

在中学排课问题中实用的模拟退火算法应用^①

唐 环, 高 健

(上海大学 机电工程与自动化学院, 上海 200072)

摘 要: 针对中学排课问题, 提出了一种分阶段的模拟退火算法解决方案. 中学排课问题难点主要在于如何解决课表中存在的大量冲突以及如何优化课表. 初始化随机生成一张带有冲突的课表, 经过算法第一阶段, 人工干预异化解结构, 使课表可行; 算法第二阶段引导性的改变课表结构使课表满足通用的软约束条件; 算法第三阶段采用启发式随机邻域异化操作, 变异课表, 产生更优解. 为了满足实际生产环境中对课表多元化的需求, 在 UI 界面中提供可以手动调节课表机制. 经过实验发现, 改进后的模拟退火算法在解决中学排课时收敛速度更快, 运行效率更高, 并且在迭代次数较少的情况下, 也能产生可行解.

关键词: 模拟退火; 排课问题; 人工智能

引用格式: 唐环, 高健. 在中学排课问题中实用的模拟退火算法应用. 计算机系统应用, 2017, 26(10): 225-230. <http://www.c-s-a.org.cn/1003-3254/6059.html>

Application of Simulated Anneal Algorithm for Curriculum Schedule Problem in Senior High Schools

TANG Huan, GAO Jian

(School of Mechatronic Engineering and Automation, Shanghai University, Shanghai 200072, China)

Abstract: A solution of staged simulated annealing is proposed to settle the schedule problem of senior high schools. The difficulty of the problem mainly lies in how to solve lots of conflicts existing in the schedule and how to optimize it. We initialize a schedule with conflicts randomly, and dissipate the structure of the solution with artificial intervention to make the schedule feasible in the first stage. In the second stage, we try to make the schedule meet general constraints by instructively changing the structure of the schedule. In the third stage, we generate optimized solution through varying the schedule with heuristic dissipate random field employed. In order to meet the demand for diversified schedule, the actual production environment in the user interface provides the manual adjustment to the schedule. It is found that the improved simulated annealing algorithm has a faster convergence speed, and higher operation efficiency in solving curriculum schedule problem of senior high schools and in the case of less number of iterations, it can also generate a feasible solution.

Key words: simulated annealing; curriculum schedule problem; artificial intelligent

学校的排课工作是教学管理任务的重要环节之一, 同时也是耗时耗力较多的一项工作. 传统的排课工作大多由经验丰富的教务人员进行手工编排, 这种人工排课会导致排课效率低、效果差. 为了优化排课质量、充分提高学生学习效率、推进学校信息化办公教

育, 利用计算机实现自动排课已经成为必不可挡的趋势.

排课问题早在 20 世纪 50 年代末, 国外学者便对其展开研究. 1976 年, 排课问题被证明是一个 NP 完全问题^[1]. 这一论证奠定了求解排课问题的理论基础, 学

^① 收稿时间: 2017-02-11; 采用时间: 2017-03-22

者们不再去刻意追求排课问题的最优解,转而采用相适应的启发式算法^[2]求得排课问题的较优解。

现阶段,解决排课问题的方法主要有基于图论算法^[3]、遗传算法^[4,5]、模拟退火算法^[6]、回溯法^[7]、蚁群算法^[8]。然而这些研究主要针对大学课表,中学排课问题有其独有的特点及难点:

(1) 为了保证中学学校的教学计划,对于学生和教室而言,中学课表一般都处于饱和状态,容易引起冲突,而大学生课表则相对自由;

(2) 中学课表问题中每个班级安排的年级课程固定,而在大学课堂上各个教室可以分配不完全相同的教学任务,甚至可以混合多个年级的课程,因此中学教室的灵活性差;

(3) 中学对课表质量要求较高,根据课程性质的不同安排不同的课时,且特殊课程安排在指定的时间较为妥当(如体育课安排在下午最后的课时),从而保证学生在上其他课程时有良好的学习状态;

