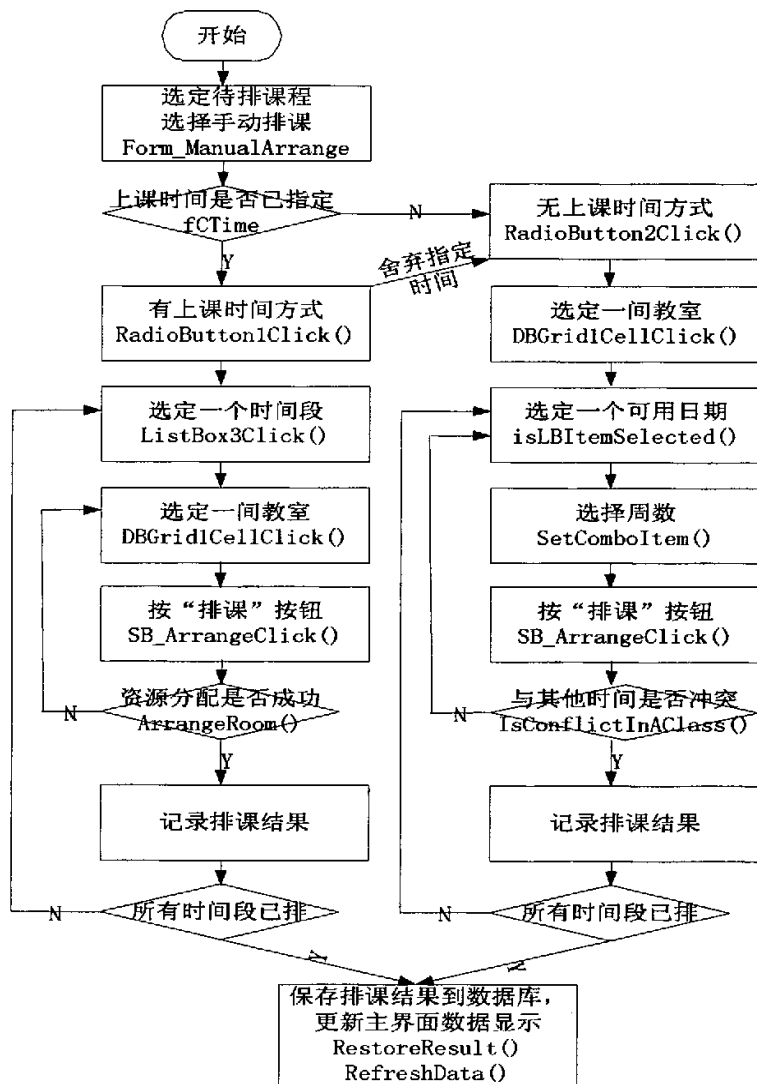


图 5-3 手动排课流程图

Fig 5-3 the flow diagram of Manual Arranging course

5.4 合并班级

系统的排课算法默认的是需要为每个班级的每一门课安排一个上课时间和地点,这样会导致不能充分利用教学资源,也可能会因为没有足够的教学资源而导致部分班级排不上课,根据排课的实际需求,可以根据排课结果的情况调整,将相同专业要求的课程班进行合并,这里系统提供教学管理人员灵活地调整合班,其实现如下:

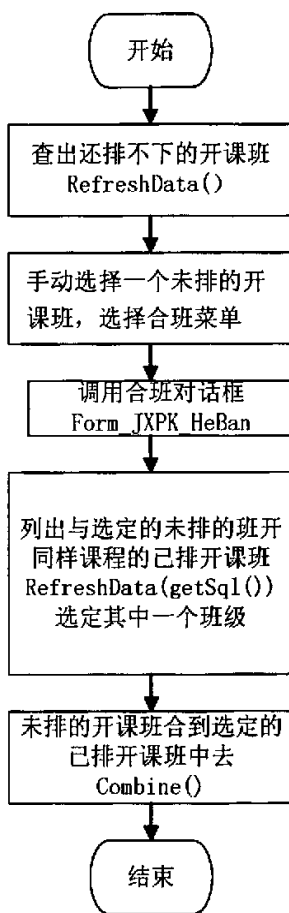


图 5-4 合并班级流程图

Fig 5-4 the flow diagram of unite class

5.5 删除排课结果

在排课过程中,有时需要删除排课结果,这时需要从该课程班的数据记录字段中分解出上课时间和教室,释放掉在教室使用情况表(T_ksClassroomUseInfo)中相应占用的资源,然后清除课程班的上课时间和地点,其实现如下:

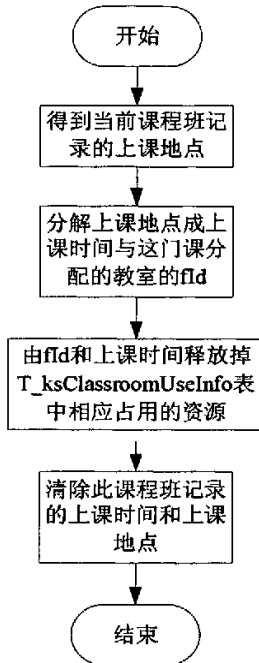


图 5-5 删除排课结果流程图

Fig 5-5 the flow diagram of Deleting the result of arranging course

5.6 保存排课结果

由于算法运行的机制是先从数据库表中搜索数据进入内存,将课程班信息和教室信息建立链表,然后利用链表操作,基于贪婪算法进行排课,当算法运行结束后,需要将排课结果写回数据库表中,这里主要将课程班结点的上课时间和地点写入教室使用情况表(T_ksClassroomUseInfo)和教学计划实施表(T_jxTeachImplement),其实现如下:

