

# 遗传算法在大学排课系统中的应用<sup>①</sup>

## Timetabling for University Classes with a Genetic Algorithm

胡义伟 郑金华 谢 勇 (湘潭大学 信息工程学院 湖南 湘潭 411105)

**摘要:** 大学排课是一个高度复杂的非线性优化问题,因此该问题的解决具有重要的理论意义和很好的应用价值。本文根据我国高校不完全学分制的需求,使用改进的遗传算法实现了新的排课系统。本系统有效地解决复杂教学进度、多节连排、学生个性化选课、分组教学、多教室类型、多教学区教学等问题,满足了新的需求,取得了较好的应用效果。

**关键词:** 遗传算法 排课 时间表问题

### 1 引言

排课问题的本质就是时间表问题——即 TTP (Timetabling Problem), TTP 问题是一类多元受限的不确定性问题,它决定一个从项目到资源的最优分配,使得总的代价最小且能满足  $K$  个条件<sup>[1]</sup>。

早在 1975 年,Shimon Even 等人就证明了 TTP 问题是 NP-Complete 问题<sup>[2]</sup>。传统的算法不适宜于求解 TTP 问题。

近年来,国内大量研究人员开始采用遗传算法解决高校排课问题,如电子科技大学的唐勇采用  $N$  维数组的遗传算法实现对课表的优化<sup>[3]</sup>,东南大学的业宁采用四维的染色体编码方式的遗传算法实现 TTP 问题求解<sup>[4]</sup>,上海交通大学的肖俊采用遗传算法很好的解决了多约束下的排课问题<sup>[5]</sup>。一系列的研究实践证明,遗传算法较好的解决了求解多约束排课问题的最优解问题,但这些研究基本都是小规模实验,以行政班级为最小授课对象,以一个自然周作为授课时间周期,具有较大的局限性。

随着高校教学改革的推进,在新环境下,现实的排课出现了许多新的问题,已有的研究很难解决这些存在的新问题:

(1) 教学进度复杂化:以前是施行周课表制(每周的课表相同)或者双周课表制(可区分单双周)。可现

在每一周的教学进度要求可能各不相同,甚至对于同一门课程的不同进度都存在授课教师、教学时长、教室类型要求的不同;

(2) 授课时间复杂化:以前授课都是两节连排,现在存在多节连排。比如:3 节连排,4 节连排,甚至 8 节连排的需求;

(3) 学生选课复杂化:以前是按行政班教学,现在施行不完全学分制或者分组教学等,学生可以选择自己个性化的课程学习,造成课程与课程之间关系更为复杂,提高了解决问题的难度;

(4) 授课场所复杂化:以前教学场所相对单一,现在多教学楼群、多教学区、多校区的情况很常见,教师跨校区授课和学生跨教学区上课都可能存在问题,在排课时需要考虑教师和学生存在交通上是否存在可行性。

随着教学中出现的新情况、新问题,以前的排课系统在功能上已不能满足各学校的需求,针对新情况研发出一套全新的功能强大的排课系统已是广大高校教务管理人员的迫切需求。实践研究证明,遗传算法应用于大规模、含离散变量等情况下的全局最优化问题是最合适的。因此,我们在研发排课系统中采用遗传算法进行排课优化。

<sup>①</sup> 基金项目:国家自然科学基金(60773047);教育部留学回国人员科研启动基金(教外司留[2005]546号);湖南省教育厅重点科研项目(06A074)

## 2 排课问题描述

### 2.1 数据结构定义

学生集:  $X = \{XS\}$ ;

教师集:  $Z = \{ZGH\}$ ;

开课集:  $K = \{(KKH, SKDX)\}$ , KKH 为开课号, SKDX 为授课学生集, 为 X 的子集;

教学进度集:  $J = \{(KKH, XH, ZGH, ZC, SC, JSLX)\}$ , 其中 KKH 表示开课号, XH 为同一开课号的进度序号, ZGH 表示教师职工号, ZC 为授课周次, SC 表示授课时长, JSLX 表示要求的教室类型;

