排课表问题的一种矩阵解法*

曾清海**
(电子科技大学宽带光纤传输与通信系统技术国家重点实验室　成都　610054)

【摘要】 对排课表问题进行了探讨，为排课表问题的软件实现提供了详细步骤。将排课表问题的求解转化为矩阵运算，阐述了矩阵运算的相关算法。该方法算法简单，易于软件实现。并对其正确性和可操作性进行了验证。

关 键 词　排课表；矩阵；　二叉树搜索；裂变
中图分类号　O157.6

排课表问题是在给定一周内各教师对各班课时数的情况下，排出一张无冲突的课表，并使一周的课时最少[1~3]。无冲突是指不存在一个教师被同时安排给一个以上的班级上课，或一个以上教师被同时安排给同一个班上课。

假设某校有 m 个班级，n 个教师。通常情况下，$m \leqslant n$ 。教师 j 一周共给班级 i 上 α_{ij} 节课，用 α_{ij} 构成 $m \times n$ 的矩阵。要排一张无冲突的课表，在矩阵运算中就是将该矩阵(称母矩阵)裂变为若干个每行、每列最多只有一个非零元素1的 $m \times n$ 阶矩阵(称子矩阵)之和[4]。而每个子矩阵就是为全校排出的一节课，非零元素所在的位置表示该列对应的教师被安排给该行对应的班级上一节课，所以问题就转化为将母矩阵裂变成尽量少的子矩阵之和。

1 矩阵运算

为把母矩阵裂变成尽量少的子矩阵，可采取如下步骤：

1) 由 α_{ij} 得 $m \times n$ 阶矩阵，如图1所示。

2) 将 $m \times n$ 阶矩阵加全为 $n \times n$ 完美方阵($n \geqslant m$)，如图2所示。其加全步骤如下：

(1) 将 $m \times n$ 阶矩阵添加成 $n \times n$ 阶方阵。在矩阵后面添置 $n-m$ 行，行的元素均取0。

(2) 求出行和与列和中的最大值 S，即

$$S = \max\left\{\sum_{i=1}^{m} a_{i1}, \sum_{i=1}^{m} a_{i2}, \cdots, \sum_{i=1}^{m} a_{in}, \sum_{j=1}^{n} a_{1j}, \sum_{j=1}^{n} a_{2j}, \cdots, \sum_{j=1}^{n} a_{mj}\right\}$$

(3) 找出行和最小的行 K，列和最小的列 L(若有多行/列的和最小，则在其中任取一个)。若 K 行的行和与 L 列的列和均等于 S，则结束，否则使 $\alpha_{kl} = \alpha_{kl} + 1$，再重复步骤(3)。由此便得到 n 阶 S 重完美方阵 $\boldsymbol{B}_0 = \{\beta_{ij}\}$。

3) 将完美方阵裂变成若干个排列方阵。排列方阵是指每行、每列都有且只有一个非零元素，且此非零元素为1的方阵。裂变的方法是先从 $\boldsymbol{B}_0$ 中裂变出一个，即 $\boldsymbol{B}_0 = \boldsymbol{B}_1 + \boldsymbol{I}_1$，$\boldsymbol{I}_1$ 即为第一个裂变出的排列方阵，$\boldsymbol{B}_1$ 为裂变后剩下的方阵，又从 $\boldsymbol{B}_1$ 裂变出第二个排列方阵 $\boldsymbol{I}_2$，即 $\boldsymbol{B}_1 = \boldsymbol{B}_2 + \boldsymbol{I}_2$，依此类推，直到 $\boldsymbol{B}_{S-2} = \boldsymbol{B}_{S-1} + \boldsymbol{I}_{S-1}$，$\boldsymbol{B}_{S-1}$ 也是一个排列方阵，可记作 $\boldsymbol{I}_S$，这样通过 $S-1$ 次裂变便得到了 S 个排列方阵。这 S 个排列方阵实际就对应 S 节课的排法。在排列方阵 $\boldsymbol{I}_i$ 中，元素1所在的行、列即表示该列对应的教师被安排给该行对应的班级上一节课(对加全的部分，对应的教师和班级实际无课)。

1999年10月13日收稿
* 电子部预研基金资助项目
** 男　27岁　博士生

由于是排列方阵，每行每列都只有一个1，其他元素均是0，所以不会出现冲突情况。当然，S 个排列方阵与 S 节课的对应关系可以任意。

$$\begin{bmatrix} \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1n} \\ \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ \alpha_{m1} & \alpha_{m2} & \cdots & \alpha_{mn} \end{bmatrix} \qquad \begin{bmatrix} \beta_{11} & \beta_{12} & \cdots & \beta_{1n} \\ \beta_{21} & \beta_{22} & \cdots & \beta_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ \beta_{n1} & \beta_{n2} & \cdots & \beta_{nn} \end{bmatrix}$$

