

浅谈基于浸润原理的事务安排算法

Brief Introduction to Schedule-making Algorithm Based on Soakage

何建强

He Jianqiang

(广西民族学院数学与计算机科学系 南宁 530006)

(Dept. of Math. & Comp. Sci., Guangxi Univ. for Nationalities, Nanning, 530006)

摘要 针对单个安排事务避免冲突的传统方法, 提出一种基于浸润原理的事物安排算法。该算法采用漫布式将各个事务分摊到各个位置, 根据条件计算权值, 并有意造成冲突, 从而发现冲突位置, 最后从最不可能冲突的位置入手, 逐步完成安排过程。该方法事务安排的复杂度为 $O(n^2 \times k \times x)$, 大大减少事务安排所需的时间。

关键词 事务安排 浸润原理 近似解

中图法分类号 TP301.6

Abstract According to the traditional method for single and conflict avoidance schedule-making, a new schedule-making algorithm based soakage is introduced. This algorithm assigning the affairs into different positions by distribution, and seeking the conflict positions by conflicts of intend to cause via the condition calculating value, then finishing the scheduling progressively from the positions impossible. This method made the complexity of scheduling course reduced and the time of scheduling course shortened.

Key words scheduling, soakage, approximate solution

事务安排是现实中常见的问题, 常见的有课程安排、考试日程安排、体育比赛赛程安排、会议安排等。凡是涉及时间、地点、参与主体的问题多数可归结为事务安排问题, 处理这类问题的关键是避免冲突和尽量满足事务的特定要求。但这类问题又是很难求出最优解的, 可能的组合方案与事务规模呈几何级数增长, 通过扫描所有组合的方法几乎不可能在有效的时间内得出方案。目前这类问题研究得最多的是排课问题, 处理方法主要是采用按优先权值先安排权值高的事务, 如果有冲突再调整。这类方法处理相对较容易, 但未必能求出较好的解。如果冲突较多, 则会距最优解有较大差距, 冲突调整过程相对也较麻烦, 通用性不强。

1 算法分析

基于浸润原理的事务安排算法一改单个安排事务避免冲突的传统方法, 采用漫布式将各

个事务分摊到各个位置，根据条件计算权值，并有意造成冲突，从而发现冲突位置，最后从最不可能冲突的位置入手，逐步完成安排过程。整个过程类似用各种颜色的墨水按各个位置的浓淡程度不同浸润于纸上，然后找出色彩最浓、色纯度最高的区域优先处理，最后过滤掉该颜色重复上述处理过程。具体处理过程如下。

一般事务安排要权衡和分配的因素很多，往往是一个多维分配问题，这大大增加处理的难度，为此应以最具有单一性、最关键的资源为基准，构造二维表。比如以时间为横轴、地点为纵轴构造二维表格(如图 1 上表)。表格的每个位置需要记录各个事务分配在该点的权重，因此该位置存放一个指针，指向一个记录各个事务在该点权重值的表记录(如图 1 下表)。这些记录 n 个事务在某点权重值的表记录有 $n + 2$ 个字段，每个字段记录一个事务在该点的“湿度”权值，这个权值以假设该事务分配于该位置为前提条件，通过综合扫描事务条件表获得。多余的 2 个字段记录该位置的最高湿度权值和重叠湿度权值(由各个事务的湿度权值相加获得)。

地点 时间	301 室	302 室	303 室	304 室
周一	1	5	9	13
周二	2	6	10	14
周三	3	7	11	15
周四	4	8	12	16

序号	最高湿度	重叠湿度	事务 1 湿度	事务 2 湿度	事务 3 湿度
1	300	750	250	300	200
2	100	170	20	50	100
3	200	310	200	100	10
4	250	500	250	150	100

图 1 二维表的构造和权重值的表记录

待第一次扫描完成后，各个事务位于各个表格内的湿度权值都已求出。然后找出有效权值 = (最高湿度权值 × 最高湿度权值 / 重叠湿度权值) 最高的位置(如图 1 箭头所示)，优先安排该位置最高湿度权值对应的事务于该位置，然后再次扫描二维表每个表格，过滤掉这个已安排事务的湿度，重叠湿度也要相应处理(如图 2 所示)，注意此时仍然要综合扫描事务条件表，求出排除上述已安排事务后的新湿度权值。继续找出其余表格中有效权值最高的位置，优先安排该位置最高湿度权值对应的事务于该位置。

重复以上过滤、搜索最高有效权值的过程，直至所有事务都已安排好为止。

从处理过程可以看出该处理方法可得到与最优解很近似的解，而且基本上很少出现因不可避免的冲突而不得不回溯调整的情况，大大简化了对各种复杂事务安排的处理，约束条件也允许更为自由灵活，有很强的通用性，可用于各种事务安排场合。

序号	最高湿度	重叠湿度	事务 1 湿度	事务 2 湿度	事务 3 湿度
1	300	500	0	300	200
2	100	150	0	50	100
3*	0	310	200	100	10
4	150	250	0	150	100

* 为已安排过的位置

图 2 已安排事务的湿度

2 算法的时间复杂度分析

设事务个数为 n , 可填充的位置数为 k , 约束条件数为 x , 则首次浸润时的循环次数为 $n \times k \times x$, 设已安排好位置的事务数为 y , 则每次搜索最高有效权值所需循环次数为 $k - y$, 每次过滤所需的循环次数为 $k \times x \times (n - y)$, 最多需要搜索和过滤 $n - 1$ 次, 故整个算法的时间复杂度为:

$$\sum_{y=0}^{n-1} (n - y) \times k \times x \approx O(n^2 \times k \times x).$$

以目前的计算机运行速度来看, 即使安排一个有上千个事务、上万个条件、数千个可安排位置的安排表, 所需时间也不过几个小时, 一般常见的应用可在几分钟内完成, 所得效果距理想值十分接近。

(责任编辑: 黎贞崇)

(上接第 161 页)

高效、快速.

5 结束语

文中提出的免解二次方程的圆形窗口裁剪算法主要特点是效率高. 例如在求出圆心到待裁剪线段的距离之后, 不仅用它来判断待裁剪线段与圆的位置关系, 还继续用它来求待裁剪线段和圆的交点. 另外, 求待裁剪线段和圆的交点时, 当求出第一个交点之后, 求第二个交点只须用中点坐标公式即可, 避免了第二次除法, 而常规求交方法或者文献 [3] 就无法做到这一点.

参考文献

- 1 郝恩(美), 贝克(美). 计算机图形学 C 语言版(英文). 北京: 清华大学出版社, 1998, 2.
- 2 黄文钧. 一种快速的参数化裁剪算法. 西北大学学报, 1999, 6: 29.
- 3 邹北骥. 基于矩形窗口裁剪的圆形窗口裁剪算法. 计算机工程与科学, 1998, 8: 20.

(责任编辑: 黎贞崇)