(1) 0层图

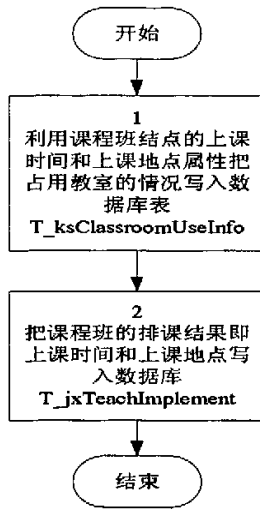


图 5-6 保存排课结果 0 层图

Fig 5-6 the zero layer diagram of storing the result of arranging course

(2) 加工 1 子图 (T_ksClassroomUseInfo 的数据保存)

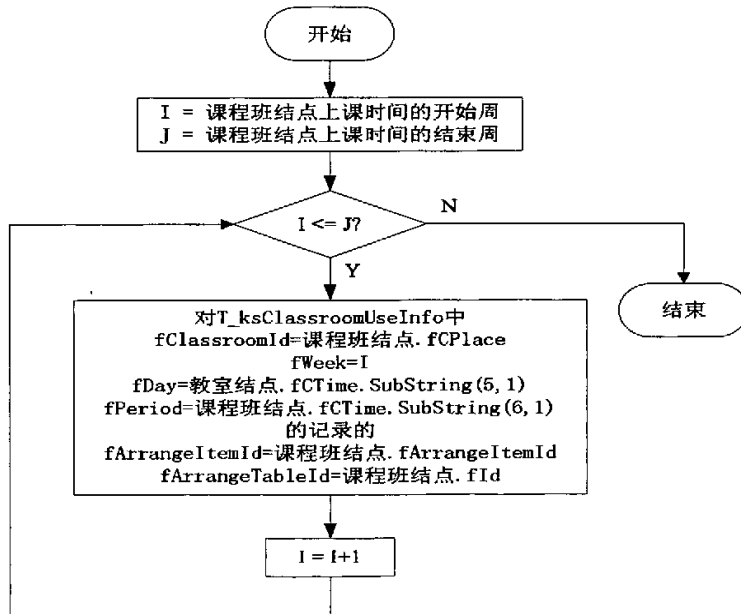


图 5-7 保存排课结果加工 1 子图

Fig 5-7 the first processing diagram of storing the result of arranging course

(3) 加工 2 子图 (T_jxTeachImplement 的数据保存)

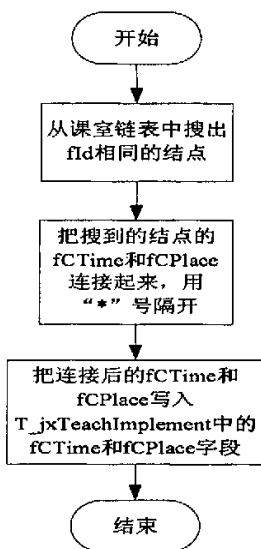


图 5-8 保存排课结果加工 2 子图

Fig 5-8 the second processing diagram of storing the result of arranging course

5.7 部分关键代码

5.7.1 总控程序关键代码

//如果是预排课操作

```

if(m_type==0)
{
    Form_ArrangeRoom->StatusBar1->Panels->Items[0]->Text="正在清空上次预排课结果...";
    Form_ArrangeRoom->ClearArrangement();
    Form_ArrangeRoom->StatusBar1->Panels->Items[0]->Text="正在初始化...";
    Form_ArrangeRoom->IniList();           //初始化教室和课程信息链表
    Form_ArrangeRoom->StatusBar1->Panels->Items[0]->Text="正在为有上课时间的课程安排教室...";
    Form_ArrangeRoom->AutoArrangeRoom(); //自动排课
    Form_ArrangeRoom->StatusBar1->Panels->Items[0]->Text="正在为无上课时间的课程安排教室...";
    Form_ArrangeRoom->ArrRoomForNoTime();
  }

```



```

Form_ArrangeRoom->StatusBar1->Panels->Items[0]->Text="正在保存排课结果...";

Form_ArrangeRoom->StoreResult(); //把排课结果写入数据库      }

//如果是撤消所有排课操作

if(m_type==1){Form_ArrangeRoom->RecallArrangement();}

//如果是清除所有排课操作

if(m_type==2){Form_ArrangeRoom->ClearArrangement();}

//如果是最终排课操作

if(m_type==3)

{

Form_ArrangeRoom->StatusBar1->Panels->Items[0]->Text="正在清空预排课结果...";

Form_ArrangeRoom->ClearPrepArrangement();

Form_ArrangeRoom->StatusBar1->Panels->Items[0]->Text="正在初始化...";

Form_ArrangeRoom->IniListForFinal();           //初始化教室和课程信息链表

Form_ArrangeRoom->StatusBar1->Panels->Items[0]->Text="正在为有上课时间的课程安排教室...";

Form_ArrangeRoom->AutoArrangeRoom();         //自动排课

Form_ArrangeRoom->StatusBar1->Panels->Items[0]->Text="正在为无上课时间的课程安排教室...";

Form_ArrangeRoom->ArrRoomForNoTime();

Form_ArrangeRoom->StatusBar1->Panels->Items[0]->Text="正在保存排课结果...";

Form_ArrangeRoom->StoreResult(); //把排课结果写入数据库

}

