

基于回答集程序的排课系统设计与实现

吕勇全, 陈 寅, 邬家炜, 庄伟杰
(华南师范大学 计算机学院, 广东 广州 510631)

摘 要:排课问题是典型的组合优化和 NP 完全问题。一个好的排课系统既需要能高效地求解排课问题,也需要能提供灵活的定义各种用户约束的方式,但现有的排课系统通常不能同时兼顾这两点。通过实现一个基于回答集程序的排课系统,较好地解决了这一问题。系统的前端接受用户的各种约束条件,并将其转换为回答集程序,后端调用回答集求解器实现排课问题的求解。实验表明,这种实现方式能适应排课条件的变化,具有灵活和可扩展的特点,并在可接受的时间内能排出教师满意度较高的课表。

关键词:回答集;排课;约束

中图分类号:TP18

文献标识码:A

文章编号:1673-629X(2010)06-0228-05

Timetabling System Design and Implementation Using Answer Set Programming

LÜ Yong-quan, CHEN Yin, WU Jia-wei, ZHUANG Wei-jie
(School of Computer, South China Normal University, Guangzhou 510631, China)

Abstract: Timetabling problem is a typical combinatorial optimization problem as well as a NP-complete problem. A good timetabling system needs not only the efficiency in timetabling solving but also the flexibility in defining user constraints. However, the systems at present usually can't deal with these two points at the same time. In this paper, realize a better timetabling system based on answer set programming. The system's Front-End receives various constraints before transforming them into answer set program. The Back-End solves timetabling problem through answer set solver. The experimental results show us that it can arrange a satisfied timetable in an acceptable time because of its flexibility and scalability.

Key words: answer set; timetabling; constraint

0 引 言

自动排课系统的核心问题是时间表问题(Time Table Problems),而时间表问题已被证明是 NP 完全问题^[1,2],并且一个具有实际用途的自动排课系统还需要能灵活地定义和修改各种排课约束。因此,实现一个自动排课系统需要既实现一个高效的求解 NP 完全问题的算法,又提供一个良好的交互界面进行排课约束的定义和修改。自动排课系统的实现相当困难^[3]。

文中实现了一个基于回答集程序设计的自动排课系统。回答集程序设计(Answer Set Programming, ASP)是基于回答集语义^[4]的逻辑程序设计,它是一种

新的知识表示和推理的方式,同时也是一个用于问题求解的工具。ASP 是一种说明性的程序设计语言,只需通过用特定的语言对问题进行正确、完整的描述,不需要程序员编写具体的解决问题的算法,就可以通过调用回答集求解器得到问题的结果。正因为 ASP 是一种说明性的程序设计语言,所以用 ASP 编写出来的程序具有很高的扩展性。当实际问题变化时,只需要修改相关的问题描述,而不必大篇幅地修改程序代码,就能得到需要的结果。

ASP 的应用范围相当广泛,包括限制性程序设计(Constraint Programming)、SAT、逻辑程序设计、智能规划(Intelligent Planning)、数据库等。因为回答集的计算在算法复杂度上被证明是 NP 完全问题^[5],所以在处理其他 NP 完全问题上,如图着色、哈密顿回路等,有着出色的表现。

高效的回答集求解器的存在,使得问题的求解效率可以和直接实现的算法相当^[6,7]。

收稿日期:2009-09-18;修回日期:2009-12-19

基金项目:国家自然科学基金(NSFC60703095);广东省自然科学基金(GDSF07300237)

作者简介:吕勇全(1977-),男,湖南娄底人,硕士研究生,研究方向为知识表示与推理、语义网;邬家炜,教授,研究方向为计算机网络应用。

1 背景知识

1.1 回答集程序

回答集程序由规则、事实、约束和一些说明性的文字组成。规则(Rules)是回答集程序设计中常用的语法类型,在规则中可以对谓词进行推理。规则形如式(1)所示:

head \leftarrow body. (1)

