

# 混合启发式算法在排课问题上的应用<sup>①</sup>

詹亚坤<sup>1,3,4</sup>, 钟绍春<sup>1,2,3,4</sup>, 门慧勇<sup>1,2,3</sup>, 王叶静<sup>1,2,4</sup>

<sup>1</sup>(东北师范大学 理想信息技术研究院, 长春 130117)

<sup>2</sup>(东北师范大学 软件学院, 长春 130117)

<sup>3</sup>(教育部数字化学习支撑技术工程研究中心, 长春 130117)

<sup>4</sup>(吉林省教育软件重点实验室, 长春 130117)

**摘要:** 对排课问题做出了形式化描述, 提出了一种用于排课的混合启发式算法, 该算法合并使用了模拟退火和迭代局部搜索两种算法。先依据图着色算法产生初始可行解, 然后应用模拟退火算法寻找最优解, 为使算法更好地跳出局部最优, 实现全局搜索, 在模拟退火算法应用过程中, 迭代使用两个邻域, 标准邻域和双 Kempe 链邻域。实验结果表明, 此算法能够很好地提高解的质量。

**关键词:** 排课; 模拟退火; 迭代局部搜索; 启发式算法

## A Hybrid Heuristic Algorithm for Course Timetabling Problem

ZHAN Ya-Kun<sup>1,3,4</sup>, ZHONG Shao-Chun<sup>1,2,3,4</sup>, MEN Hui-Yong<sup>1,2,3</sup>, WANG Ye-Jing<sup>1,2,4</sup>

<sup>1</sup>(Ideal Institute of Information and Technology, Northeast Normal University, Changchun 130117, China)

<sup>2</sup>(School of Software, Northeast Normal University, Changchun 130117, China)

<sup>3</sup>(Engineering & Research Center of E-learning, Changchun 130117, China)

<sup>4</sup>(E-learning Laboratory of Jilin Province, Changchun 130117, China)

**Abstract:** This text makes a formal description for Course Timetabling Problem, and proposed a hybrid heuristic algorithm for course timetabling problem by combining simulated annealing with iterative local search algorithm. First, we generate an initial feasible solution based on graph coloring algorithm, and then apply the simulated annealing algorithm to find the optimal solution. In the process of annealing algorithm, we use two neighborhoods iteratively in order to escape from local optimum to search the global optimum. Computational results show that it significantly improves the quality of solution.

**Key words:** timetabling problem; simulated annealing; iterated local; heuristic algorithm

近些年来, 排课问题在研究和实践领域引起人们越来越多的兴趣。实质上, 排课问题由若干带有大量特征的事件组成, 这些事件将根据约束被分配到有限的时段和教室中。一般排课问题已经被证明是 NP 难题, 在这种情况下, 精确解只有在有限规模的排课问题中可能被得到。而启发式算法成为解决这类问题的有效方法, 包括进化算法, 模拟退火算法, 迭代局部搜索算法, 禁忌搜索算法, 蚁群算法等。

模拟退火以其理论完善, 局部寻优能力强, 计

算时间快等优点, 被越来越多地应用于排课问题。Abramson<sup>[1]</sup>, Melicio<sup>[2]</sup>, Elmohamed<sup>[3]</sup>尝试使用模拟退火算法解决排课问题, 但由于模拟退火的全局寻优能力差, 比较容易陷入局部最优。为了逃离局部最优及当发现有希望的搜索区域时自动进行精细搜索, 本文提出了一种将迭代局部搜索算法和模拟退火算法混合应用的排课算法, 在模拟退火算法中迭代使用了两种邻域, 包括标准邻域和双 Kempe 链邻域。

① 基金项目: 中央高校本科业务费专项资金(10SSXT130)

收稿时间: 2011-06-10; 收到修改稿时间: 2011-07-03

# 1 排课问题建模

## 1.1 问题描述

本文采用的排课问题是将一组课程安排到一个以星期为单位的时间表中，每门课程的每节课必须根据给定的约束条件分配给一个时间段和一个教室。在排课问题中，必须被满足的约束称为硬约束，可以被破坏的约束称为软约束。一个可行的课程表应该是所有课程的每节课都被分配一个时间段和教室，而没有硬约束的破坏。本文将约束条件定义如下<sup>[4]</sup>：