```

5.7.2 排课的关键代码

//自动为所有没有上课时间的课程根据面授周数和每一面授天所需上课单元数。

//生成上课时间并安排全适的上课时间

```

while(cmIter!=courselist.end())

{
    rmlter=roomlist.begin();

    while(rmlter!=roomlist.end()) //如果还没有扫描到 roomlist 的尽头

    { //如果教室符合课程的要求

        if(cmIter->second.fArrangeItemId.ToInt()==rmlter->second.fArrangeItemId.ToInt())

```

```
&& cmIter->first.fFunctionId==rmIter->first.fFunctionId

&& DayCount(rmIter->second.fDays)>=cmIter->second.fTeachWeeks.ToInt())

{//生成时间

time=CreateCTime(rmIter->second.fDays,rmIter->second.fPeriod,cmIter->second.fTeachWeeks.ToInt());

//如果生成的时间与同一开课对象的上课时间没有冲突

if(!IsConflictInAClass(ADOQuery3,cmIter->first.fTotalStudent,cmIter->second.fClassNo,CB_Term->Text,time))

{pos=courselist.equal_range(cmIter->first);

vector<AnsiString> vec;

while(pos.first!=pos.second)

{

if(!pos.first->second.fCTime.IsEmpty() && pos.first->second.fCTime!="")

{

vec.push_back(pos.first->second.fCTime);

}

pos.first++;

}

if(!IsNotInterval(vec,time)) //符合间隔的要求

{//写课程班结点

cmIter->second.fCTime=time;

cmIter->second.fCPlace=rmIter->first.fId;

DispartRoomElement(rmIter,cmIter->second.fCTime); //分裂教室结点

break;

}

}

}

rmIter++;

}

cmIter++;

}
```

5.7.3 信息结点初始化的关键代码

```

using namespace std;//使用标准库名字空间

typedef RoomMap::value_type Room_ValueType;

typedef CourseMap::value_type Course_ValueType;

//教室信息结点键-值对

RoomKey roomkey;

RoomValue roomvalue;

//课程信息结点键-值对

CourseKey coursekey;

CourseValue coursevalue;

roomlist.clear();

courselist.clear();

roomkey.fTotalSeat=totalSeat.ToInt();

roomkey.fFunctionId=fFunctionId.ToInt();

roomkey.fId=ADOQuery_Room->FieldByName("序号")->AsString.ToInt();

roomvalue.fArrangeItemId=ArrangeItemIdStart;

roomvalue.fDays=ChangeToWeek(WeekStart)+ChangeToWeek(WeekEnd)+"0";

roomvalue.fPeriod=DayInfoStart+PeriodStart;

//在教室信息链中插入一个结点

roomlist.insert(Room_ValueType(roomkey,roomvalue));

coursekey.fTotalStudent=totalStudent.ToInt()+totalStudentOtherCombine;

coursekey.fFunctionId=fFunctionId.ToInt();

coursekey.fId=ADOQuery_Course->FieldByName("序号")->AsString.ToInt();

coursevalue.fClassNo=ADOQuery_Course->FieldByName("开课对象")->AsString;

coursevalue.fTeachWeeks=ADOQuery_Course->FieldByName("面授周数")->AsString;

coursevalue.CountPerWeek=GetPeriodsPerWeek(ADOQuery_Course->FieldByName("面授学时

")->AsString);

coursevalue.fArrangeItemId=fArrangeItemId.ToInt();

coursevalue.fCTime=*viter;

```

```
coursevalue.fCPlace=PlaceMatchTime(*viter,ADOQuery_Course->FieldByName("上课地点代码")->AsString);
```

```
//在课程班信息链中插入一个结点
```

```
courselist.insert(Course_ValueType(coursekey,coursevalue));
```

5.7.4 分裂教室结点的关键代码

//对一门课分配教室,则原先的教室结点在最一般的情况下会分裂成三个结点。如原先的教室可用时间资源:(可用周数:01200,可用节次:01),课程所需时间资源:(上课时间:0315011第03周至15周01节次单周),则教室分裂成的三个结点的时间资源是:1、(可用周数:01020,可用节次:01);2、(可用周数:16200,可用节次:01);3、(可用周数:04142,可用节次:01)如果上课时间是全周,则不会有第三个结点

```
//如果教室可用日期的开始时间与上课开始时间不相等,生成第一个结点
```

```
if(rmlter->second.fDays.SubString(1,2)!=fCTime.SubString(1,2))
```

```
{
```

```
roomvalue.fDays=rmlter->second.fDays.SubString(1,2)+PreviousWeek(fCTime.SubString(1,2))+rmlter->second.fDays.SubString(5,1);
```

```
//把此结点插入到 roomlist 中
```

```
roomlist.insert(Room_ValueType(roomkey,roomvalue));
```

```
}
```

```
//如果教室可用日期的结束时间与上课结束时间不相等,生成第二个结点
```

```
if(rmlter->second.fDays.SubString(3,2)!=fCTime.SubString(3,2))
```

```
{ roomvalue.fDays=NextWeek(fCTime.SubString(3,2))+rmlter->second.fDays.SubString(3,2)+rmlter->second.fDays.SubString(5,1);
```

```
//把此结点插入到 roomlist 中
```

```
roomlist.insert(Room_ValueType(roomkey,roomvalue));
```

```
}
```

```
//如果教室可用周数是全周并且课程上课时间不是全周,生成第三个结点
```

```
if(rmlter->second.fDays.SubString(5,1)==0 && fCTime.SubString(7,1)!=0)
{
    roomvalue.fDays=fCTime.SubString(1,2)+fCTime.SubString(3,2);
    //如果上课时间是单周
    if(fCTime.SubString(7,1)==1)
    {
        roomvalue.fDays+="2";
    }
    else if(fCTime.SubString(7,1)==2)
    {
        roomvalue.fDays+="1";
    }
    //把此结点插入到 roomlist 中
    roomlist.insert(Room_ValueType(roomkey,roomvalue));
}
//删除原先的结点
roomlist.erase(rmlter);
}
```