(4) 中学会区分班主任老师,每个班级至少需要一名班主任老师任课,并且一名班主任老师同时只能在一个班级任职班主任。

1 排课问题的模型及模拟退火算法

1.1 排课问题的模型及模拟退火算法

根据学校教学计划,将一个年级所有班级一周的时刻表在一个平面上展开,用 (c_i, l_j) 表示平面上的单元格,即第 i 个班级第 j 节课时,用 C 表示所有班级的集合, L 表示一周内上课的课时集合,则 $c_i \in C, l_j \in L$ 。设所有老师的集合为 T ,排课的实质就是向所有的 (c_i, l_j) 单元格中填且仅填入一位老师 $t_k(t_k \in T)$,并且保证课程表在完全满足所有的硬约束的条件下,尽量满足软约束条件。其中硬约束和软约束条件归纳如下^[9]。

硬约束:

(1) 全体老师必须完成教学计划中的所有课时,即课程表不能有空缺单元格;

(2) 同一位老师不能在相同的课时在不同的班级同时有教学任务;

(3) 一个班级不能安排不同的老师上同一门课程;

(4) 一个单元格只能安排一位老师;

(5) 班主任数目必须与班级数目相等,且各班至少分配一名不同的班主任老师任课。

软约束:

(1) 体育课尽量安排在下午的最后两节课;

(2) 尽量不安排课程连堂上;

(3) 不同性质的课程,应该酌情分配在不同的时间。例如,主干课程尽量安排在上午1、2节学习效率较高的时段。

(4) 其他差异性需求,例如:需要课程连堂,部分老师需要固定的时间段等。

1.2 模拟退火算法

模拟退火算法由Kirkpatrick于1983年提出^[10],算法主要用于求解大规模组合优化问题,特别是针对NP完全组合优化问题的求解。本文基于模拟退火算法的框架,对于排课这种特殊问题,通过改进其邻域结构的产生策略,从而生成一套高效的求解中学排课问题的方案。

模拟退火算法思路来源于物理学中的模拟退火冶金时采用的Metropolis准则,简而言之,算法的核心思想是随着迭代次数的增加,产生较差的解被接受的概率也越来越小,而产生的较优解一直被接受。

原始模拟退火算法在求解组合优化问题时,采用的是一种高度随机产生邻域结构的策略,算法的迭代次数较长,收敛较慢。

1.2 排课问题的模型及模拟退火算法

(1) 基础方案

将硬约束条件(2),同一位老师不能同时在不同的班级上课,看成软约束条件,其他条件保持不变,由此可生成一张中学课表的解,且该解可能不可行。然后利用模拟退火算法,每次迭代时随机选择同一班级的两门课时进行交换,并按照退火算法中的Metropolis准则决定是否保留变换之后的课表。

(2) 改进方案

模拟退火算法的缺点在于过多的迭代次数,算法的效率较低。基础方案中随机挑选交换课时是导致迭代缓慢的根本原因。人工排课过程可描述为先编排出带有冲突的课表,然后解决冲突,再调整连堂课时,最后全局调优课表。本文基于人工排课的思想提出一种分阶段的课表邻域变化优化策略,有针对性的、目的性的在退火过程中加入“人工干预”。

2 算法实现步骤

2.1 衡量排课质量的优劣

使设课表 S 对应的代价值为 $f(S)$,有:

$$f(S) = \sum_{i=0}^T (\sum_{j=0}^{t_i.M} dc[t_i.c][j] + Q_1 \cdot t_i.N + Q_2 \cdot t_i.F) \quad (1)$$

其中, T 为该年级老师总数; t_i 表示第 i 位老师; 将中学课程分成 7 种不同的性质: 语、数、外、体、文、理、其他, 根据软约束条件的第 (3) 条定义二维数组 $dc[k][j]$ 表示第 k 门性质的课程被安排在第 j 课时产生的代价值; $t_i.c$ 表示该位老师任教的课程性质索引号; 设老师已安排的单元格属性为 $arrangeCells$, 则 $t_i.M = t_i.arrangeCells.size$; 记老师连堂的单元格属性 $connectCells$, 则 $t_i.N = t_i.connectCells.size$; 记老师冲突的单元属性 $conflictCells$, 则 $t_i.F = t_i.conflictCells.size$; Q_1 、 Q_2 分别代表连堂和冲突的代价值常数。

