

基于图着色的事务调度算法的设计

刘智珩¹, 李腊元^{1,2}

(1 武汉生物工程学院 计算机与信息工程系, 湖北 武汉 430415;

2 武汉理工大学 计算机科学与技术学院, 湖北 武汉 430063)

摘 要: 将处理对象抽象转换为事务, 对于事务的调度问题提出了基于图着色思想的算法. 将事务以及之间的联系建立事务调度模型, 同时等价地转化为图着色问题, 通过对图中的顶点着色来实现具有冲突的事务的调度. 与一般图着色处理方式不同的是, 本算法思想采用了对节点进行着色的思想来实现事务调度. 基于图着色的算法的设计与实现使多事务多冲突问题得到解决, 并且最大程度满足事务执行所需各元素的特殊要求.

关键词: 图着色; 事务; 事务模型; 二分图; 调度算法

中图分类号: TP391

文献标识码: A

文章编号: 1000-7180(2012)04-0097-04

Based on Graph Coloring Affairs Schedule Algorithm for the Design

LIU Zhi-jun¹, LI Na-yuan^{1, 2}

(1 Department of Computer and Information Engineering, Wuhan Bioengineering Institute, Wuhan 430415, China;

2 Department of Computer Science and Technology, Wuhan University of Technology, Wuhan 430063, China)

Abstract: The objects convert abstract affairs, proposed algorithm for the affairs scheduling problem based on graph coloring ideas. The link between the affairs established affairs scheduling model, and convert into equivalent graph coloring problem, vertices on graph coloring to achieve the affairs of a scheduling conflict. Compared with general graph coloring approach is different, the nodes are shaded to the thought to achieve affairs scheduling. The algorithm based on graph coloring the design and implementation of more than many affairs issue has been resolved, and conflict affairs executed satisfy needed the special requirements of each element.

Key words: graph coloring; affairs; affairs model; binary chart; scheduling algorithm

1 引言

在实际生活中存在着在时间上或者处理程序上具有冲突的一系列事务, 如何合理安排这些事务, 形成事务时间表, 是活动中常见的困难工作. 在每一个时段内, 事务时间表的安排既受到不同事务参与者可得性的限制, 也受到事务执行场地可得性的限制.

图论为事务时间表问题提供了描述工具, 使用图着色问题作为数学模型来解决事务时间表安排问题^[1]. 在现实生活中使用图论、边着色等方式解决相关事务处理的算法思想比较广泛^[2], 例如在计算机

网络文件传输中, 每台计算机 X 有一个有限的数 $f(x)$ 的沟通端口. 对于每一个有大量的文件需被转移的计算机之间的配对的情况, 如何安排文件转移, 以尽量减少对整体转让过程的总时间很重要. 另外, 大学排课系统的优化问题、考试的考场的安排问题、商务会议时间的安排问题等等都可以在规定了某些特定条件后归结成为使用图论、着色问题来进行处理, 以达到最佳的事务安排的结果^[3].

2 问题的抽象

一般而言, 由现实的事务调度问题等价转化得

收稿日期: 2011-07-18; 修回日期: 2011-09-22

基金项目: 国家自然科学基金项目(60773211); 国家软件开发环境重点实验室开放基金(SKLSDE-2011KF-01); 湖北省教育厅科研项目(320114605, B20094002)

到的图着色问题,首先将处理的事务的基本对象确定对应到图的顶点以及边^[4-5].例如在大学中,考试考场的安排问题就是典型的事务时间表问题.对于监考,每个事务都由(教师,班级)二元组构成.同一个时间段内,如果两个事务拥有共同的教师或班级,则两事务冲突;此外还要考虑此时间段内事务执行场所(教室)的可得性.需要把有冲突或者场地不可得的事务调整到其他时段.那么使用 T 表示事务,则事务集合表示为: $\{T_1, T_2, T_3, \dots, T_n\}$, 资源使用 S 表示,资源集合表示为: $\{S_1, S_2, S_3, \dots, S_m\}$ ^[6]. 将具体问题抽象后事务 T 与事务所需的资源 S 之间的关系二分图表示如图 1 所示.