5.8 本章小结

本章对各部分功能，包括预排课、最终排课、手动排课、合并班级、删除排课结果和保存排课结果给出详细的实现，还将总控程序、排课、信息结点初始化、分裂教室结点的关键代码都给出来了。

第六章 系统运行情况分析展望

6.1 系统运行情况

6.1.1 自动排课

进入排课系统打开排课主界面，选择要排课的学期和校区（鉴于现在所有的数据，建议选择学期：200502，校区：成教学院），如果想排全部课程，下面的“待排课程”下拉框可不选（默认为所有课程都进行排课），否则可选择想排的具体的哪门待排课程。选择自动排课，则排课会自动分六个步骤完成：

(1) 初始化教室信息；(2) 初始化待排课程班；(3) 对有事先指定上课时间的课程班优先进行安排教室；(4) 为没有指定上课时间的课程班分配教室；(5) 保存教室占用信息；(6) 保存课程班排课结果。

图 6-1 是最终排课运行时的截图：

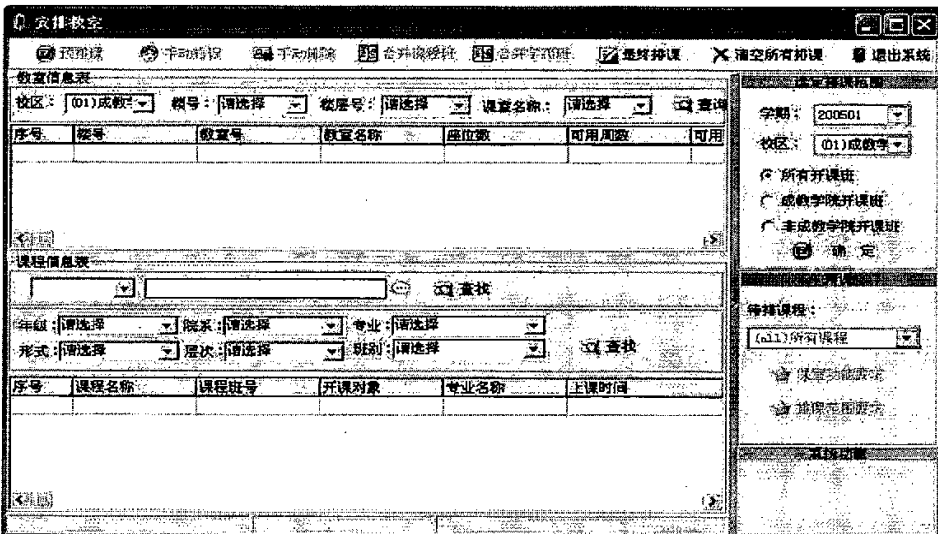


图 6-1 最终排课运行截图

Fig 6-1 the running interface diagram of final arranging course

图 6-2 是最终排课运行结束的截图：

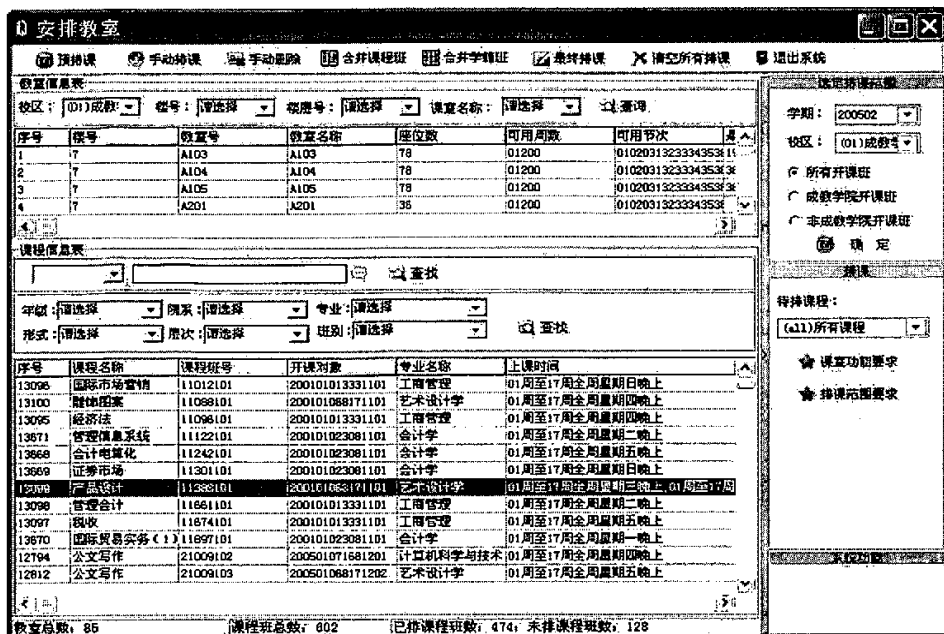


图 6-2 最终排课运行结果

Fig 6-2 the running result of final arranging course

6.1.2 手动排课

对着某个课程班单击手动排课或右键选手动排课，将弹出手动排课对话框。

已指定上课时间方式

图 6-3 是已有上课时间方式的手动排课，操作步骤为：选点一个上课时间段，再选定上课时间符合的课室点击「排课>>」，可以选择不同时间段的课室，其操作类似，最后点击「确定」即可保存排课结果。