### (1) 硬约束

**H1:** 每门课程的每节课都必须被分配给一个时间段和一个教室。

**H2:** 任何两节课不能被分配同样的时间段和同样的教室。

**H3:** 在同一个课程组里的课程或者由同样教师讲授的课程不能被安排在同样的时间段，这里的课程组由共有相同学生的课程组成。也就是说，任何时间段不能有重叠的学生或教师。

**H4:** 如果一门课程的教师不能在给定的时间段上课，那么这门课不能被安排在这个时间段。

### (2) 软约束

**S1:** 对于每节课，听课学生的数量不应大于分配给这门课程的教室的容量。

**S2:** 任何一门课程的所有节课应该被分配在同一个教室。如果无法达到这样，那么被占据的教室数量应该尽可能少。

**S3:** 任何一门课程的所有节课应该被分布在给定的最小工作日内。

## 1.2 排课问题的形式化描述

本文的排课问题就是将一组课程  $C = \{c_1, \dots, c_i, \dots\}$  安排在一组时段  $T = \{t_{11}, \dots, t_{dp}, \dots\}$  和一组教室  $R = \{r_1, \dots, r_k, \dots\}$  中。每门课程  $c_i$  每周由  $\{cl_{i1}, \dots, cl_{ij}, \dots\}$  具体的授课组成。一个时段是一个由日和每日时域构成的数对，如  $t_{dp}$  时段即为  $d$  天（星期  $d$ ）的  $p$  时域，也就是  $t_{dp} = (d, p)$ 。另外，课程组是由共有相同学生的课程组成的集合。具体的符号和定义见表 1：

符号	描述
$c_i$	课程集 $C$ 里的第 $i$ 门课程， $c_i \in C$
$cl_{ij}$	课程 $c_i$ 在一周里的第 $j$ 节课。 $j < l_{i\max}$ （课程 $c_i$ 的每周最大课节数）
$t_{dp}$	一周中 $d$ 天（星期 $d$ ）的第 $p$ 个时段。

	$t_{dp} < t_{\max}$
$r_k$	教室集 $R$ 里的第 $k$ 个教室， $r_k \in R$
$rm_k$	第 $k$ 个教室的最大容量
$pc_i$	课程 $c_i$ 的任课教师
$gc_i$	课程 $c_i$ 属于的课程组
$sc_i$	参加课程 $c_i$ 的学生数量
$ava(pc_i, t_{dp})$	课程 $c_i$ 的任课教师在 $t_{dp}$ 是否是可用的，若 $ava(pc_i, t_{dp}) = 0$ 表示不可用，否则为可用。
$rm_i(S)$	在解 $S$ 中课程 $i$ 占据的教室数量
$dc_i(S)$	在解 $S$ 中课程 $i$ 占据的天数

$S$  是由  $(cl_{ij}, t_{dp}, r_k)$  组成的集合。

给定了这些符号，我们能够以形式化的方式描述排课问题。四个硬约束和三个软约束的形式化描述如下：

### (1) 硬约束

**H1:**

$$\forall cl_{ij}(c_i \in C, j < l_{i\max}), \exists (cl_{ij}, t_{dp}, r_k) \in S$$

**H2:**

$$\forall (cl_{ij}, t_{dp}, r_k), (cl_{i'j'}, t_{d'p'}, r_{k'}) \in S \text{ and } cl_{ij} \neq cl_{i'j'} \\ (t_{dp} \neq t_{d'p'}) \vee (r_k \neq r_{k'})$$

**H3:**

$$\forall (cl_{ij}, t_{dp}, r_k), (cl_{i'j'}, t_{d'p'}, r_{k'}) \in S \\ (gc_i \neq gc_{i'}) \vee (pc_i \neq pc_{i'})$$

**H4:**

$$\forall c_i \in C, \quad ava(pc_i, t_{dp}) = 0, \\ (cl_{ij}, t_{dp}, r_k) \notin S$$