图1　母矩阵　　　　图2　完美方阵

下面分析裂变过程。显而易见，S−1次裂变算法是完全一样的。不失一般性，本文给出第一步 $\boldsymbol{B}_0=\boldsymbol{B}_1+\boldsymbol{I}_1$ 的裂变算法如下：

1) 令 $K=0$，$M=\Phi, I=N=\{1,2,3,\cdots,n\}$；

2) $i=1$，在第 i 行找非零元素，取第1个非零元素(设所在的列的列号为 j)，令 $M_1=j$，$M=\{M_1\}$，$N=\dfrac{I}{M}$ (即 $N=I\cap\overline{M}$)；

3) $i=i+1$，当 $i=n$ 时，取 M_n=为 N 中的那个元素(只有一个元素)，$M=M\cup\{M_n\}$，该次裂变结束；否则在第 i 行找非零元素，若存在$\beta_{ij}>0(j\in N)$，令 $M_i=j$，$M=M\cup\{M_i\}$，$N=\dfrac{I}{M}$，返回该步骤3)；否则第 i 行的非零元素的列号必定属于 M，设有 k 个非零元素，列号从小到大依次为$0<p_1<p_2<\cdots<p_k$，再令 $t=0$；

4) $t=t+1$，在 M 中查找值为 p_t 的元素，设 $M_c=p_t$，则在第 c 行查找$\beta_{c,\,pt}$以外的所有非零元素，若存在$\beta_{c,\,j}>0(j\in N)$，令 $M_i=p_t$，$M_c=j$，$M=M\cup\{M_i\}$，$N=\dfrac{I}{M}$，返回步骤 3)；否则 c 行中$\beta_{c,\,pt}$以外的非零元素的列号属于 M，设有 d 个非零元素，列号从小到大依次为 $0<q_1<q_2<\cdots<q_d$。对 d 个非零元素进行类似于元素$\beta_{i,\,p1}$往下的搜索过程，直到找到一个非零元素的列号(设为 j)属于 N，并转入 5)；若搜索完$\beta_{i,\,p1}$节点以下的所有分支都没有搜索到列号属于 N 的非零元素，则重复 4)，否则转入 5)；

5) 设搜索过程成功搜索到该列所经过行的行号顺序为：i，p_1，$p_2,\cdots,p_k$，则 $M_i=Mp_1$，$Mp_1=Mp_2,\cdots,Mp_{k-1}=Mp_k$，$Mp_k=j$，$M=M\cup\{M_i\}$，$N=\dfrac{I}{M}$，再返回 3)；

以上第 3)、4)步的循环搜索过程的数据结构是一棵树。对构成环的搜索枝则终止该枝的搜索，该枝不成功。经过上述的裂变过程，即得到集合 $M=\{M_1, M_2,\cdots,M_n\}$，其中 $M_j\,(j=1,2,\cdots,n)$表示这次裂变对应课时为 j 班上课的老师编号。这样，分离出来的排列方阵就是第1行第 M_1列、第2行第 M_2列、⋯、第 n 行第 M_n列的元素为1，其他元素为0的 $n\times n$ 阶方阵。从 n 阶 S 重完美方阵分离出一个排列方阵后，完美方阵变成 n 阶 S−1重完美方阵。容易看出，以上第1次裂变过程对任意非零重完美方阵都适用。因为裂变过程搜索的是非零元素，而不涉及元素的大小，搜索枝的搜索过程如图3所示。图中，在第1~6行能找到满足条件的非零元素(实心圆)，要在第7行找一个满足条件的非零元素，如β_{72}满足(实心圆和三角形均表示非零元素)，但第2列的β_{42}已被安排在下一裂出排列方阵的第4行，所以就在第4行另找一个非零元素来替代β_{42}，结果找到β_{43}，而第3列也有β_{23}已被安排(在第2行)，所以又需在第2行另找一个非零元素来替代β_{23}，如此下去，直到找到β_{18}这个空闲列的非零元素。这样，在图中用三角形代替搜索过程途径的实心圆就形成一种新的到第7行的安排。

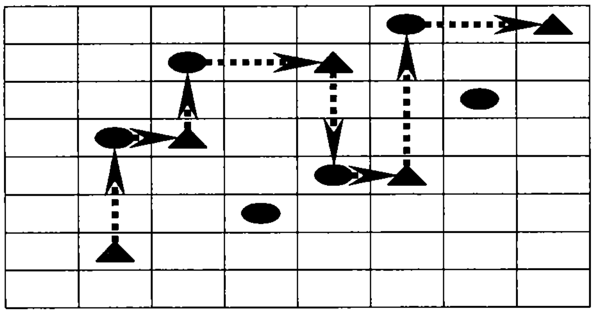

图3 搜索过程示意图

2 方法的验证

按步骤1的加全法每次加1，且只对行/列和小于 S 的行/列进行，这就不可能存在某行/列由于加全一次而导致行/列和超过 S。另外，整个矩阵无论按行加或按列加，总和是一样。所以，行和中有小于 S 的行，则必然有列和小于 S 的列，不可能出现加全一次后，加1元素所在行的行和不超过 S 但列和超过 S 或相反的情况，故 n 阶 S 重完美方阵是可获得的。