6.1.3 合并班级

选定一待合班的课程班，按“合并班级”按钮，打开合并班级对话框，按“查找可合并班级”按钮，得到可合并目标班级的列表，如图 6-5 所示。双击其中一个目标班级，点击“合并班级”按钮即可完成合并班级的操作。

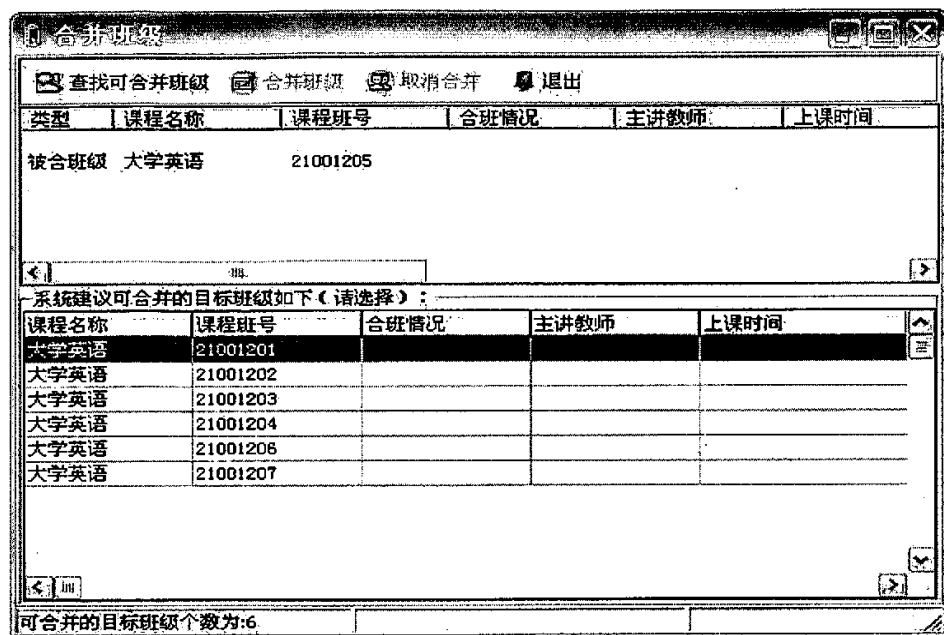


图 6-5 合并班级

Fig 6-5 unite class

6.1.4 打印课表

排课运行后进入打印课表的操作，得到课程表如下：

深圳大学成人教育学院
2002 年年度第一学期教学开课程表

上课地点、课程名称										
课程名称	课 时	课 时	课 时	课 时	课 时	课 时	课 时	课 时	课 时	课 时
课程名称/学时	课程名称/学时	课程名称/学时	课程名称/学时	课程名称/学时	课程名称/学时	课程名称/学时	课程名称/学时	课程名称/学时	课程名称/学时	课程名称/学时
2002 年 11 月 1 日	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)
2002 年 11 月 2 日	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)
2002 年 11 月 3 日	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)
2002 年 11 月 4 日	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)
2002 年 11 月 5 日	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)
2002 年 11 月 6 日	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)
2002 年 11 月 7 日	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)
2002 年 11 月 8 日	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)
2002 年 11 月 9 日	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)
2002 年 11 月 10 日	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)
2002 年 11 月 11 日	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)
2002 年 11 月 12 日	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)
2002 年 11 月 13 日	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)
2002 年 11 月 14 日	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)
2002 年 11 月 15 日	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)
2002 年 11 月 16 日	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)
2002 年 11 月 17 日	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)
2002 年 11 月 18 日	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)
2002 年 11 月 19 日	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)
2002 年 11 月 20 日	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)
2002 年 11 月 21 日	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)
2002 年 11 月 22 日	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)
2002 年 11 月 23 日	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)
2002 年 11 月 24 日	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)
2002 年 11 月 25 日	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)
2002 年 11 月 26 日	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)
2002 年 11 月 27 日	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)
2002 年 11 月 28 日	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)
2002 年 11 月 29 日	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)
2002 年 11 月 30 日	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)	大学英语 (1)

图 6-6 打印课表

Fig 6-6 print timetable

6.2 排课系统运行情况分析

目前这套系统已实际运用在深圳大学成人教育学院的排课作业中，课程班总数为 602，教室总数为 85，经过排课后，得到已排课程班数为 474，未排课程班数为 128（其中大部分是教学管理中指定不需要排课的课程，如毕业设计或者另有安排的）。这一排课过程中，程序运行总共耗时不到两分钟。根据自动排课出来的结果再稍微手动调整（主要是合班操作），就可以得到可用的课程表。用户对排课出来的结果和系统反映速度都比较满意。所以说这个系统基本就是一个达到用户需求的系统。

6.3 展望和进一步的工作

6.3.1 展望

基于贪婪算法的排课系统，能够较好地满足用户的需求，有力地支持整个教学管理系统的运行，有着很好的发展前景。

广东工业大学计算机工程研发中心是华南地区近几年涌现出来的 IT 科技生力军，它依托广东工业大学计算机学院，拥有教授、副教授、博士等几十名高级研究人员，以及大批由研究生为主、包括本科生组成的开发队伍，具有雄厚的技术研发实力。一直以来，广东工业大学计算机工程研发中心致力于教学管理系统、电子商务、电子政务、网络管理信息系统、办公自动化及小区智能化系统研究。正在形成有自己独立开发产品和自主研究关键技术的产学研一体。早在成立之初，广东工业大学计算机工程研发中心就开始了教学管理系统的研发，现在基于贪婪算法的排课系统已经在实际运行中测试验证通过，将进一步完善，应用于以下项目：