### (2) 软约束

对于软约束，它们的代价函数如下：

**S1:**

$$\forall (cl_{ij}, t_{dp}, r_k) \in S \\ f_{sr}(cl_{ij}, t_{dp}, r_k) = \begin{cases} a(sc_i - rm_k) & sc_i > rm_k \\ 0 & otherwise \end{cases}$$

**S2:**

$$\forall c_i \in C, (cl_{ij}, t_{dp}, r_k) \in S \\ f_{cr}(c_i) = \beta(rm_i(S) - 1)$$

**S3:**

$$\forall c_i \in C, (cl_{ij}, t_{dp}, r_k) \in S \\ f_{cd}(c_i) = \begin{cases} \gamma(mwd_i - dc_i(S)) & mwd_i < (S) \\ 0 & otherwise \end{cases}$$

注： $mwd_i$  为给定的课程  $c_i$  的最小工作日

在软约束的形式化描述中， $\alpha$ ， $\beta$ ， $\gamma$  分别为相

应软约束代价函数的惩罚参数，根据实际对各约束的要求进行取值，在本文中，它们的取值分别为： $\alpha=1$ ， $\beta=0.8$ ， $\gamma=1.2$ 。

有了上面的形式化描述，给定一个候选解，我们能够根据公式(1)给定的代价函数  $f$  计算总的软约束代价。目标就是找到能使下面代价函数值最小的一个可行解。

$$f(S) = \sum_{(c_{ij}, t_{dp}, r_k) \in S} f(c_{ij}, t_{dp}, r_k) + \sum_{c_i \in C} f_{cr}(c_i) + \sum_{c_i \in C} f_{cd}(c_i) \quad (1)$$

## 2 算法描述

本文提出的混合启发式算法的基本思想是合并模拟退火算法和迭代局部搜索算法的优点特征。

模拟退火 (SA) 算法是一种通用的概率算法<sup>[5]</sup>，用于在大的搜索空间中寻找全局最优解。温度的大小决定着 SA 算法是进行广域搜索和还是局部搜索，若温度下降过快，SA 将很快从广域搜索进入到局部搜索，很容易陷入局部最优，为了跳出局部最优，往往需要增加 SA 算法的内外循环次数，这将大大增加算法进程的 CPU 时间。本文引进迭代局部算法来指导模拟退火算法跳出局部最优，通过干扰操作破坏局部最优解，并通过迭代邻域，搜索解空间中新的区域。

### 2.1 产生初始可行解

根据前面的问题描述，本文提出的产生初始可行解的方法主要分为两步：

首先，依据图着色方法<sup>[6]</sup>，为每个课程分配相应的时间段，使得共有相同学生或教师的课程不会被分配在同一时段。如图 1，节点表示课程，边连接共有相同学生（属于同一课程组）或教师的两个课程，颜色表示时间段。注意在着色过程中，同时要满足硬约束 H4，某些课程不能被着某些颜色（分配在某些时间段）。

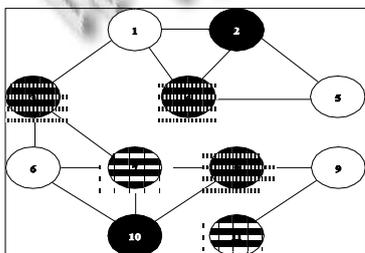


图 1 图着色方法产生的解

然后，为每个课程分配相应的教室。根据每个时

段（颜色）课程的数量，根据一定的启发信息（如参与课程的学生数量和教室容量）选择相同数量的教室，依次分配给相应的课程。如标记为白色的课程有四个，总教室数量为 7 个，则选择合适的四间教室，分配给这四个课程。

由上面的方法，得到的时间表满足前面提到的所有硬约束，故为一个可行解。

### 2.2 邻域结构

在邻域搜索过程中， $x \oplus mv$  表示解  $x$  经过一次移动  $mv$ ，产生了一个新的可行解。 $M(x)$  为从当前解  $x$  可以进行的所有一步移动的集合。 $x$  的邻域  $N(x)$  被定义为：

$$N(x) = \{x \oplus mv \mid mv \in M(x)\}$$

邻域搜索算法最重要的一个特征就是邻域的定义<sup>[7]</sup>，本文采用简单交换和双 Kempe 链交换两种距离移动，这两种方式产生的邻域  $N_1$  和  $N_2$  定义如下：

