第17卷 第1期
2007年1月

计算机技术与发展
COMPUTER TECHNOLOGY AND DEVELOPMENT

Vol.17 No.1
Jan. 2007

R_时刻表及其应用

殷世民,张 磊,程家兴
(安徽大学 计算智能与信号处理教育部重点实验室,安徽 合肥 230039)

摘 要:求解R_时刻表是一个NP难问题,文中改进了一种有效的求解R_时刻表的时间规划算法并加以实现。该算法是建立在Allen的时间世界模型基础上,利用时间关系的关系矩阵方法来得到一致满足所有时间关系约束R_时刻表的一种算法。利用该算法实现了一个简化运动会项目的安排,验证了算法的有效性和实用性。

关键词:时间规划;R_时刻表;关系矩阵

中图分类号:TP301.6;O221 **文献标识码**:A **文章编号**:1673-629X(2007)01-0088-03

R_Time Table and Its Application

YIN Shi-min, ZHANG Lei, CHENG Jia-xing
(Ministry of Education Key Laboratory of Intelligent Computing and Signal Processing,
Anhui University, Hefei 230039, China)

Abstract: Solving the R_time table is an NP difficult question. An efficient algorithm for solving R_time table of temporal planning problem is improved and achieved in the text. It is based on the Allen temporal world model, which is an algorithm of using relational matrix method to obtain the R_time table to meet all time constraints. Using this algorithm realized a simplification of the arrangement of sport project and confirmed the validity of and usability of algorithm.

Key words: temporal planning; R_time table; relational matrix

0 引 言

时间规划是指:有一组与时间有关的事件的集合,并给定一组约束条件。要求在满足这些约束下,给出各事件发生、结束时间的时刻表。时间规划是人工智能所涉及的特殊规划领域,它是以时间关系约束作为推理依据的。

在生活中许多实际问题,如火车时刻表、运动会比赛项目安排、生产计划等都属于时间规划问题。所以已有一系列的研究成果,如Allen的时间世界模型。Allen规定了两个区间之间的13种不同的关系,13种关系能比较全面地反映各事件之间的时间关系。文中在Allen提供的描述时间规划问题模型基础上,改进了一种更有效的求时刻表的方法并加以实现。

1 时间关系的表示法

1.1 基于点的时间关系表示法

Allen[1]将事件与事件发生的时间区间I联系起来,于是讨论不同事件之间的关系就转化为对应事件区间之间的关系。一个区间由两个端点确定,于是两区间之间的关系就转化为四个端点位置关系,这称为基于点的时间逻辑。

设有两事件$I_1(a_1,b_1)$、$I_2(a_2,b_2)$,其中I_1的左端点a_1与区间(a_2,b_2)的关系有五种情况:$a_1 < a_2$、$a_1 = a_2$、$a_2 < a_1 < b_2$、$a_1 = b_2$、$a_1 > b_2$,同样b_1与区间(a_2,b_2)也是五种关系。假设把I_2固定,a_1与区间(a_2,b_2)关系的五种情况用x轴的1,2,3,4,5表示,同样y轴表示b_1与区间(a_2,b_2)的关系。x、y轴上的1,2,3,4,5分别表示$(-\infty,a_2)$、$[a_2]$、(a_2,b_2)、$[b_2]$、$(b_2,+\infty)$五个区间。则事件I_1和I_2之间的关系可用图1表示。

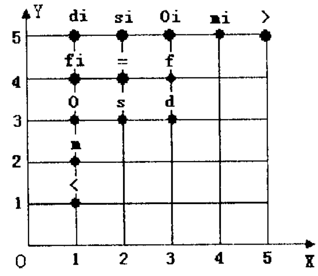

图1 时间关系平面表示

进一步,可将$X(y) = \{1,2,3,4,5\}$表示为$X = X_1 * X_2$的子集。其中$X_1 = \{-1,0,1\}$,而$-1,0,1$分别表示$(-\infty,a_2)$、$[a_2]$、$(a_2,+\infty)$三个区间,$X_2 = \{-1,0,1\}$,$-1,0,1$分别表示$(-\infty,b_2)$、$[b_2]$、$(b_2,+\infty)$三个区

收稿日期:2006-04-24
基金项目:教育部博士点基金(200403057002)
作者简介:殷世民(1979-),男,安徽全椒人,硕士研究生,研究方向为优化方法;程家兴,教授、博导,研究方向为智能计算与优化方法等。