1. 深圳大学成人教育学院教学管理系统
2. 广州大学继续教育学院教学管理系统
3. 广东商学院继续教育学院教学管理系统

6.3.2 进一步的工作

本系统具有很多的特殊性：

1. 针对深圳大学成人教育学院的排课需求而设计的，具有其应用的特殊性；
2. 贪婪准则的确定也具有其特殊性，只针对项目中的排课规则进行设计，对其它排课的人文因素考虑较少；
3. 教师上课时间的冲突处理由客户方的实际运作情况消除，而没有让系统进行直接处理；
4. 运行平台的特殊性——Windows 操作系统平台，在其它平台上运行还需要做较多工作。

针对上述的特殊性，进一步的工作还有：

1. 针对平台的特殊性问题：接下来的工作是可以使用跨平台的编程语言，修改程序中的平台特有性系统调用，尽量使用 ANSI 标准中定义的系统接口，在不同的平台上进行编译、运行和测试，提供不同平台的版本；
2. 针对教师上课时间冲突的处理问题：接下来的工作是进行分层次任务的贪婪算法设计，先处理教师与课程班的编排问题，再处理课程班安排教室的问题；
3. 针对贪婪准则的问题：接下来的工作中是根据用户的需求，在贪婪准则中加入实际项目中的考虑因素。
4. 针对应用的特殊性问题：接下来的工作是对排课系统进行详尽的研究分析，提炼出公共部分，归纳参数，形成根据参数设置就可以形成应用的排课系统。

6.4 本章小结

本章介绍了基于贪婪算法的排课系统的运行测试结果和分析，并给出了系统的展望和进一步的工作。

总结

随着计算机的广泛应用和互联网技术的高速发展,在全国高校中许多教学管理系统相继投入使用。排课工作是任何一个学校每个学期的首要例行性业务,是负责课程业务人员的最大工作负担;由于其牵涉到的不同资源如上课的校区、大楼、教室、课程、教师、班级、学生及上课器材等,而如何避免资源的冲突使用则是一项困难的工程,再加上学校本身的其他相关行政规定所构成的排课规定,可以看出要进行能同时满足这些因素和限制条件的排课业务是非常复杂和困难的。正由于其复杂和困难而没有结构的特性,排课问题很难以量化模式描述而难以将其以电脑化来处理,因此会耗费大量的人力与时间来处理该项业务。

本论文的选题是来源于我在研究生期间接触到的一些实际项目和涉猎到的感兴趣的研究方向。在跟踪同期国内外排课算法最新研究动态的过程中,我注意到贪婪算法较好地满足用户排课需求,应用前景非常广阔,并选择了基于贪婪算法的排课系统的研究和实现作为论文的主题。

本文主要讨论了基于贪婪算法排课系统中的研究和实现,对目前项目中排课系统遇到的问题,深入调研和对大量的文献的研究,提出应用贪婪算法解决排课系统问题的方法,并结合实际项目予以初步实现。

论文主要在以下几个方面做了一些研究工作:

1. 在参照了排课算法的大量文献上,根据项目中排课的实际需求,设计基于贪婪算法的排课方案、数据结构、主要算法等。
2. 根据排课的需求,依据排课时的实际考虑因素,确定贪婪算法中的贪婪策略。
3. 采用三层体系结构,将基于贪婪算法的排课核心程序写成线程的形式成为中间层,由排课主界面程序调用后在后台开辟线程进行运行,不影响其他功能的继续操作。
4. 为了提高排课速度,将数据库中的数据读取到内存中进行处理,算法应用了 C++ STL (标准模板库) 中的 `multimap` 关联容器建立两条链表,一条是教室信息链,另一条是课程班信息链,然后进行基于资源匹配的贪婪准则的排课算

法的运行。

5. 采用面向对象方法分析设计整个排课系统, 用 C++Builder6.0 工具开发实现该排课系统, 并在实践中验证。

但由于自身水平、设备条件、时间仓促等因素的制约, 论文所做的研究还有很多不足, 在现有研究工作的基础上, 有待补充、改进或者完善的工作主要有:

1. 进一步完善基于贪婪算法的排课系统, 达到较高的通用性和效率, 推广到更多的实际应用项目中。

2. 对于贪婪准则的确定还是建立在人工排课考虑的因素上, 没有充分考虑提高课表质量的其它人文因素, 需要加强这方面的研究。

3. 目前实现的排课系统, 只是满足用户现在的排课要求, 还没有实现真正的智能排课, 需要进一步研究。

基于贪婪算法的排课系统, 使教学管理的工作人员在面对复杂、庞大的课表编排任务时不会感到压力太大, 它可以按贪婪原则进行安排, 得到满足要求的课表, 在教学管理系统中将会有更好的应用前景。

总之, 贪婪算法有着非常广阔的应用前景, 基于贪婪算法的排课系统的研究和应用, 对高校的教学管理的信息化有着较大的促进作用。相信不久的将来贪婪算法的应用越来越广泛, 它将为社会的发展做出更大的贡献。