当 $f(S)$ 值越小时, 则课表 S 越合理。

2.2 生成初始解并优化解

2.2.1 划分班级

为了满足硬约束条件中的 (3) 和 (5), 可按如下步骤分配老师任课班级:

Step1. 令 $i=1, c=1$, 不同课程名集合 $COL = \emptyset$;

Step2. 若老师 t_i 为班主任, 则 $t_i.CLL += c, c++$, 其中 $t_i.CLL$ 表示老师 t_i 的任课班级集合. 老师 t_i 的任教课程为 CO_i , 若 $co_i \notin COL$, 则 $COL += co_i$;

Step3. 若 $i \leq T.size$, 则返回 Step2 继续遍历老师集合, 否则转入下一步;

Step4. 令 $N = COL.size$, 令 $k=1$;

Step5. 对于第 k 门课程, 统计任教该课程的所有老师索引号集合 TIL ;

Step6. 遍历 TIL 中老师的 CLL 集合, 生成由 Step2 产生的对于第 k 门课程已分配的班级号集合 CIL ;

Step7. 生成一个从 1 到 c 的列表 $TList$, 并将其乱序;

Step8. $TList - CIL$;

Step9. 根据老师的教授班级数目属性, 对 TIL 中的老师按照 $TList$ 顺序分配每位老师的任课班级加入老师的 CLL 集合中;

Step10. $k++$, 如果 $k \leq N$, 则返回 Step5, 否则班级分配完成, 退出循环。

2.2.2 生成带有冲突的课表

为了满足软约束的第 (1) 条, 可将任教特殊性质课程 (如体育课) 的老师放在老师集合的首位, 其他老师按照一周内总上课课时数目从多到少排序。

由上一节的划分班级, 可以得到每个班级具体由

哪些老师授课. 为了满足硬约束条件 (1), 对于每一位老师, 根据其在每个班级一周上课课时数目的属性, 分配对应数量的该班级课时, 并将所有分配的单元格记录到对应老师的 $arrangeCells$ 属性中; 为了满足硬约束条件 (4), 设计一张班级禁忌表, 用于判断该班级的每个课时是否已被分配. 对于每个可分配的课时, 通过 $dc[k][j]$ 获取固定代价值, 通过老师一周中已上课的课时 $weekY$ 属性决定是否额外添加连堂代价值或冲突代价值, 若发生冲突, 则将该老师的 $conflictCells$ 属性不重复的添加发生冲突的两个单元格. 通过遍历老师的 $arrangeCells$ 属性, 填充二维数组 $sheetInfo[c_i][l_j]$, 其值表示在 c_i 班, l_j 课时任课老师索引号. 连堂属性 $connectCells$ 可以由 $sheetInfo$ 遍历来获取。

2.2.3 解决冲突并优化课表

模拟退火算法的关键步骤在于改变邻域结构, 产生新解, 并以不同的概率接受新解. 针对中学排课问题, 本文将退火过程分为三个不同的阶段, 采用三种不同的邻域异化操作。

(1) 退火第一阶段

这一阶段仅作用于发生冲突的单元格, 改变初始解结构, 生成可行解. 算法步骤如下:

Step1. 若 $\exists t$, 满足 $sheetInfo[c_1][l] = sheetInfo[c_2][l] = t$, 表示教师 t 在第 l 课时在 c_1 、 c_2 班级发生冲突, 进入 Step2 处理, 否则, 课表没有冲突, 退出第一阶段;

Step2. 遍历教师 t 所有的单元格, 获取冲突所在的课时集合 $FLL = \{l_1, l_2 \dots l_n\}$, $FC_i = \{c_1, c_2 \dots\}$ 对应冲突课时 l_i 所在的班级, $FCL = \{FC_1, FC_2 \dots FC_n\}$ 是所有发生冲突的班级的集合, 与 FLL 一一对应;