时间集:  $T = \{\text{offset} \mid 0 \leq \text{offset} \leq 699\}$ , 一天分为 0、1、2、3、4 共五个节次, 前 4 个节次每个为 2 个课时, 即上午 2 个, 下午 2 个, 晚上第 4 节次为 3 课时。考虑到一周 7 天, 一个学期教学周有 20 周, 则一学期共包含 700 个节次;

教室集:  $R = \{(JSH, JSLX, JSRL, JXQ)\}$ , 其中 JSH 为教室号, JSLX 为教室类型, JSRL 为教室容量, JXQ 为教室所在教学区。

### 2.2 概念定义

课头相关:  $\forall K_i, K_j \in K, (K_i \setminus \{XS\} \cap K_j \setminus \{XS\}) \neq \Phi$ , 则称  $K_i$  与  $K_j$  课头相关, 记为  $K_i \leftrightarrow K_j$ 。

课头及进度:  $\forall J_j \in J, \exists K_i \in K$ , 使得  $J_j.KKH = K_i.KKH$ , 则称  $J_j$  是  $K_i$  的第  $J_j.XH$  个进度。

教学区可交通: 定义矩阵  $A[i][j]_{\{i \in XQ\} \times \{j \in XQ\}}$ ,  $A[i][j]$  表示第  $i$  个教学区和第  $j$  个教学区的交通代价,  $A[i][j] \geq 0$ , 0 为无代价。

### 2.3 约束及目标

我们的目标是使所有进度能被安排到合适的时间、地点, 没有冲突, 并满足一定的要求, 具体包括以下约束:

- (1) 所有的进度都被安排, 且只被安排一次;
- (2) 进度的周次要求必须满足;
- (3) 对于某开课的进度, 满足进度顺序约束, 即同一开课的小序号进度安排在大序号进度的前面;
- (4) 同一开课的不同进度不能安排在同一天;
- (5) 教室容量要大于或等于上课人数;
- (6) 教室的类型满足进度的教室类型要求;
- (7) 教室不冲突, 即同一时间同一教室只被安排一个进度;
- (8) 教师不冲突, 即同一时间每个教师只上一个

进度;

(9) 开课之间不冲突, 即同一时间安排的进度中没有两个进度的开课是相关的;

(10) 时长必须满足;

(11) 交通不冲突, 上午一二节和三四节之间、下午五六节和七八节之间, 学生跨教学区上课和教师跨教学区授课交通代价为 0。

在满足以上约束的条件下, 目标优化追求最大化, 优化目标如下:

- (1) 教室使用数尽可能少;
- (2) 使用的教室, 学生数和课时尽可能饱满;
- (3) 学生上课课时分布均衡;
- (4) 教师授课课时分布均衡;
- (5) 尽量满足一些教师的授课时间地点要求。

## 3 遗传算法设计

### 3.1 算法框架

我们在利用遗传算法进行算法设计时, 发现在经过传统遗传算法的交叉操作后, 出现了大量的冲突, 导致排课问题难于解决, 我们在尝试舍弃交叉操作后, 问题得到了较好的解决。同时, 为了降低问题规模和避免搜索陷入局部最优, 我们针对学校排课问题的实际需求, 对传统遗传算法进行了改进, 设计了一种在传统遗传算法基础上改进的进化算法, 以便于问题的求解。具体算法框架如下:

- (1) 对进度信息进行分类预处理;
- (2) 使用随机种子的贪心方法对种群初始化;
- (3) 变异生成新个体;
- (4) 如果有排课失败的开课, 则对这些开课尝试排课;
- (5) 使用重生操作产生下一代新种群;
- (6) 判断算法结束条件, 如果不满足, 则使用新种群代替老种群, 转(3); 否则, 算法结束。

### 3.2 预处理

虽然我们研发的是排课到节次和到学生个人的排课系统, 但在实际应用中, 大部分教学进度是有规律的, 可以通过对同一开课的进度进行分类, 以达到降低问题规模和避免同类进度排课结果分散导致课表难以打印等问题, 对同一开课的进度按以下算法分类:

- (1)  $S_i = \{J_i\}$ ;

(2) 如果  $S_i$ 、 $S_j$  中所有元素的 ZGH, JSLX, SC 都相同, 且所有元素的 ZC 两两都不相同, 则  $S_i = S_j \cup S_i$ , 并删除  $S_j$ ; 如果存在多个集合符合条件, 取包含进度序号最小的两个集合合并;

(3) 重复 (2), 直到不可再合并为止。

### 3.3 染色体编码

我们是用直接编码的方式进行染色体编码, 用一个结构数组表示对一类进度的安排。每个结构体包括四个域 (KK\_INDEX, JDL\_INDEX, JSH, OFFSET), KK\_INDEX 表示开课数组中的第 KK\_INDEX 的序号, JDL\_INDEX 表示第 KK\_INDEX 开课的第 JDL\_INDEX 进度类的序号, JSH 表示占用的教室编号, OFFSET 表示占用的起始时间, 取 0 到 34, 表示一周内从星期日第一、二节到星期六第九、十、十一节。

### 3.4 初始化

我们使用一种贪心算法对种群进行初始化操作, 使得初始的个体能很大程度上接近可行解。为了使得种群具有随机性, 我们对不同个体采用不同的随机种子达到这个目的。我们称之为随机种子的贪心初始化算法, 具体算法如下:

(1) 随机选择一个未安排的开课, 对其的若干进度类按所含最小进度序号升序排列, 然后依次安排。初始 OFFSET = 0;

(2) 对于进度类, 选择一个 OFFSET 最小、教室容量大于或等于上课学生数且最接近上课学生数、教室类型匹配的空教室作为这一类进度的安排结果。如果不存在这样的安排, 则本开课的所有安排作废, 本开课排课安排标记为失败;

(3)  $OFFSET = (OFFSET/5 + 1) * 5$ , 其中除法为整数除; 如果本开课还有其他进度类, 则如果  $OFFSET > 34$  则本开课的所有安排作废, 本开课排课安排标记为失败, 否则取下一个进度类, 转 (2), 直到这个开课的进度类全部安排为止;

(4) 如果所有开课都已安排或安排失败, 则结束, 否则转 (1)。

### 3.5 适应度函数设计

我们采用罚函数作为处理约束的办法, 对于所有的子目标 (极小优化函数) 通过加权聚集得到个体的目标函数; 对于所有的约束, 统计对应约束发生冲突的次数和对应的权值的加权和作为个体的约束函数的

值。适应度函数等于目标函数与罚函数之和, 其中罚函数是与约束函数和遗传代数相关的函数, 这样的设计使得初期个体可以在较大的空间内搜索, 而后期又能强化约束限制。

$$(1) \text{ 目标函数 } obj(\vec{x}) = \sum_{i=1}^{|I|} \omega'_i f_i(\vec{x})$$

$$(2) \text{ 约束函数 } con(\vec{x}) = \sum_{i=1}^{|J|} \omega_i c_i(\vec{x})$$

$$(3) \text{ 适应度函数 } fit(\vec{x}) = obj(\vec{x}) + kcon(\vec{x})/T$$

以上函数中,  $\vec{x}$  表示个体;  $\omega'_i$  表示第  $i$  个子目标权值;  $f_i(\vec{x})$  表示个体的第  $i$  个子目标的值;  $\omega_i$  表示第  $i$  个子约束函数的权值;  $c_i(\vec{x})$  表示个体的第  $i$  个子约束函数的值;  $k$  为一个比例系数,  $t$  表示当前遗传代数, 初始值为 1;  $T$  表示最大遗传代数。