规则(1)的直观语义是如果规则体(body)被满足,则规则头(head)相应地也应该得到满足。事实(fact)可以表示为 course(C)、course_teacher(C, G)等形式, course(C)表示第 C 门课程, course_teacher(C, G)表示第 C 门课程由 G 老师来上。约束(Constraints)是规则头为空的规则,表达的是回答集程序中一类必须满足或者应当尽可能得到满足的关系。有关回答集语义和回答集程序设计的详细内容可参看文献[4,8]。

回答集程序的求解过程一般由三个步骤组成:

(1)编写程序源代码文件;

(2)调用求解器前端(Front End)对源代码文件进行解释,得到一个不包括变量的基程序(Grounding program),常用的前端是 Lparse^[8];

(3)调用回答集求解引擎求解基程序的一个模型,常用的回答集求解引擎有 Smodels^[7]、Clasp^[6]等。

1.2 排课问题

在国外五十年代末就有人开始研究用计算机来实现自动排课,1962 年 Gotlied 曾提出一个课表问题的数学模型^[9],此后人们对课表编排问题的算法^[10,11],解的存在性等问题做了许多探讨,从而使课程表编排问题成为组合规划中的典型问题。

排课问题在教务工作中处于非常重要的地位。我国各类学校根据教学计划的要求,在每学期开学之前,必须根据本校课程设置以及教师、学生、教室、设备等具体情况,编排课程表,以便有秩序地组织全校的教学活动。解决排课问题主要是利用有限的资源将每门课安排在何时、何地,并且教师、班级以及教室的安排不会发生冲突,同时又满足所有约束。

排课系统基本要素是一些变元的集合,分别是课程集合、班级集合、教师集合和时间集合,这些变元的集合就是回答集程序中的事实。排课问题中最关键的是数学模型的建立,文中所用到的模型是以约束为基础的约束模型。排课系统中的约束一般分为三类:基本硬约束、硬约束和软约束。基本硬约束是任何排课系统都必须满足的条件,如:同一时间同一教师不能给两门不同的课程上课。硬约束是排课过程中根据学校的教学条件必须满足的约束条件,如:某些教师或班级在某个时间上可能不能够利用。软约束是应当尽量满

足的一类约束条件,但违反软约束不会导致排课的失败。排课中面临的主要问题是约束条件复杂,条件不断变化,并且要安排出合理优化的排课结果。

2 自动排课系统设计与实现

2.1 系统概述

由于文中是使用回答集程序来求解排课问题的,所以文中的工作就归咎于对排课问题的描述。排课问题可以分为事实和规则部分,事实就是与排课系统相关的基本要素,而规则就是对系统要求的约束条件。

自动排课系统推理引擎使用 lparse + clasp 组合,此前还有一个自行实现的解释子系统,目的是把用户输入的数据转换成 lparse 语法形式的规则和事实。

2.2 排课系统基本要素的 ASP 表示

对于排课系统中的基本要素,在回答集程序中可以用谓词加以表示。变元用一元谓词表示,变元所具有的属性用二元谓词表示。

各基本要素的 ASP 表示如下:

课程集合 Course(C):C 是课程编号,每一个课元对应一个课程编号。课程的属性还包括课程名称、课程优先级等,可以相应地用二元谓词表示为 course_name(C, N)、course_priority(C, P)等。

班级集合 Class(L):L 是班级编号,每一个班级对应一个班级编号。班级属性包括班级名称、班级人数等,可以表示为 class_name(L, N)、class_size(L, S)等。

教师集合 Teacher(G):G 是教师编号,每一名教师对应一个教师编号。教师属性包括教师姓名、学科等,可以表示为 teacher_name(G, N)、course_teacher(C, G)等。

时间集合 Time(T):T 是时间编号,每一节课对应一个时间编号。时间属性包括星期、节次等,可以表示为 time_day(T, D)、time_original(T, O)等。

定义了这 4 个关键变元的谓词表示之后,还需要定义一个最为重要的谓词,也就是排课系统的最终目标,教学任务安排表谓词 timetable(C, L, G, T)。C、L、G、T 分别表示课程编号、班级编号、教师编号及时间编号。定义了这些谓词以后,就可以按照特定的教学任务编写相应的规则代码,进而求得用户需要的课表。