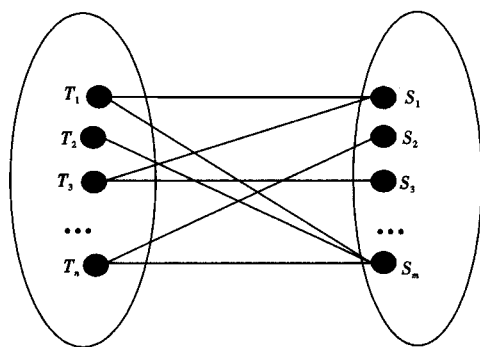


图 1 事务与资源关系的二分图

通过具体分析,将每个事务表示为一个顶点,所有事务构成顶点集 V ; 如果两个事务安排在同一时间段有冲突那么两顶点间拥一条无向边连接,反之,如果两顶点间没有边则代表两个事务安排在同一时间段无冲突,所有无向边构成边集 E ; 按此方法,把事务集转化为一个无向图 $G(V, E)$ ^[7]. 转换成图着色问题时,由于同一时间段内可得教室的数量是有限的,也就意味着每种颜色的允许使用次数是有限的. 用不同颜色代表不同时间段,给无向图 $G(V, E)$ 的顶点着色,保证任意一对有边连接的两顶点着不同颜色. 那么对于事务时间安排问题就转换成了图中找可行的着色方案的问题.

3 事务模型的建立

3.1 模型的建立

通过问题转换将处理的事务与图中的顶点和边按照一定的约定对应起来,以此建立基本的事务处理模型:

(1)形成事务集:给定事务集预定要完成的任务,给定构成事务的元素集合分别为 $A_1, A_2, \dots, A_n$.

定义 1 事务集合 T 是 $T_1, T_2, \dots, T_n$ 的笛卡尔积为 $T_1 \times T_2 \times \dots \times T_n = \{(t_1, t_2, \dots, t_n) | t_i \in T_i, i = 1, 2, \dots, n\}$.

定义 2 事务集合 $T(T_1, T_2, \dots, T_n)$, 其中的一个事务为 V , $V \in T$ 表示 V 是集合 T 的一个 n 元组. $v[T_i]$ 则表示元组 v 中相应于属性 A_i 的一个分量.

每个事务是笛卡尔集的一个 n 元组,即是从构成事务的元素集合的笛卡儿集中选取元素,形成事务. 多个事务的集合可以完成预定的任务,因而对于一项需完成的任务则转化得到一个满足预定任务的事务集 T .

(2)将事务设定为图中的顶点,即一个顶点表示一个事务集中的事务. 将任一对安排在同一时间段中有冲突的顶点作一条无向边,而在同一时间段内没有冲突的顶点不连接. 将完成预定任务的事务集转化为一张无向图.

定义 3 设 $G = \langle V, E \rangle$, $V = \{v_1, v_2, \dots, v_n | i = 1, 2, \dots, n\}$, $E = \{e_1, e_2, \dots, e_n | i = 1, 2, \dots, n\}$, 边 $e_i = (v_i, v_j) \in E, v_i, v_j \in V$.

其中每个顶点 v_i 对应定义 2 中的事务 V , $V \in T$, 表示 V 是集合 T 的一个 n 元组. v_i, v_j 是事务集中具有冲突的事务.

按照对图中节点和边的定义,事务集以及事务集之间的关系如图 2 所示. 在同一时间段有无冲突作为定义边的依据,将事务以及关系转换成图.

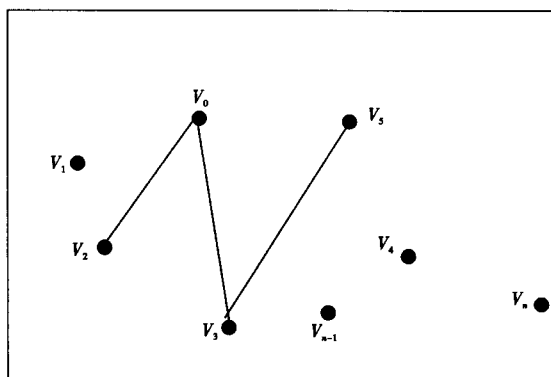


图 2 事务及关系转换的图

(3)对于事务的处理在定了基本的图的表示以后,使用对节点进行着色的方式来安排有冲突事务的处理,那么着色节点集合定义如下:

定义 4 设 $G = \langle V, E \rangle$, $C = \{c_i = v_j | v_j \in V, 1 \leq i \leq m, 1 \leq j \leq n\}$

其中 v_i 表示图中节点,即事务集中的一个事

务.对于着色节点集中的着色点的数量的上限值 m 是事务所需用资源 S 的个数和.