间。同理 y 可表示成 $y_1 * y_2$ 的子集。这样任一关系约束均可表示成 $(X_1 * X_2) * (Y_1 * Y_2)$ 的子集形式。则称 $R(I_1, I_2)$ 为单成分的 $\Leftrightarrow$ 若存在 $A_1 \subset X_1, A_2 \subset X_2; B_1 \subset Y_1, B_2 \subset Y_2$,使得 $R(I_1, I_2) = [(A_1 \times A_2) \times (B_1 \times B_2)] \cap R$,其中 R 是所有13种关系在 $(X_1 \times X_2) \times (Y_1 \times Y_2)$ 上对应的点的全体。

1.2 时间关系的关系矩阵表示法

设 $R(i,j)$ 表示 I_i 与 I_j 之间的时间关系约束,$R(i,j) = ((X_1(i,j) \times X_2(i,j)) \times (Y_1(i,j) \times Y_2(i,j)))$,则 $\{R(i,j)\}$ 可用下面关系矩阵表示[2]:

$$\boldsymbol{M}(i,i) = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

$$\boldsymbol{M}(i,j) = \begin{pmatrix} X_1(i,j) & X_2(i,j) \\ Y_1(i,j) & Y_2(i,j) \end{pmatrix}, (i \neq j)$$

$$\boldsymbol{M} = \begin{bmatrix} M(1,1) & M(1,2) & \cdots & M(1,n) \\ M(2,1) & M(2,2) & \cdots & M(2,n) \\ \vdots & \vdots & \vdots & \vdots \\ M(n,1) & M(n,2) & \cdots & M(n,n) \end{bmatrix}$$

当 $\{R(i,j)\}$ 不是单成分时,不能惟一表示成上述矩阵 $\boldsymbol{M}$ 的形式。显然 $\boldsymbol{M}$ 是反对称矩阵。

2 单成分下的R_时刻表算法

2.1 相关概念

定义1(有序划分)　设 $T = \{X_1, X_2, \cdots, X_n\}$ 是一有限集合,$E = \{e_1, e_2, \cdots, e_m\}$ 是 T 的一个划分。在 E 中定义序关系如下:若 $x \in e_i, y \in e_j$,则 $x < y \Leftrightarrow i < j$,记作 $x < y(E)$。若 $x, y \in e_i$,则记作 $x \sim y(E)$。则称划分 E 连同上面定义的序为 T 的一个有序划分。

定义2(R_时刻表)　设 $\boldsymbol{M}$ 是时间规划的关系矩阵,$T = \{t_1, t_2, \cdots, t_{2n}\}$ 是对应的 n 个时间区间的端点时刻的集合。令 $E = \{e_1, e_2, \cdots, e_m\}$ 是 T 的一个有序划分,且满足 $x, y \in T, x < y(E) \Rightarrow -1 \in m(x,y); x, y \in T, x \sim y(E) \Rightarrow 0 \in m(x,y)$,其中,$m(x,y)$ 是 $\boldsymbol{M}$ 中对应于 x, y 的元素,则称有序划分 E 是 $\boldsymbol{M}$ 的一个R_时刻表。

2.2 R_时刻表算法

(1)主程序[3]。

C1:输入事件之间的关系,构造 $\boldsymbol{M}$($\boldsymbol{M}$ 为关系矩阵),若失败就停止;否则进入C2。

C2:求出 $\boldsymbol{M}$ 的简化矩阵,若失败,停止。不然得简化矩阵,仍记为 $\boldsymbol{M}$。令 $S = \varnothing, J = J(\boldsymbol{M})$,进入C3。

C3:若 $J = \varnothing$,则输出 S,称 S 为 $\boldsymbol{M}$ 的R_时刻表,算法结束;否则,进入C4。

C4:求 $\boldsymbol{M}$ 的所有非正行向量,并用 e 表示非正行向量的行号集合。若 $e = \varnothing$,失败,停止;否则进入C5。

C5:求 e 的相容子集,若失败,停止;否则得 e 的一相容子集,记为 d。进入C6。

C6:将 $\boldsymbol{M}$ 中 $j \in d$ 的第 j 行(列)删去,得到的子矩阵为 $\boldsymbol{M}_1$。令 $\boldsymbol{M} \leftarrow \boldsymbol{M}_1, J \leftarrow J/d, S \leftarrow (S, d)$,返回C3。