2.3 排课系统约束条件的 ASP 表示

如果将课表按照星期和时间点来划分,一周 $M(1 \leq M \leq 7)$ 天,每天 $N(1 \leq N \leq 10)$ 个时间点。假如一共有 K 个课元,有 C 个班级需要进行排课,则可供调度的资源为 $S = M \times N \times C$ 个,也就是要把这 K 个课元合理安排在这 S 个资源中($K \leq S$)。

中学自动排课属于四元组合问题,即将班级、课

程、教师、时间进行合理组合,在完成教学任务和满足基本硬约束和硬约束条件下,尽可能地满足各种软约束条件。高校自动排课比中学略为复杂,除了考虑上述的四个变元外,还需要考虑教室的容量等问题,在回答集程序中也可以类似地加以描述。

在这里,可以用四维变量谓词 $\text{timetable}(C, L, G, T)$ 来直接定义自动排课的约束条件,但由于四维变量太过庞大,即使很小规模的问题也很难求解。并且实际上每个教学班应上的课程是确定的,课程对应的老师也可以根据学校教学计划确定,即班级、课程、教师的组合基本已经确定。因此,可以采用分层的方法将四元组合问题分解为一次三元组合和一次二元组合。即先进行课程、教师、时间的组合,定义三元谓词 $\text{timetables}(C, G, T)$ 。再进行课程和班级的组合,最终得到所需要的四元变量谓词 $\text{timetable}(C, L, G, T)$ 。经过这样处理后降低了排课的复杂度,提高了排课效率。

2.3.1 逻辑约束

排课系统首要的规则如式(2)、(3)所示:

$1 | \text{timetables}(C, G, T) : \text{time}(T) | 1 : - \text{course_teacher}(C, G).$ (2)

$\text{timetables}(C, L, G, T) : - \text{timetables}(C, G, T),$
 $\text{course}(C), \text{teacher}(G), \text{time}(T), \text{course_class}(C, L).$ (3)

规则(2)的语义是为每一个课元找到一个上课的时间点,其中每一课元所对应的教师可以在学校教学计划中事先指定,规则(2)的目的是为了得到三元谓词 $\text{timetables}(C, G, T)$ 。规则(3)的语义是得到三元谓词 $\text{timetables}(C, G, T)$ 后,可以与二元谓词 $\text{course_class}(C, L)$ 进行组合得到最终所需要的四元谓词 $\text{timetable}(C, L, G, T)$ 。规则(2)和规则(3)实现把每一课元安排在合适的时间点上,并且返回该课元所对应的班级和教师的编号。

2.3.2 基本硬约束

B1:同一时间同一教师不能给两门不同课程上课。基本硬约束 B1 可以用约束(4)表达如下:

$: - \text{timetables}(C1, G, T), \text{timetables}(C2, G, T),$
 $\text{course_teacher}(C1, G), \text{course_teacher}(C2, G), \text{time}(T), C1! = C2.$ (4)

约束(4)表达的语义是如果两个不同的课元 $C1$ 和 $C2$ 所对应的教师和时间点都相同,在回答集程序中将会导致求解失败。这样有效地避免了同一时间同一教师给两门不同的课程上课的情况。

B2:同一时间同一班级不能安排两门不同课程。基本硬约束 B2 可以用约束(5)表达如下:

$: - \text{timetables}(C1, L, G1, T), \text{timetables}(C2, L, G2,$

$T), \text{course_teacher}(C1, G1), \text{course_teacher}(C2, G2),$
 $\text{time}(T), \text{class}(L), C1! = C2.$ (5)

约束(5)表达的意思是如果两个不同的课元 $C1$ 和 $C2$ 所对应的班级和时间点都相同,在回答集程序中将会导致求解失败。这样有效地避免了在同一时间给同一班级安排两门不同的课程的情况。

2.3.3 硬约束

由于硬约束条件随各学校的具体情况不同需要不断改变,因此把硬约束条件设置为可以由用户自行输入的方式,用户只须在图形界面上添加约束条件,解释系统就会自动把用户的输入转换成相应的 lparse 程序中的约束。