Step3. 令 $k=1$;

Step4. 令 $i=1$;

Step5. 对于 l_k 和 $FC_k = \{c_1, c_2 \dots c_m\}$, 获取 $ALL = \{l_1, l_2 \dots\}$ 满足 $\forall l_x \in ALL, sheetInfo[c_i][l_x] \neq sheetInfo[*-c_i][l_k] \& sheetInfo[*-c_i][l_x] \neq sheetInfo[c_i][l_k]$, 其中 $*$ 表示所有班级, $[-c_i]$ 表示除 c_i 班以外的班级, ALL 集合表示在 c_i 班中能与 l_k 交换且不使交换的两处位置发生冲突的所有课时集合;

Step6. 剔除 ALL 中包含与特殊性质课程交换的课时, 再依据交换后是否与周围课时连堂而产生额外代价, 采用轮盘赌的方式择优选择一个交换课时, 加入备选交换课时集合 PY 中;

Step7. $++i$, 如果 $i \leq m$ 返回 Step5, 否则执行 Step8;

Step8. 随机选择 PY 中的优选课时对各班进行交换, 选择的个数为 $PY.size - 1$, 更新 $sheetInfo$ 以及发生变化的位置处的老师各个属性值;

Step9. $++k$, 如果 $k \leq n$, 返回 Step4, 否则返回 Step1.

退火第一阶段解决了初始化课表中所有发生冲突的位置, 课表的整体代价值必然是减少的, 算法也一定会接受这些改变.

(2) 退火第二阶段

这一阶段着力解决存在连堂的单元格, 并且保证变化后的解仍为可行解. 算法步骤类似于退火第一阶段, 区别在于观察的是老师的 $connectCells$ 属性. 当分离了连堂的单元格, 课表的整体代价值必然减少, 算法也会接受该解; 当程序无法分离连堂单元格时, 自动转入退火第三阶段.

(3) 退火第三阶段

该阶段处于退火的最后一个阶段, 采用全随机方案优化课表结构, 产生代价值更小的解. 算法步骤如下:

Step1. 随机获取一个班级 c , 并置是否交换参数 s 为假;

Step2. 初始化循环次数常量 N ;

Step3. 随机获取两个不相等的课时 l_1, l_2 教师 $t_1 = sheetInfo[c][l_1]$, 教师 $t_2 = sheetInfo[c][l_2]$;

Step4. 若 $t_1 \neq t_2$ 且 $l_1 \notin t_2.weekY$ & $l_2 \notin t_1.weekY$ 且 $t_1 \neq sheetInfo[c][l_2 \pm 1]$ & $t_2 \neq sheetInfo[c][l_1 \pm 1]$ 则置参数 s 为真, 否则直接转入 Step5;

Step5. 若 s 为假则转入 Step6, 否则转入 Step7;

Step6. 若 $--N > 0$, 则返回 Step3, 否则进入 Step8;

Step7. 交换 l_1, l_2 课时的任课老师, 更新 $sheetInfo$ 以及发生变化的位置处的老师各个属性值;

Step8. 返回退火框架更新最优解结构、温度以及迭代次数信息, 判断是否退出该阶段.

本文中模拟退火算法的内外层循环的终止准则均采用循环总数控制法, 退火总框架伪代码如下:

Step1. 由 2.2.2 节得到初始不可行解 $S = S_0$, 初始化算法参数, 降温次数 T , 迭代步数 N , 初始温度 $t = t_0$, 降温系数 α , 令 $k=0$;

Step2. 若 $k < T$, 令 $n=0$, 否则退出程序;

Step3. 若 $\exists l \in L, (c_1, c_2) \in C$, 使得 $sheetInfo[c_1][l] = sheetInfo[c_2][l]$, 则根据第一阶段邻域变换规则产生新解 $S' = stage1(S)$, 进入 Step5, 否则转入 Step4;

Step4. 若 $\sum_{i=1} t_i.connectCells.size \neq 0 (t_i \in T)$, 则根据第二阶段邻域变换规则产生新解 $S' = stage2(S)$, 进入 Step5, 否则根据第三阶段邻域变换规则产生新解 $S' = stage3(S)$;

Step5. $\Delta f = f(S') - f(S)$, 若 $\Delta f < 0 \parallel e^{-\Delta f/t} > random(0, 1)$, 则 $S = S'$;

Step6. 若 $++n < N$, 则返回 Step3; 否则, $t = t \cdot \alpha$, $++k$, 返回 Step2.