### 3.6 变异和重排

由于课表合法性问题, 采用交叉操作往往会出现大量的冲突, 需要消耗大量精力去纠错或造成罚函数值非常大, 导致搜索难以进行, 为此我们对传统遗传算法进行了改进, 舍弃了交叉操作, 只使用变异操作产生新的个体。但单纯的变异不是因为基因变异太少造成搜索范围过小, 从而导致解陷入局部最优解造成早熟; 就是因为基因变异太多造成搜索范围过大, 从而导致搜索变成随机搜索, 收敛速度下降。为了平衡, 我们采用以下策略进行变异, 使得两者能够兼顾。

(1) 对于同一个父个体采用不同的变异算子和变异次数产生多个子个体;

(2) 尝试将父个体中安排失败的开课重新安排到子个体中;

(3) 对子个体分别计算子个体的适应度, 取最优的子个体作为父个体的子个体。

具体变异算子设计如下:

(1) 时间变异算子: 随机选择一个进度类, 保持 JSH 不变, 随机选择一个与原 OFFSET 不同的 OFFSET 作为新的时间安排;

(2) 教室变异算子: 随机选择一个进度类, 保持 OFFSET 不变, 随机选择一个与原 JSH 不同的但满足教室类型、容量要求的空教室作为新的教室安排;

(3) 对换变异算子: 随机选择一个进度类, 寻找另外一个进度类, 这两个进度类的教室的类型相同, 并且容量都符合两个上课人数的要求, 然后对换其时间

安排;

(4) 撤销安排变异算子. 随机选择一个开课, 将其所有进度类的安排撤销, 并将其放入安排失败开课队列的尾部。

### 3.7 重生

为了即避免搜索陷入局部最优解, 又要保持搜索的收敛速度, 我们采取了重生方法生成下一代种群。

假设  $f_p$  为父个体  $p$  的适应度,  $f_c$  为该个体执行变异等操作后生成的子个体  $c$  的适应度,  $p_r$  为重生率,  $p_r \in (0,1)$ , 重生具体操作如下:

- (1) 如果  $f_c \geq f_p$ , 则将新个体  $c$  放入下一代种群;
- (2) 如果  $f_c < f_p$  且个体  $p$  是上一代最优个体, 则将  $p$  放入下一代种群; 否则, 产生一个随机数  $r, r \in (0,1)$ , 如果  $r \leq p_r$ , 则将  $c$  放入下一代种群, 否则将  $p$  放入下一代种群。

## 4 结果与结论

我们以 C++ Build 作为开发工具, 采用 Oracle 10g 作为后台数据库管理系统, 通过采取初始化分类、舍弃交叉、采用变异策略等手段, 对传统遗传算法进行了改进, 设计和实现了基于改进的遗传算法的自动排课系统。系统实现后, 我们采用该系统对湘潭大学 2007 年下学期课表进行了编排, 此学期课程涉及的数

据包括 2091 门开课, 3726 个进度类, 227 个教室, 18 种教室类型, 2 个教学区, 23900 名学生, 20 周教学时间, 共 700 个时间单元。设置算法种群大小为 20, 最大遗传代数 2000, 重生概率为 0.3, 最终的优化可行解满足所有约束, 且较好的实现了课程、教师授课、学生上课的均衡分布, 以及教室的合理利用, 有效的配合了学校学分制选课、分组教学、分级教学等教学改革的实施, 取得的较好的应用成果。

### 参考文献

- 1 黄千平, 刘娟. 解“时间表问题”的启发式算法. 武汉大学学报(自然科学版), 1996, 42(1): 71-74.
- 2 SHIMON EVEN, ALON ITAI, ADI SHAMIR. On the Complexity of Timetable and Multicommodity Flow Problems. SIAM Journal on Computing. 1976, 5(4): 691-703.
- 3 唐勇, 唐雪飞, 王玲. 基于遗传算法的排课系统. 计算机应用, 2002, 22(10): 93-97.
- 4 业宁, 梁作鹏, 董逸生. 一种基于遗传算法的 TTP 问题求解算法. 东南大学学报(自然科学版), 2003, 33(1): 1-4.
- 5 肖俊. 遗传算法的工程应用. 计算机科学, 2005, 32(11): 247-248.