H1:某些班级的某些课程可以先行手工安排时间点。H1 可以表达为约束(6)的形式:

$: - \text{not timetables}(C, G, T).$ (6)

对于某一具体的课元可以表示为“ $: - \text{not timetables}(21110130, 10010, 6).$ ”的形式,表示由编号为 10010 的教师所上的编号为 21110130 的课元预先安排在时间点 6 上。

H2:某些教师或班级在某个时间上可能不能够利用。

H2 中如果是某一教师不想在某些时间点上上课,则可以表达为约束(7)的形式:

$: - \text{timetables}(C1..C2, G, T1..T2).$ (7)

如“ $: - \text{timetables}(21110101..21110106, 10001, 1..4).$ ”表示编号为 10001 的教师所上的编号为 21110101 - 21110106 的课元不能排在时间点 1 - 4 上。由于回答集求解器前端 lparse 不支持这种类型的语法格式,因此在解释系统中上述语句被解释为 24 条形如“ $: - \text{timetables}(21110101, 10001, 1).$ ”的约束。

H2 中如果是某一班级某些时间段固定无课,可以表达为约束(8)的形式:

$: - \text{timetable}(C, L, G, T1..T2).$ (8)

具体可表达为“ $: - \text{timetable}(C, 11, G, 1..4).$ ”的形式,表示班级编号为 11 的班级在第 1 - 4 四个时间点上不能排课。

H3:体育课安排在上午 3、4 节或者下午,体育课之后不安排课程,实验课、实习课等课程有自身的安排方式。H3 可以用规则(9)进行处理:

$\text{timetable}(C_0, L, G, T+1) : - \text{timetables}(C, G, T),$
 $\text{course}(C), \text{teacher}(G), \text{class}(L).$ (9)

在规则(9)中,把课程表上不排课的情形当成一个空白课元来处理,空白课元序号记为“ C_0 ”。具体可表达为“ $\text{timetable}(21110000, L, G, 4) : - \text{timetables}(C, 10009, 3), \text{course}(C), \text{teacher}(G), \text{class}(L).$ ”其中

“21110000”为空白课元编号“C₀”,这条规则表示如果由教师编号为“10009”(体育教师)所上的课程安排在上午第三节,则该班的第四节课不排课。

H4:某一时间段不安排课程。H4 可以表达为约束(10)的形式:

:- timetables(C, G, T). (10)

如“:- timetables(C, G, 6).”表示全校在第6个时间点不能排课,一般这些全校统一不排课的时间可以用来作为学校的集体活动时间,方便学校灵活支配,如开会等。

H5:某课程连堂。H5 可以表达为约束(11)的形式:

:- timetable(C1, L, G, T), not timetable(C2, L, G, T+1) (11)

约束(11)表达的语义是,C1, C2 为同一门课的两个课元,如果 C1 排在时间点 T 上,则 C2 必须排在 T+1 时间点上,从而达到连堂的效果。具体可表达为“:- timetable(2111101, 11, G, T), not timetable(2111102, 11, G, T+1), teacher(G), time(T).”的形式。

2.3.4 软约束

软约束指的是排课过程中应当尽量满足的一类约束条件,满足软约束条件可以提高教师的满意度。对于软约束条件的处理增加了排课系统的智能,因此在系统中应当尽最大可能来满足各类软约束条件。在对排课系统进行测试的过程中,发现回答集程序的求解体现出一定的规律性,一般会优先安排课程编号较小的课元,对应的时间点也从时间点编号较小的时间点开始安排。利用这一点可以容易地处理一些软约束条件,提高教师对排课的满意度。

S1:班级课程表在星期上尽量分布均匀。采取的措施是对时间点的编号记星期一的第一节课为 time(1),星期二的第一节课为 time(2),依此类推,这样在排课过程中可以使得班级课程表在星期上尽量分布均匀。除此之外,还可以添加约束条件(12)来限制每门课程在一天中最多出现的次数。