3 数据实验结果

本文采用模拟院校数据: 12 门课程; 50 位老师; 15 个班级; 每天上午有 4 节课, 下午有 3 节课; 每周有六天上课时间, 其中周六只安排上午课程, 下午不安排课程; 一共需要安排 585 节课. 为了对比的有效性, 修改 2.2.1 节的 Step7, 生成列表后不乱序, 即保证每次老师被分配到的班级一致, 通过控制变量法来比较不同的排课方案在求解中学排课问题的效率.

实验硬件环境为 PC 机, 配置为 Inter(R)Core (TM)i3 CPU, 4G RAM; 软件平台为 Eclipse Mars.1, 数据存储在 Mysql 数据库中, 基于 Java 语言编程.

表 1 通过保持其他参数不变, 改变降温次数 T 和迭代步长 N , 即控制算法的外层和内层的循环次数. 经过重复 5 次实验取时间和最优解的平均值得到两种算法实验结果对比表.

表 1 两种算法运行效率对比表

(T, N)	基础方案		改进方案	
	时间(ms)	最优代价值	时间(ms)	最优代价值
(10, 10)	682.3	全部无解	661.2	1317.4
(30, 20)	815.4	1286.3(两次无解)	703.8	1293.8
(90, 30)	1068.8	1207.8	1061.6	1208.8
(200, 40)	1342.2	1201.8	1314.6	1199.0
(250, 50)	1672.6	1197.6	1375.0	1195.8

由表 1 中的数据可以看出当总循环次数在 100 左右时, 基础排课方案很难找到不发生冲突的课表, 而改进的分段模拟退火算法可以快速找到可行解; 改进的方案较基础方案在运行时间上也有提高; 两种算法在求得最优解结果上基本保持一致.

取 $(T, N)=(250, 50)$, 横坐标为 $(T \cdot N)$, 纵坐标为总代价值, 为便于观察前期两种算法迭代结果的不同, 可将横坐标取对数, 绘于图 1.

由图 1 可以看出本例中基于改进方案的模拟退火

算法在解决中学排课问题中收敛速度比基于基础方案退火算法快接近 10 倍;两种方案最后产生的最优解值相近.

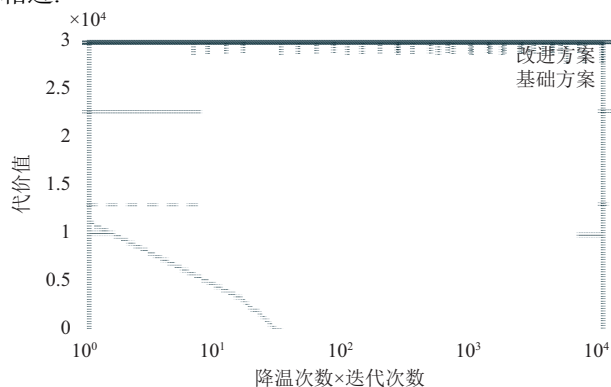


图1 改进前后方案寻优半对数曲线图

4 排课系统

在实际应用中,中学排课问题的软约束条件更加复杂.为了满足软约束(4),即多元化的排课需求,本系统基于SSM(Spring, SpringMVC, Mybatis)框架提供可以 and 用户交互的机制.

4.1 前端显示模块

前端利用 JQuery 编写数据录入逻辑以及校验逻辑的代码.如图2所示.

图2 前端显示

合法性校验主要包括:班主任数目=班级数目;输入不为空.