参考文献

- [1] 《电脑在排课作业上的应用——问题的性质与几个系统的作法》 刘时渊 资讯与教育杂志 1993.4.
- [2] Selim,SM,"An algorithm for constructing a University faculty timetable". Comput. Edu., 6. pp.323-332. 1982.
- [3] Selim,SM, "An algorithm for producing course and lecture timetable". Comput. Edu., 6. pp.323-332. 1983.
- [4] Loo,E.H. Goh,T,N. and Ong, H.L., " A heuristic approach to scheduling university timetables.", Comput. Edu. 10(3),pp.397-388.1986.
- [5]胡小兵,鲁宏伟.基于模糊专家系统的排课系统关键技术研究[J].长沙电力学院学报(自然科学版),2001;16(4)
- [6]李盘林,李立健.基于启发性知识研究生院课表编排系统[J].计算机学报,1992;(11)
- [7]董艳云,钱晓群,张宇舒.基于课元组相关运算的高校排课算法[J]西南交通大学学报.1998;33(6)
- [8]曾清海.排课表问题的一种矩阵解法[J].电子科技大学学报,2000,29(5)
- [9]陈谊,杨怡,张国龙等.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;20(2)
- [10]Gotlieb. The construction of Class-teacher time tables. Proceeding IFIP congress, Amsterdam,1963,pp73-74
- [11]S.Even,A.Itai, A.Shamir. on the The Complexity of Timetable and multicommodity flow problems SLAM Journal on Computing 5(1976)691-703
- [12]Time B.Cooper, Jeffrey H.Kingston. The Complexity of Timetable Construction Problems. Proc. ICPTAT'95 ,Pp:183-295,1995
- [13]Ferland,J.A. and Roys. Timetabling problem for university as assignment of activities to resource, Computer and Operations Research, 1985, vol.12, No.2,207-218
- [14]吴金荣.求解排课问题的分支定界算法:[硕士学位论文].中国科学院数学与系

统科学研究院, 2002

[15]何永太, 二部图在排课系统设计中的应用.安徽水利水电职业技术学院学报[J]. 2003[3]:45-47

[16]胡顺仁, 邓毅, 王铮.基于高校排课系统中的图论问题研究. 计算机工程与应用.2002[4]:221-223

[17]刘继清, 陈传波. 模拟退火算法在排课中的应用.武汉船舶职业技术学院学报

[18]李增智, 王云岚, 陈靖. 课程表问题的一种混合型模拟退火算法. 西安交通大学学报 37(4)2003

[19]孙雯. 基于时间位图迭加匹配的课表编排算法. 华东船舶工业学院学报(自然科学版)[J].15[6],2001

[20]陈谊, 杨怡, 张国龙, 王尚忠. 基于优先级自动排课算法 PCSA 的设计和实现方案.北京工商大学学报(自然科学版). 20[2],2002

[21]赵辉, 秦维佳. 基于资源匹配的一种大学排课方法. 沈阳工业大学学报 23[3] 2001

[22]陈本庆,马永强,何虎.改进型回溯法在高校排课中的应用. 成都信息工程学院学报. 18[2]2003:150-154

[23]孙建平,梅晓勇,肖政宏,史忠植.关联规则在高校智能排课系统中的应用. 计算机应用.22(5)2002:37-39

[24]M.Dimopoulou, P.Miliotis. An automated university course timetabling system developed in a distributed environment. A case study European Journayl of Operational Research 153(2004)136-147

[25]S.Daskalaki, T.Birbas. Efficient solutions for a university timetabling problem through integer programming. European Journayl of Operational Research (2004)1-15

[26]辛延军. 课表问题及其求解策略的研究; 天津大学硕士学位论文, 1996

[27]D.Abramson. Constructing School Timetable using Simulated Annealing: Sequence and Parallel Algorithm. Management Science,1991

[28]Dvid Avramson, Mohan Krisnamoorthy, Hery Dang. Simulated Annealing Cooling schedules for Timtabling problem,1997

[29]吴承祯, 洪伟. 模拟退火法优化约束条件下造林规划设计的研究. 自然资源学报, 2000

- [30]Chu P C, Beasley, A genetic algorithm for the generalized assignment problem[J]. European Journal of Operationg,1995
- [31]Saffai D, Sigeru O. Incorporating constraint propagation in genetic for university timetable problem [J]. Engineering Application of Artificial Intelligence.1999, 241-253
- [32]Luan F,Yao X. Solving real-word lecture room assignment problems by genetic algorithm, Complexity International[J]. An Electronic Journal of Complex Reseach,1996[3]
- [33]Colorni A, Dorigomarco,Maniezzo V. Metaheuristics for high school timetabling [J]. Computer Optimization and Application. 1998,[9]:275-298
- [34]张春梅, 行飞. 用自适应的遗传算法求解大学课表问题. 内蒙古大学学报(自然科学版) 2002. 33[4]
- [35]Ho sung C. Lee Timetabling Highly Constraint System via Genetic Algorithm [master degree dissertation] Department of Mathematics College of Science University of the Philippines Diliman, Quezon city
- [36]杨宇. 高校排课系统理论与开发—遗传算法在课表问题中的应用. [硕士学位论文] 北京: 北京理工大学 2003
- [37]胡小兵, 鲁宏伟. 基于模糊专家系统的排课系统关键技术研究. 长沙电力学院学报(自然科学版) 16[4],2001
- [38]龙一飞, 郭文宏. 基于知识推理的排课系统. 电脑开发与应用 13(6)2000
- [39]王晓昕. 专家系统在自动排课中的应用研究.北京理工大学[硕士学位论文]2002
- [41]朱文兴, 张千里. 基于 GENET 的时间表问题自动求解算法.小型微型计算机系统 24[7]2003:1335-1337
- [42] Danny Kalev. 使用 <multimap> 库创建重复键关联容器. <http://www.vckbase.com/document/viewdoc/?id=1398>. 2006
- [43]林漳希, 林尧瑞. 清华大学学报(自然版) 24(2)1-8,1984
- [44]王鸣, 吴北雅. 教学调度试用版. 1998
- [45]清华大学计算机与信息管理中心. 清华大学综合教务管理系统简介