:- timetable(C1, L, G, T1), timetable(C2, L, G, T2), timetable(C3, L, G, T3), class(L), teacher(G), time_day(T1, D), time_day(T2, D), time_day(T3, D). (12)

约束(12)表达的意思是如果由同一老师所上的三个课元 C1、C2、C3 所在的时间点都在同一天,则会导致求解失败,这样就限制了每一门课程在一天中出现的最大次数,从而达到排课均匀。

S2:课表中的每个时间有一定优度。由于排课时

会优先安排编号较小的时间点,这样上午的时间点就具有一定的优度,这样与中学排课过程中课程尽量排上午的方针是一致的。

S3:主课排上午,副课排下午。采取的措施是在对课程编号的过程中先对主课进行编号,后对副课进行编号,这样在排课过程中就会优先对课程编号小的主课进行排课,从而使主课优先安排在上午。

2.4 用户界面的实现

基于回答集求解器的排课系统运行环境为:

1. linux 操作系统;2. JRE 1.6 平台。排课系统由用户添加约束条件界面如图1所示。

用户在界面上添加固定无课时间后被自动解释为约束(7)的形式。

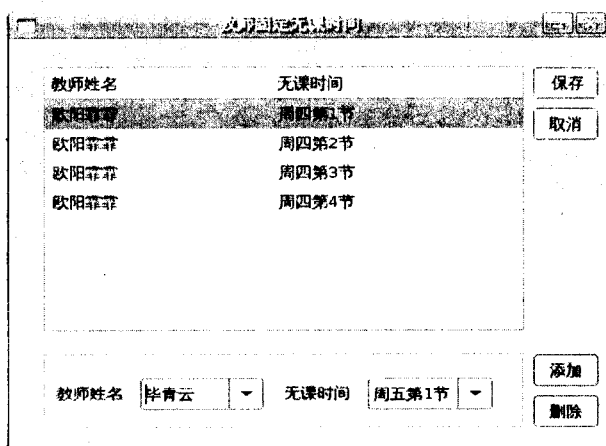


图1 基于回答集求解器的排课系统

2.5 实验结果

在 AMD 3000+、1G 内存、Ubuntu 8.04 的环境下,回答集求解引擎采用了 lparse + clasp 组合,对该系统进行测试。

对于不同规模的数据测试结果如表1。

表1 测试数据

测试 次数	教学任务规模				输入文件大 小单位(Mb)	求解时间 单位(秒)	教师满 意度(%)
	课元数	班级数	教师数	时间元			
1	60	2	10	30	7.02	1.78	100
2	120	4	18	30	41.0	13.72	98.3
3	180	6	19	30	133.3	48.5	96.7
4	240	8	24	30	311.4	406.72	95.8

其中输入文件大小指的是经 lparse 解析后得到的基程序的大小,求解时间指的是前端预处理的时间和回答集求解引擎求解的时间之和,教师满意度指课表中教师满意的课元数与总课元数的百分比。由于排课问题是 NP 完全问题,测试数据显示,随着班级数、课元数及老师的增加,输入文件大小与求解时间呈指数级增长。

3 结束语

由于采用了回答集程序设计,文中的自动排课系统能够灵活地定义排课问题的基本需求,同时也能根据用户的需求定义各种约束。通过一个用户界面,用户定义的需求可以直接被解释为回答集程序的约束进行求解。高效的回答集求解器的使用也使得系统的效率和直接实现的求解算法相当。

实验结果显示该系统的运行效率和求解的问题规模与目前已有的系统相当^[10]。实验中处理了最大规模为 240 课元的直接排课过程。如果考虑到实际系统中对年级和院系的分层,这个规模已足够适用于一般的中学和大学的排课系统。

由于排课系统能灵活地处理各类约束,获得了较高的教师满意度,因此在实践和理论研究中都是非常有意义的。

参考文献:

- [1] Even S, Itai A, Shamir A. On the complexity of timetable and multicommodity flow problems[J]. SIAM J. of Computing, 1976(4):691-703.
- [2] Cooper T B, Kingston J H. The complexity of timetable construction problems[C]//Proc. ICPTAT'95. New Orleans, Louisiana, United States:[s. n.],1995:183-295.