3.2 图着色过程的描述

图着色的主要思想是按照图中边的设定关系,将出现冲突的节点使用不同的颜色标记出来.设定预计给节点着色的总数目,使用不同颜色进行标记,即对于出现冲突的节点,每一个点对应一种不同的颜色.以颜色代表冲突节点.为上面所建立的事务模型得到的无向图进行着色.

(1)对每个顶点着色,着色的要求是每一顶点都着同一种颜色.由于顶点代表的是事务集中的具体事务,初始时不考虑边,即不考虑冲突情况,将颜色统一.

(2)针对冲突情况考虑边,任意一对有边连接的顶点改着不同颜色.

选从哪些边优先开始对顶点处理很重要,从实际角度来看,事务的安排,对于出现冲突的情况应该按照重要程度进行处理,因为越往后越难处理,如果再与其他必要条件冲突,就有可能无法安排.统计图中边的数量 m ,以及统计每个节点的度数,首先针对度数较大的节点开始考虑.其次引入权的思想,从带权图来看,权重高的边对应的顶点应该尽早着色;如果权相同,则可根据边所连接的顶点未处理情况排序.因此,选择正确的边即冲突情况优先处理可以提高算法成功率.

在带权图中,将边按权大小降序排列即按照冲突的情况的重要程度进行排序^[8]:

$E_w = \{e_i \mid e_i \in E, 1 \leq i \leq m\}$ (带权位排序的边的集合)

(3)实现能满足预定任务的特殊着色要求.得到可行的着色方案,每一个可行的着色方案对应事务集的一个可行的安排方案.

4 完整算法设计的描述

4.1 事务调度算法的描述

设 $G = \langle V, E \rangle$, $V = \{v_1, v_2, \dots, v_n\}$, $E = \{e_1, e_2, \dots, e_n\}$, $\lambda(v_i) = c$ 表示给顶点着色为 c , 具体的算法如下:

(1) 令 $c=1$, 对图中所有的顶点着色 c , 即 $\lambda(v_i) = 1 (1 \leq i \leq n)$.

(2) 令 $i=1$, 统计图中边的总数为 m , 在边信息中引入 q , 记录权位数据并将边集按照 q 的降序进行排列;统计图中各个节点的度,按照图 G 中的各节点度数的降序对节点进行排列,假设个数为 $k (1$

$\leq k \leq n)$;

(3) 取 $e_i \in E$, 比较与 e_i 连接两个顶点取度数较低的顶点着色即 $c=c+1$, 并将顶点进入着色顶点集合 $N(v_i)$

(4) $i=i+1$, 若 $i \leq m$, 则转移到(3).

(5) 着色完成, 根据着色方案得到事务集的一个安排方案.

一般而言,由现实的事务时间安排问题等价转化得到的图着色问题除了要求有边连接的顶点必须着色不同外,还有其他的硬性要求.这些要求主要是在实际问题中提出的事务与事务所占资源的问题.事务调度算法必须将这些约定考虑到处理中,这样所得的事务调度方案才合理,具有实际推广意义.

选取考试考场安排问题为例,将考试考场安排转化的图着色问题.对于监考,每个事务都由(教师,班级)二元组构成,所使用的资源就是考场即教室.由于同一时间段内可得教室的数量是有限的,那么对应的可着色的颜色数量的上限是确定的.若着同一颜色的点的数量超过此颜色代表的时间段可得教室的总数,就意味着监考事务单元着这个时间段内找不到教室可用,这个要求是硬性的,在调度中必须考虑.解决这些额外的要求可以通过冲突解决规定来调节冲突实现算法的可行性和合理性.也就是说,对于额外的硬性要求,在算法中设置相应的参数进行体现,或者作为判别是否着色的条件,使算法更符合实际需求.