4.2 模拟数据模块

数据库连接池由 Spring 框架托管,并通过 Mybatis 框架从后台服务器的数据库中调用一组模拟数据,自动填充图2所示的所有表单字段,用于快速模拟排课流程.表单提交至 deal.action 后台路径,由后台处理并返回至 result.jsp 页面中.

4.3 前端调整模块

如图3所示在 result.jsp 页面中,鼠标悬停单元格以查看教师详情.利用 html5 的拖拽特性,允许用户手动调整课程单元格,单击单元格变色固定再次提交后台处理.

	1 班
星期一	吴峰
	胡素芬
	俞运良
	吴菲菲
	汤进进
	王峰
	吴峰
星期二	胡素芬
	吴菲菲

图3 调整并固定单元格

图3显示了得到一次后台处理结果之后,人为调整,即将两节“吴峰”的语文课程固定.前端记录调整信息并提交至后台 change.action, 后台模拟用户调整行为,并解决冲突、优化课表,最后返回结果至 result.jsp 页面中.重复以上动作,即可生成带有人工干预的个性化课表.

4.3 后台处理模块

将前端输入数据字段名称规范化,基于 SpringMVC 框架即可自动将传入的数据转换成 POJO,节省开发成本^[1].

根据不同的 action 使用不同阶段的模拟退火算法.对于 deal.action,应用第一阶段模拟退火算法生成可行解,应用第二阶段模拟退火算法指向性优化解,应用第三阶段模拟退火算法随机优化解;对于 change.action,只使用前两个阶段的模拟退火算法生成并优化解,因为第三阶段的退火过程会随机全局打乱课表,导致调整后课表其他单元格可能会发生较大的差异,即用户为了满足特殊需求调整的课表方案无需经过退火第三阶段处理.如图4是对图3调整后的处理结果.

5 结束语

分阶段的模拟退火算法在解决中学排课问题中,根据当前的课表状态,采用不同的邻域异化操作,可以快速生成可行课表,全局搜索优化课表并达到收敛.为了满足对于课表的更多复杂需求,基于本文中提出的改进后的模拟退火算法,设计了一套完整的可以与用

户交互的系统,并投放于生产环境中.

星期一	1班
	吴峰
	胡素芬
	俞运良
	吴菲菲
	汤进进
星期二	吴峰
	吴峰
	胡素
	吴菲菲

教师
课程
班级
周课
班主任
全周

图 4 后台处理结果

参考文献

1 Even S, Itai A, Shamir A. On the complexity of timetable and multicommodity flow problems. SIAM Journal on Computing, 1976, 5(4): 691–703. [doi: 10.1137/0205048]

2 赵玉新, Yang XS, 刘利强. 新兴元启发式优化方法. 北京: 科学出版社, 2013.

3 张健. 基于图论的高校排课系统实现. 重庆师范大学学报 (自然科学版), 2005, 22(1): 35–38.

4 许琦. 基于遗传算法的高校排课问题的研究[硕士学位论文]. 广州: 华南理工大学, 2012.

5 祝勇仁, 曹焕亚. 应用遗传算法求解排课问题. 计算机应用与软件, 2007, 24(12): 130–132, 141. [doi: 10.3969/j.issn.1000-386X.2007.12.051]

6 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, 2003, 37(4): 343–345, 350.

7 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, 2003, 18(2): 150–154.

8 赵惠怡. 基于蚁群算法的排课问题的研究[硕士学位论文]. 辽宁: 大连海事大学, 2007.

9 姜谦. 中小学排课系统的研究与设计[硕士学位论文]. 广州: 华南理工大学, 2010.

10 Kirpatrick S, Gelatt CD, Vecchi MP. Optimization by simulated annealing. Science, 1983, 220(4598): 671–680. [doi: 10.1126/science.220.4598.671]

11 吴婉楠. 基于 SpringMVC 和 MyBatis 框架的炒股比赛系统的设计与实现[硕士学位论文]. 南京: 南京大学, 2016.