根据步骤 3)，在方阵 $\boldsymbol{B}_i$ 中按行顺序依次获得排列方阵 I_i 的各行元素，并且在搜索的过程中通过调整已安排行的元素避免了冲突，所以该搜索过程是正确的。

显然，只要证明裂变搜索过程最终能达到一个属于空闲列的非零元素。根据搜索过程可知，如果所有空闲列的非零元素都不在前 i 行(假设前 i–1行已经安排，已在安排第 i 行)，则肯定无法达到。但是，这种情况是不可能出现的。为方便说明，以第1次裂变过程为例，设前 i–1个被安排元素为$\beta_{11}, \beta_{22}, \cdots, \beta_{i-1\,i-1}$。由前面的假设可知$\beta_{kl}=0(k\leqslant i，l\geqslant i，k、l\in N)$，由行和可知

$$\sum_{k=i+1}^{n}\sum_{l=1}^{n}\alpha_{kl}=(n-i)S \Rightarrow \sum_{k=i+1}^{n}\sum_{l=i}^{n}\alpha_{kl}<(n-i)S$$

由列和可知

$$\sum_{k=1}^{n}\sum_{l=i}^{n}\alpha_{kl}=\sum_{k=i+1}^{n}\sum_{l=i}^{n}\alpha_{kl}=(n-i+1)S$$

因为上面两式是矛盾的，所以这种假设不成立。

另外一种情况，假设该搜索过程变成死循环，永远在前面排好的若干行中无休止的循环搜索。为方便，同样以第1次裂变过程为例，且不失一般性，设前 i–1个被安排元素为$\beta_{11},\beta_{22},\cdots,\beta_{i-1\,i-1}$，并设在前 $m(m\leqslant i-1)$行中循环。若搜索过程在前 m 行循环，则必然是在这 m 行中没有一个非零元素在第 m 列以后，也即$\beta_{kl}=0(0<k\leqslant m，m<l\leqslant n)$，则前 m 行的总和为

$$\sum_{k=1}^{m}\sum_{l=1}^{n}\alpha_{kl}=\sum_{k=1}^{m}\sum_{l=1}^{m}\alpha_{kl}=mS$$

而前 m 列的总和为

$$\sum_{k=1}^{n}\sum_{l=1}^{m}\alpha_{kl}=\sum_{k=1}^{m}\sum_{l=1}^{m}\alpha_{kl}+\sum_{k=m+1}^{n}\sum_{l=1}^{m}\alpha_{kl}=mS$$

故

$$\sum_{k=m+1}^{n}\sum_{l=1}^{m}\alpha_{kl}=0$$

则当 $m<k\leqslant n$，$0<l\leqslant m$ 时，$\alpha_{kl}=0$。

由搜索过程可知，搜索总是从第 i 行的某个非零元素开始，而以上假设的循环搜索过程显然无法从第 i 行的某个非零元素进入，因为第 i 行的前 m 列上的元素均为0。由此可见，这种假设也是不成立的。

由上面的证明可知，裂变搜索过程是可行的，可裂变出 S 个排列方阵。显然，S 是必须安排的最小课时数，所以也满足课时尽量少的要求。

3 结　论

本文使用矩阵运算方法解决了排课表问题。该方法算法简单，易于软件实现，并得到了正确性和可行性的验证。

参 考 文 献

1 楼世博，金晓龙，李鸿祥. 图论及其应用. 北京：人民邮电出版社, 1982

2 Dempster M A H. Two algorithms for the time-table problem. In combinatorial Mathematics and its Applications (Welshed D J A ed.). New York: Academic Press, 1971: 63~85

3 De-Werra D. On some combinatorial problems arising in scheduling. INFOR, 1970, 8: 165~175

4 赵冬梅, 郭耀煌, 陶章华. 多目标动态规划问题的非劣矩阵解法. 电子科技大学学报, 1998,27(2): 204~208

A Matrix Solution for Time-table Problem

Zeng Qinghai

(National key Lab of Optical Fiber Transmission and Communication Networks, UEST of China　Chengdu　610054)

Abstract　A problem of the course's time-table is discussed in this paper, and the detail software implementation steps are given. The time-table problem is translated into matrix operation, and the correlative arithmetic is expounded. The arithmetic presented in this paper is very simple and can be implemented conveniently by software. In the end, the correctness and maneuverability of the method are verified.

Key words　course's time-table;　matrix;　bin-tree;　fission