(1) 标准邻域  $N_1$ : 通过简单交换两个课程的位置（时段-教室）产生的邻域。交换两个课程  $C_1$  和  $C_2$ ，即将  $C_1$  移动到  $C_2$  到的位置，将  $C_2$  移动到  $C_1$  的位置。注意将课程移动到一个空位置是简单交换的一个特例，这种情况也包含在  $N_1$  邻域中。

(2) 双 Kempe 链邻域  $N_2$ <sup>[4,8]</sup>: 通过交换两条 Kempe 链而产生的移动。同 2.1 节对可行解的描述，每个排课问题实例被看成一张图  $G$ ，节点表示课程，边连接两个共有相同学生或教师的课程，颜色表示时段。在一个可行解中，任意两种颜色的节点及它们之间的边构成图  $G'$  的子图  $G'$ ， $G'$  中的一个连通分支即是一条 Kempe 链。一次 Kempe 交换就是通过交换两个时间段中的课程产生一个新的可行解，这些课程属于两条特定的 Kempe 链。更进一步形式化， $K_1$  和  $K_2$  是关于两个时段  $t_1 t_2$  的子图中的 Kempe 链，用  $(t_1 \setminus (K_1 \cup K_2)) \cup (t_2 \cap (K_1 \cup K_2))$  替代  $t_1$ ， $(t_2 \setminus (K_1 \cup K_2)) \cup (t_1 \cap (K_1 \cup K_2))$  替代  $t_2$ ，产生了一次 Kempe 交换。

如下面例子，图 1 为一次可行分配，随机选择属于两个不同时段， $t_1 = \{1,5,6,9\}$ ， $t_2 = \{3,4,8\}$ ，找到关于这两个时段的两条 Kempe 链， $K_1 = \{6,3,1,4,5\}$ ， $K_2 = \{8,9\}$ ，如图 2。则计算  $(t_1 \setminus (K_1 \cup K_2)) \cup (t_2 \cap (K_1 \cup K_2)) = \{3,4,8\}$ ， $(t_2 \setminus (K_1 \cup K_2)) \cup (t_1 \cap (K_1 \cup K_2)) = \{1,5,6,9\}$ ，交换 Kempe 链  $K_1$  和  $K_2$ ，得到  $t_1 = \{3,4,8\}$ ， $t_2 = \{1,5,6,9\}$ ，得到的分配如图 3。

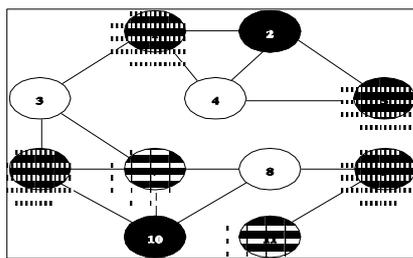


图 2 双 Kempe 链交换前的解

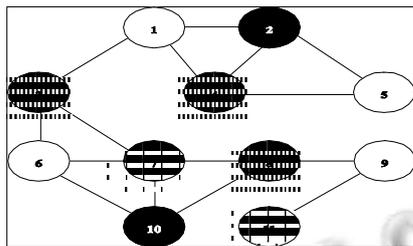


图 3 双 Kempe 链交换后的解

### 2.3 使用迭代邻域的模拟退火算法

在模拟退火算法搜索过程中，每个温度下，依次使用  $N_1$  和  $N_2$  这两个邻域，得到当前温度的热平衡状态。具体的说，如图 4，在温度  $V$  下，从初始解  $S_0$  出发，先以  $N_1$  邻域的方式进行  $m_1$  次移动，得到  $N_1$  邻域下的热平衡状态  $S_{11}$ ，再对  $S_{11}$  进行干扰操作，得到干扰后的解  $S'_{11}$ ，从  $S'_{11}$  出发，以邻域  $N_2$  的方式再进行  $m_1$  次移动，得到  $T_0$  温度下的最优解  $S_1$ ，降低温度，进入温度  $T_1$ ，从  $S_1$  出发再依次以  $N_1$  和  $N_2$  的邻域方式移动，得到  $T_1$  温度下的最优解，逐步降低温度，邻域移动，直到到达终止温度  $T_f$ ，得到全局最优解。