4.2 事务调度算法的实现

在事务调度算法中 $E_count()$ 函数功能是统计图的边数, $V_degree()$ 函数的功能是统计节点的度数.定义 p_color 结构记录每个点的着色情况,记录对节点进行着色使用函数 $V_color()$ 函数, $error()$ 函数是错误处理模块,所有节点着色的结构数组的就是一种事务时间调度的方案.

下面是调度算法模块的核心算法:

```
for( $i=0, c=0; i \leq n; i++$ )
```

```
 $V\_color(*point, c);$ 
```

```
//给所有节点进行着色
```

```
 $m = E\_count();$  //统计图中的边的条数
```

```
for ( $k=1; k \leq m; k++$ )
```

```
{
```

```
if( $V\_degree(*point_i) >$ 
```

```
 $V\_degree(*point_j)$ )
```

```
point=point_j;
```

```

else
point=point_j;
V_color(*point,c++);
//选取边中度较小的节点进行着色
if c>MAX_C {error();break;}
//着色的数量超过资源总值,此方案不可行
}

```

MAX_C 的值是表示事务与事务所占资源的关系,对于部分事务调度算法还有其他硬性规定的可以将其转换成处理中额外判别条件. 核心算法处理后,将节点及其着色表示出那么一套事务调度方案就完成了. 不难看出,对于核心算法其理论时间复杂度 $O(n)$.

5 结束语

本算法的主要是通过具有冲突的顶点着色来实现事务的调度,主调度程序理论上的时间复杂度为 $O(n)$. 将实际问题转换成事务,并通过算法实现得到可行的事务时间表,使得每一事务都不与同时段执行的其他事务冲突,且每一事务在自己的执行时段内都有自己的执行场地. 通过将本算法应用到考试考场安排、商务会议安排等多事务处理的方案中,能得到合理的可行的时间表,而且最大程度满足

事务执行所需各元素的特殊要求.

参考文献:

- [1] Buckley F, Lewinter M. 图论简明教程[M]. 李慧霸, 王凤芹, 译. 北京:清华大学出版社, 2005.
- [2] 王世安, 王玉贤, 李太寿. 协同软件中的一种资源调度仿真算法研究[J]. 微电子学与计算机, 2010, 27(10): 163-165.
- [3] 毛影, 黄明和, 徐斌. 对启发式算法实现图着色的优化[J]. 江西师范大学学报: 自然科学版, 2009, 33(6): 709-711.
- [4] 范敏, 胡伟. 基于改进遗传算法的异构环境混合调度策略研究[J]. 微电子学与计算机, 2010, 27(8): 119-123.
- [5] 韩丽霞, 王宇平, 兰绍江. 基于有序划分编码的图着色算法[J]. 电子学报, 2010, 38(1): 146-150.
- [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[J]. 西安交通大学学报, 2003, 37(4): 1821.
- [7] 张健. 基于图论的高校排课系统实现[J]. 重庆师范大学学报: 自然科学版, 2005, 22(1): 35-38.
- [8] 孙波, 钟声. 匹配限制着色排课模型[J]. 计算机工程, 2008, 34(3): 111-113.

作者简介:

刘智璐 女, (1980—), 硕士, 讲师. 研究方向为计算软件.

(上接第 96 页)

- [7] 吴国锋. 基于 Specman Elite 的自动功能验证平台[D]. 西安: 西安电子科技大学, 2010.
- [8] Andreas S Meyer. Principles of functional verification[M]. hoiland: elsevier Science, 2004, 32-36.

作者简介:

张妮娜 女, (1986—), 硕士研究生. 研究方向为数字 IC 验证中的关键技术.

王万财 男, (1976—), 高级工程师. 研究方向为芯片验证高效率技术.

龚 衡 男, (1964—), 硕士, 教授. 研究方向为高速数字信号处理及片上数字系统(SOC)和嵌入式系统.