- [3] Gueret C, Jussien N. Building university timetables using constraint logic programming[J]. Information and Computation, 1995(8):12-23.
- [4] Gelfond M. Answer sets[M]//In Handbook of Knowledge Representation. [s. l.]:Elsevier,2007:285-316.
- [5] Dantsin E, Eiter T, Gottlob G, et al. Complexity and expressive power of logic programming[J]. ACM Computing Surveys,2001,33(3):374-425.
- [6] Neumann A, Schaub T, Gebser B, et al. Conflict-driven answer set solving[C]//In Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI2007). Menlo Park: AAAI Press,2007:386-392.
- [7] Simons P, Niemela I, Soeninen T. Extending and implementing the stable model semantics[J]. Artif. Intell.,2002, 138 (1-2):181-234.
- [8] Syrjanen T. Lparse 1.0 user's manual[M]. [s. l.]:GPL, 2000.
- [9] Gotlieb C. The construction of class-teacher timetable[C]//Information processing, Proc. IFIP Congress. Vancouver, British Columbia, Canada:[s. n.],1962:73-77.
- [10] 陈章辉, 黄小晖, 任文艺. 基于双倍体遗传算法求解大学排课问题[J]. 计算机应用,2008,28(12):3074-3076.
- [11] 王 璐, 邱玉辉. 基于协商的智能排课系统的研究[J]. 计算机科学,2006,33(6):214-217.

(上接第 227 页)

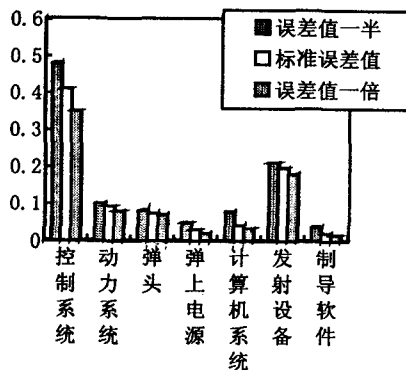


图 3 静态测试时各部件随 Δ 变化的超差百分比

(3) 关于 $v(t)$ 动态测试与积分运算,根据实践经验,主要是精度控制。影响精度的主要因素是精度步长 Δt 的选择。必须理论分析与实践相结合来优选步长大小,在积累误差和积分舍入误差中间求其优化点。

通过利用数据挖掘来对故障进行分析,其优点是可以利用长期积累的测试海量数据分析各部件在可能发生故障中所占的比例,也就是故障的分布规律。这可以有效地指导人们的测试任务的分配,也可为维修零件的采购提供建议。

参考文献:

- [1] 吴文正,周慧中,于广汇,等. 导弹引论[M]. 北京:国防工业出版社,1990.
- [2] 余超志. 导弹概论[M]. 北京:国防工业出版社,1982.
- [3] 崔吉俊. 火箭导弹测试技术[M]. 北京:国防工业出版社,2000.
- [4] Painter M K, Erraguntla M, Jr Hogg G L. Using Simulation, Data Mining, Knowledge Discovery Techniques for Optimized Aircraft Engine Fleet Management[C]//Proceedings of the 2006 Winter Simulation Conference. Monterey, USA: IEEE Press, 2006:1253-1258.
- [5] Han Jiawei, Kamber M. 数据挖掘:概念与技术[M]. 范明,孟小峰,等译. 北京:机械工业出版社,2001.
- [6] Kumaz S, Cetin O, Kaynak O. Fuzzy Logic Based Approach to Design of Flight Control and Navigation Tasks for Autonomous Unmanned Aerial Vehicles[J]. Journal of Intelligent and Robotic Systems, 2009,54(3):230-236.
- [7] 宋远芳. 基于本体的数据挖掘技术在商务智能中的应用[J]. 计算机技术与发展,2009,19(1):184-186.
- [8] 邵晓巍,邵长胜,赵长安. 一类军用无人飞行器协作效能评估方法研究[J]. 导弹与航天运载技术,2005(1):60-62.