本模块算法流程如图 4 所示：

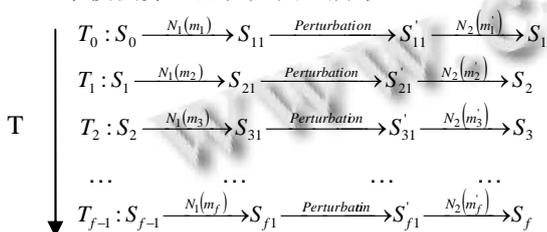


图 4 使用迭代邻域的模拟退火算法求解过程

本文中的终止温度  $T_f$  通过初始温度  $T_0$  和参数  $\mu$  计算得到， $T_f = T_0 / \mu$ ， $\mu \in [10, 1000]$ 。降温函数采用  $T_{k+1} = T_k \cdot \rho$ ， $\rho \in [0.98, 0.998]$ 。

我们的干扰操作由随机选择给定数量的简单交换

和 Kempe 交换的移动得到，这里至少有一个被移动的课程属于最高阶惩罚。具体来讲，在当前温度下，以邻域  $N_1$  方式得到最优解时，所有的课程根据它们涉及的软约束数量以递减的顺序排列。然后，一定数量的课程被从最高阶惩罚中选择出来。注意，选择高惩罚的课程实质上是由于这些课程对约束破坏大，即对解的质量影响大。

显然，干扰强度（记为  $\theta$ ）是迭代局部搜索算法中一个重要的因素，它决定了破坏前和破坏后两个解的质量差距， $\theta$  越大越容易跳出局部最优。在我们的实例中， $\theta$  随当前温度被适应性判断。由于温度越低，越容易陷入局部最优，因此干扰强度将随温度降低而逐渐增大，从而跳出局优，得到全局最优解。若当前温度为  $T_k$ ，则  $\theta = 1 - (T_k \cdot \mu - T_0) / (T_0 \cdot \mu - T_0)$ ，其中  $\mu = T_0 / T_f$ 。

### 3 实验结果和比较

为了验证本文提出的混合启发式算法的有效性，我们采用 C++ 语言实现了该算法，并选取了 20 个排课实例进行测试，实例部分信息如表 2：

表 2 20 个排课实例的部分信息

实例	课程数	教室数	每周上课天数	每天上课时间段数	课程组数	约束数量
1	80	17	5	10	24	486
2	120	14	5	10	43	424
3	76	15	5	10	35	477
4	210	34	5	10	51	562
5	30	8	5	10	12	234
6	98	23	5	10	21	499
7	256	35	5	10	65	612
8	72	12	5	10	21	410
9	87	19	5	10	21	376
10	78	18	5	10	19	345
11	154	23	5	10	46	456
12	134	21	5	10	42	423
13	95	20	5	10	16	389
14	112	24	5	10	34	467
15	123	27	5	10	41	412
16	145	32	5	10	46	445
17	46	10	5	10	11	245
18	67	13	5	10	15	278
19	187	28	5	10	27	422
20	235	31	5	10	45	550

本文算法采用的是标准邻域和双 Kempe 链邻域迭代使用的混合邻域结构, 为测试该邻域结构对算法的作用, 在算法实现过程中, 我们分别采用了标准邻域, 双 Kempe 链邻域, 混合邻域 (标准邻域和双 Kempe 链邻域迭代使用) 三种邻域方式。在三种邻域结构下, 对 20 个实例运行 30 次, 得到的平均代价函数值在迭代 1520 次后的变化如图 5:

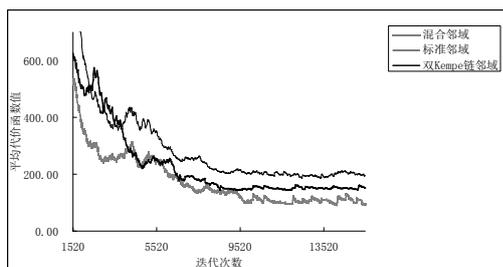


图 5 三种邻域结构下, 平均代价函数值随迭代次数变化的对比图

其中, 经过反复测试, 参数设置如表 3:

表 3 参数设置

初始温度 $T_0$	降温参数 $\rho$	温度参数 $\mu$
33.67	0.996	38.56

由图 5 可见, 随着迭代次数的增加, 采用混合邻域的算法得到解的平均代价函数值下降明显, 在迭代 9000 次左右后, 三种邻域结构下的平均代价函数值趋于稳定, 而混合邻域下的代价函数值为最低。

在邻域交替时, 对解进行干扰, 使其跳出局部最优, 是本文算法一个显著的优点。在算法分析中, 我们比较了加入干扰操作与未加入干扰操作两种情况下算法的性能, 得到的分析数据如图 6:

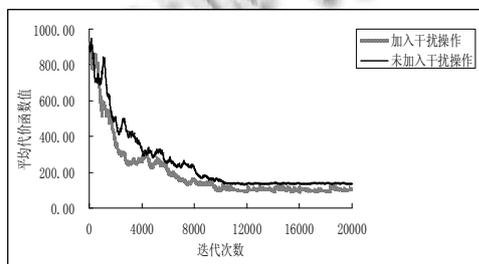


图 6 算法在加入干扰操作和未加入干扰操作情况下平均代价函数值的变化情况

由此可见, 加入干扰操作以后, 解的波动比较大,

在迭代 1000 次后, 未加干扰操作的算法得到的解基本无变化, 而加入干扰操作的算法得到的解仍旧在波动, 以寻找最优解, 直到经过若干次波动后, 仍没有优于最优解的解出现。

#### 4 结论

本文对简单排课问题做出了形式化定义, 给出了解决排课问题的具体方法。本文提出的混合启发式算法, 综合模拟退火算法和迭代局部搜索算法的优点, 解决了传统模拟退火算法在解决排课问题中全局寻优能力差的问题, 并且算法灵活, 求解效率高。

为了体现算法通用性的特点, 本文在实例测试中根据经验设置了统一的参数, 但在具体的实际应用中, 为了得到更满意的效果, 最好根据具体实例的特点设置适合的参数。

#### 参考文献

- 1 Abramson D. Constructing school timetables using simulated annealing: sequential and parallel algorithms. Management Science, 1991, 37: 98-113.
- 2 Melicio F, Caldeira J. Timetabling implementation aspects by simulated annealing. IEEE Systems Science and Systems Engineering, Beijing. Aceite, 1998.
- 3 Elmohamed S, Fox G, Coddington P. A comparison of annealing techniques for academic course scheduling. in Practice and Theory of Automated Timetabling (PATAT) II, Lecture Notes in Computer Science, Burke E. and Carter M. Eds. Berlin: Springer-Verlag, 1998. 1408. 146-166.
- 4 Lv ZP, Hao JK. Solving the course timetabling problem with a hybrid heuristic algorithm. Lecture Notes in Computer Science, Springer, 2008, 5253: 262-273.
- 5 Wikipedia. Simulated annealing-Wikipedia, the free encyclopedia. [http://en.wikipedia.org/wiki/Simulated\\_annealing](http://en.wikipedia.org/wiki/Simulated_annealing).
- 6 Burke E, McCollum B, Meisels A, Petrovic S, Qu RA. Graph-based hyper heuristic for timetabling problems. European Journal of Operational Research, 2007, 176: 177-192.
- 7 Gaspero LD, Schaerf A. Neighborhood portfolio approach for local search applied to timetabling problems. Journal of Mathematical Modeling and Algorithms, 2006, 5(1): 65-89.
- 8 Lewis R, Paechter B, Doria RO. Metaheuristics for university course timetabling. In Evolutionary Scheduling (Studies in Computational Intelligence. Dahal K. Kay Chen Tan P. Cowling (Eds.) Berlin: Springer-Verlag, 2007, 49. 237-272.