矩阵简化的实际意义:将开始时间或者结束时间一致的时间区间端点进行合并,并且将其用新的矩阵行代替。

(2)构造 $\boldsymbol{M}$ 矩阵。

C1:输入事件 i 与事件 j 之间的约束关系。进入C2。

C2:将 $R(i,j)$ 表示成 $(X_1(i,j) \times X_2(i,j)) \times (Y_1(i,j) \times Y_2(i,j))$,若失败则停止;否则修改 $\boldsymbol{M}$ 相应元素的值。进入C3。

C3:事件之间的约束关系是否输入完毕。否,返回第一步;是,进入C4。

C4:将 $\boldsymbol{M}(i,i)$ 的值用 $\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$ 矩阵代替。没有任何约束的事件 i 事件 j 对应的 $\boldsymbol{M}(i,j)$ 的值用 $\begin{pmatrix} -1,0,1 & -1,0,1 \\ -1,0,1 & -1,0,1 \end{pmatrix}$ 矩阵代替。

(3)求相容子集的算法。

输入 $\boldsymbol{M}$ 以及 $\boldsymbol{M}$ 的非正行向量的行号集合 e。令 $e = \{x_1, x_2, x_3, \cdots, x_m\}, J = J(\boldsymbol{M})$。

C1:i 从1开始,若 $i >= m+1$ 即失败,不然进入C2。

C2:令 $L_0 = \{x_i\}, L(X_i) = L_0$,若任意 $Z \in J$ 且 $Z \notin L(X_i)$,有 $m(y,z) \neq \varnothing$ 且 $\neq \{1\}$,进入C3,不然返回C1。

C3:$L_{j+1} = F(L_j) = \{y \mid x \in L_j, m(x,y) = \{0,1\}\}$。若 $L_{j+1} = \varnothing, d = L(X_i)$,称 d 是 e 中含有 x_i 的相容子集,成功,返回主程序第五步。若任意 $Z \in J$ 且 $Z \notin L(X_i), y \in L_{j+1}$,有 $m(y,z) \neq \varnothing$ 且 $\neq \{1\}$,则 $L(X_i) = L(X_i) \cup L_{j+1}$,返回C3,不然返回C1。

求相容子集的实际意义:排到某步,对余下的元素而言,有资格排在第一位的一定在非正行号集 e 中。当 $x \in e$,x 要排在第一位,那么所有 $m(x,y) = \{0,1\}$ 的元素 y,一定要排在第一位,此为求相容子集的理由。当然 e 的相容子集一般不惟一,当取不同的相容子集将得到不同的基本划分。

利用 $\boldsymbol{M}$ 矩阵中 $m(x,x) = 0, m(x,y) = m(y,x)$ 的特性并通过对刚获得的每个 L_j 检查是否满足约束,若满足跳到C3求 L_{j+1},不满足跳到C1,从而提前结束求解不满足条件的 $L(X_i)$,使得改进后的相容子集算法的时间复杂度减小一半。

2.3 算法的一些不足

算法的不足之处有[4]:

(1)很难为多组事件设置它们的约束条件,专业性强。

(2)此算法要求时间关系是单成分的,并且生活实例常需要设置较强的约束条件,常出现无解。

(3)算法的时间和空间复杂度较高,算法比较复杂。

3 R_时刻表在运动会中的应用

3.1 问题描述

用R_时刻表实现一个简化的校园运动会项目安排表。比赛项目包括径赛:100米预复决赛、200米预决赛、4×100米决赛、800米决赛、5000米决赛;田赛:跳远决赛、

铅球决赛、跳高决赛[5]。

并作如下规定：

(1)进行男子(女子)项目径赛的同时交叉进行女子(男子)项目的田赛。

(2)比赛分学女、学男、教工男、教工女四组。

(3)同一比赛项目尽量女子项目先比赛。

为了简化问题，排除了在多个比赛场所同时进行比赛的情况。将每个比赛项目看成不同的事件，共得 26 个事件。

1.学女 100 米预赛；2.学女 100 米复赛；3.学女 100 米决赛；4.学男 100 米预赛；5.学男 100 米复赛；6.学男 100 米决赛；7.学男 200 米预赛；8.学男 200 米决赛；9.学女 4×100 米决赛；10.学男 4×100 米决赛；11.教工男 4×100 米决赛；12.学女 800 米决赛；13.教工女 800 米决赛；14.教工男 800 米决赛；15.学男 1500 米决赛；16.学女跳远；17.教工女跳远；18.教工男跳远；19.学女铅球；20.学男铅球；21.教工女铅球；22.教工男铅球；23.学女跳高；24.学男跳高；25.学男标枪；26.教工男标枪。