攻读学位期间发表的学术论文

- [1] 聂小东, 李振坤, 傅秀芬. 学生网上选课系统中应对高峰访问的策略和实现.
现代计算机. 2005. 06

独创性声明

秉承学校严谨的学风与优良的科学道德，本人声明所呈交的论文是我个人在导师的指导下进行研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，不包含本人或其他用途使用过的成果。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明，并表示了致谢。

本学位论文成果是本人在广东工业大学读书期间在导师的指导下取得的，论文成果归广东工业大学所有。

申请学位论文与资料若有不实之处，本人承担一切相关责任，特此声明。

指导教师签字：



论文作者签名：

夏小东

2006年 5月 25日

致谢

在论文完成之际，也是我将结束硕士研究生阶段之时。研究生三年的学习生活使我一步一步走向成熟，是我一生都铭记于心的人生历程。

首先要向我的导师李振坤教授表示衷心的感谢。在三年的研究生学习生活期间，无论是在学习还是在生活中，李老师都给予了我无微不至的关怀。他朴素的工作作风，严谨的科研精神，渊博的科学知识，敏锐的科学嗅觉，远见卓识的科学创新，正直的人格，都给我留下了深刻的影响，令我毕生铭记。他不但教会了如何求学问，更教会了我如何做人。论文从选题、设计到撰写、定稿过程中，自始至终都得到了李老师的悉心指导和帮助。

衷心感谢广东工业大学计算机工程研发中心的陈平华老师、王帮海老师、刘广聪老师、黄益民老师、王文彦老师、陈靖宇老师、胡志斌老师、王秋杰老师、林穗老师、刘怡俊老师等诸位老师在学习上和生活上给予我的无私指导和帮助。另外衷心感谢计算机学院的余永权教授、傅秀芬教授、凌捷教授、李卫华教授、李代平教授、滕少华教授、徐海水副教授、杨文伟副教授、何振炎书记以及计算机学院其他老师对我的关心和培养。

衷心感谢我的师兄柳正青、刘怀亮、沈芳阳、许兴鹏、朱兵章、师姐邓静、何佳嘉、宋静静、张喆等等，师妹刘畅、张朝晖、黄娟、李和香、师弟李小强、曹鹏、龚志新、沈昱等。衷心感谢项目组中的钟志、罗海生、汪志力、王振生、刘远开、童小杰、余俭烽等所有开发人员。衷心感谢 2003 级研究生同学孙延海、梁海键、刘竹松、谭石强、蓝芳华、陈作霞等，你们给予我很多帮助和快乐。我想我会永远怀念我们一起度过的快乐时光。祝你们幸福、快乐！

感谢生我、养我、爱我的父母，献给所有我爱的人和爱我的人，是你们给了我无限的幸福和动力。今后的无论我走到哪里，你们依然是我爱的港湾。

最后，感谢在百忙之中前来审阅我的论文的老师 and 参加答辩会的老师和同学。

聂小东

2005 年 4 月 20 日

附图表目录

图 2-1 教学管理系统功能流转的业务流程图.....	7
图 2-2 教学管理系统的总体网络拓扑图.....	9
图 2-3 三层体系结构图	10
图 2-4 教学管理系统总体功能结构图.....	11
图 2-5 部分主要静态数据表	13
图 2-6 部分主要动态数据表	14
图 2-7 排课业务流程图	16
图 2-8 学籍管理程序	17
图 2-9 成绩录入程序	18
图 2-10 安排教室程序	18
表 4-1 排课约束条件	30
图 4-1 排课系统用例图	32
表 4-2 排课系统用例说明	33
图 4-2 排课系统的体系结构	35
图 4-3 排课系统的类图	36
表 4-3 教学计划实施表	37
表 4-4 教室信息表	39
表 4-5 课室使用信息表	39
图 4-4 排课系统功能结构图	40
图 4-5 预排课顺序图	41
图 4-6 预排课活动图	41
图 4-7 最终排课顺序图	42
图 4-8 最终排课活动图	42
图 4-9 手动排课顺序图	43
图 4-10 手动排课活动图	44
图 4-11 合并班级顺序图.....	45

图 4-12 合并班级活动图	45
图 4-13 删除课程班排课结果顺序图.....	46
图 4-14 手动删除活动图	46
图 4-15 排课主算法	52
图 4-16 教室信息结点初始化流程.....	53
图 4-17 安排教室的算法	54
图 4-18 分裂教室结点的算法.....	55
图 5-1 预排课流程图	56
图 5-2 最终排课流程图	57
图 5-3 手动排课流程图	58
图 5-4 合并班级流程图	59
图 5-5 删除排课结果流程图	60
图 5-6 保存排课结果 0 层图	61
图 5-7 保存排课结果加工 1 子图.....	61
图 5-8 保存排课结果加工 2 子图.....	62
图 6-1 最终排课运行截图	68
图 6-2 最终排课运行结果	69
图 6-3 已有上课时间方式手动排课.....	70
图 6-4 无指定上课时间方式手动排课.....	70
图 6-5 合并班级	71
图 6-6 打印课表	72