由问题描述，为求得符合要求的 R_时刻表，首先分析出各个项目之间的约束条件。要考虑到运动员兼项，例如许多运动员既报短跑也报跳高跳远，所以尽量不要让这些项目同时比赛。在这里仅给出一些项目之间的约束条件：

$R(1,4) = \{m\}$；$R(1,18) = R(1,21) = \{=\}$；$R(4,13) = \{m\}$；$R(4,16) = \{s\}$；$R(4,17) = \{=\}$；$R(13,2) = \{m\}$；$R(13,6) = \{f\}$；$R(2,5) = \{m\}$；$R(2,19) = \{s\}$。

3.2 实验与分析

为验证算法的有效性，在 Windows 操作系统下，用 Delphi 实现了上述算法。

通过图 2 界面将所有事件之间的关系输入，利用上述算法得出事件发生的先后顺序。其中合并端点记录了每个事件端点在化简后的矩阵中所对应的端点号。R 时刻表给出了事件端点的排列顺序，即事件发生的先后等关系。最终可得运动会项目安排顺序如图 3 所示。

从事件的安排顺序来看，该算法能够较好地实现一般运动会的项目安排。但它只给出了一种安排顺序。

4 结束语

改进了求解时间规划问题 R 时刻表算法中相容子集算法，使得相容子集算法时间复杂度减小一半。并通过实验验证了该算法的实用性。

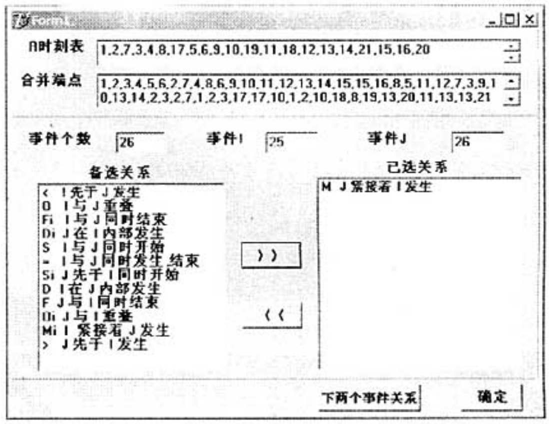

图 2 R_时刻表交互界面

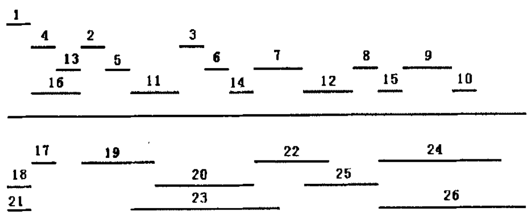

图 3 项目安排顺序

但该算法还具有一定的缺陷和不完善的地方。算法的主要不足是：很难设置事件之间的约束关系，只能排列出事件发生的先后顺序，不能给出事件发生的起始时间和结束时间，只适用于单成分。实际生活中许多问题是多成分的，关于多成分和涉及时间宽度约束的时间规划问题将在以后讨论。

参考文献：

[1] Allen J F. Maintaining knowledge about temporal interval[J]. Communication of the ACM, 1983, 26(11): 43 - 52.

[2] 张 钹，张 铃. 时间规划的关系矩阵法[J]. 计算机学报，1991, 14(6): 411 - 422.

[3] 张 钹，张 铃. 问题求解的理论及应用[M]. 北京：清华大学出版社，1990: 308 - 312.

[4] 方思行. 一种有效的 R_时刻表综合算法[J]. 华南理工大学学报：自然科学版，1995, 23(9): 43 - 46.

[5] 中国体育协会. 田径竞赛规则[M]. 北京：人民体育出版社，2002: 123 - 145.

(上接第 87 页)

[3] 和 洋. 天下维客使用手册[EB/OL]. 2006 - 05 - 10. http://www.allwiki.com/wiki/Help:%E7%BC%96%E8%BE%91.

[4] 飞 鸟. 10 大流行 Wiki 引擎[EB/OL]. 2005 - 06 - 02. http://www.allwiki.com/wiki/Wiki%E5%BC%95%E6%93%8E.

[5] Gustavo. JSPWik 引擎[EB/OL]. 2006 - 04 - 07. http://www.jspwiki.org/.

[6] Pagallo G. MediaWiki 引擎[EB/OL]. 2006 - 03 - 04. http://www.Mediawiki.org/.