

时间 Petri 网的随机模拟

潘理 杨勃

(湖南理工学院信息与通信工程学院 岳阳 414006)

摘要 模拟是 Petri 网进行系统分析的常用方法之一。由于时间 Petri 网采用时间区间来描述变迁实施的时间范围,因此变迁的实施时间点在区间内是不确定的。提出了时间 Petri 网的随机模拟方法。该方法在变迁开始使能时,根据某种随机分布确定实施区间内的实施时间点;然后基于模拟仿真的实验数据,运用统计分析方法及算法,构造时间 Petri 网状态类树,计算变迁实施区间及实施概率,为时间 Petri 网的系统模拟提供了一种新的探索途径。

关键词 时间 Petri 网,随机模拟,实施区间,状态类,实施概率

中图分类号 TP301 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.11.024

Stochastic Simulation of Time Petri Nets

PAN Li YANG Bo

(School of Information and Communication Engineering, Hunan Institute of Science and Technology, Yueyang 414006, China)

Abstract Simulation is a common method of system analysis for Petri nets. Time Petri nets describe firing time ranges of transitions by time intervals, thus the firing time points of transitions are uncertain in their time intervals. A stochastic simulation method for time Petri nets was proposed. When a transition becomes enabled, a firing time point in the firing interval of this transition is determined according to a certain random distribution, such as the uniform distribution. Based on experimental data by Petri net simulation, we presented an algorithm for constructing a state class tree of the Petri net model, and used a statistical analysis method for evaluating firing time intervals and its probabilities of transition sequences. The information obtained by the stochastic simulation is of good value for system analysis using time Petri nets.

Keywords Time Petri nets, Stochastic simulation, Firing interval, State classes, Firing probability

1 引言

时间 Petri 网是建模和分析实时系统最常用的形式模型之一^[1]。近 20 年来,关于时间 Petri 网的研究引起了国内外学者的高度关注,在图灵模拟能力、时间表达能力、状态类构造方法、模型检测等理论方面取得重要突破^[2-6],在先进制造、航空航天、生物医学、化工流程、网络通信等领域得到了广泛应用^[2,3,7-9],并且研究学者研制了界面友好、功能强大的时间 Petri 网工具软件^[10,11],有些已达到工业化应用水准^[11]。

时间 Petri 网使用时间间隔来描述事件的时间约束。已经证明,时间间隔比固定延时具有更强的表达能力^[2],因为时间间隔既能反映事件发生的不确定性,又能刻画事件发生的时间范围。在实际生产过程中,由于实时系统的离散特性和外界不确定因素的影响,一个事件的持续时间并不能准确地确定下来,但是事件结束的时间范围多数可以界定^[12-15]。此时,使用时间 Petri 网来建模和分析这类问题是一个合适的选择。

Petri 网的模拟或仿真(Petri Net Simulation),就是根据系统分析的目的分析系统各要素性质及其相互关系,建立能

描述系统结构或行为过程的时间 Petri 网模型,并在此基础上进行实验或定量分析,以获得正确决策所需的各种信息。模拟是 Petri 网进行系统建模和分析的常用方法,特别对于大型复杂系统,当某些问题无法运用现有 Petri 网理论方法进行求解,或者计算成本太高时,模拟将是 Petri 网进行系统分析的另一条有效途径。

时间 Petri 网的模拟需要考虑变迁的实施时间。但是,时间 Petri 网只是通过时间区间描述了变迁实施的时间范围,而对区间内如何选择实施时间点并没有精确的描述,即变迁实施时间点的选择是不确定的,这为时间 Petri 网的模拟实现带来了一定困难。目前,时间 Petri 网的模拟方法通常采取取区间上、下界值的方法^[10,11],但是这种方法忽略了区间内实施时间点的影响,并不能完整反映时间 Petri 网在这些时间点上行为特性。因此需要设计某种合理的规则来选择每个变迁的实施时间点。

本文提出时间 Petri 网的随机模拟方法。该方法在变迁开始使能时,根据某种随机分布确定实施区间内的变迁实施时间点,解决实施时间点的模拟实现问题;然后基于模拟实验的数据,运用统计分析方法,得到状态类、实施区间、实施概率

到稿日期:2015-08-22 返修日期:2016-03-05 本文受国家自然科学基金(61300039),湖南省教育厅科学研究重点项目(15A079),湖南省科技计划项目(2014GK3026,2013GK3099)资助。

潘理(1975—),男,博士,副教授,主要研究方向为 Petri 网、形式化建模与优化, E-mail: panli_hnist@gmail.com; 杨勃(1974—),男,博士,副教授,主要研究方向为模式识别。

等重要信息,为时间 Petri 网的系统模拟提供了一种新的探索途径。本文第 2 节给出时间 Petri 网的基本定义和行为描述,并提出基于均匀分布的随机模拟方法;第 3 节利用随机模拟得到的实验数据,提出状态类构造算法、变迁实施区间计算方法、变迁实施概率计算方法;最后总结全文。

2 时间 Petri 网随机模拟方法

2.1 时间 Petri 网

假设 R 是实数集, R^+ 是非负实数集, N 是自然数集 $\{0, 1, 2, \dots\}$ 。给定 $a, b \in R$ 且 $a \leq b$, 定义 $I = \{x \in R \mid a \leq x \leq b\}$ 为实数闭区间, 记为 $I = [a, b]$ 。用 $\uparrow I (\downarrow I)$ 表示区间 I 的上(下)界。用 $IR (IR^+)$ 表示所有(非负)实数闭区间的集合。

定义 1^[2] 一个时间 Petri 网是一个六元组 $TPN = (P, T, Pre, Post, M_0; SI)$, 其中:

- (1) P 是库所集;
- (2) T 是变迁集, 且 $P \cap T = \emptyset$;
- (3) $Pre \in N^{P \times T}$ 是向前关联矩阵;
- (4) $Post \in N^{P \times T}$ 是向后关联矩阵;
- (5) $M_0 \in N^P$ 是初始标识;
- (6) $SI: T \rightarrow IR^+$ 是变迁的静态实施间隔映射函数, 对 $\forall t \in T$, $SI(t)$ 表示变迁 t 的静态实施间隔。

$Pre(p, t) = i (i > 0)$ 当且仅当存在一条从库所 p 到变迁 t 的权值为 i 的弧; $Pre(p, t) = 0$ 当且仅当不存在从库所 p 到变迁 t 的弧。 $Pre(t) \in N^P$ 表示 t 的输入库所多重集, 即对应向前关联矩阵 Pre 中 t 列所对应的向量。对 $Post(p, t)$ 和 $Post(t)$ 也类似定义。标识 $M \in N^P$ 是库所集上的多重集, 可表示为一个 $|P|$ 元向量, 第 i 个元素表示库所 p_i 中的令牌数。

定义 2^[2] 一个变迁 $t \in T$ 在标识 M 下被称为是使能的, 当且仅当 $Pre(t) \leq M$ 。

令 $En(M)$ 表示标识 M 下所有使能变迁的集合。

定义 3^[2] 如果 t_f 在标识 M 下是使能的, 那么 t_f 可以实施并产生一个新的后继标识 M' , 且 $M' = M - Pre(t_f) + Post(t_f)$ 。

用 $Newly(M, t_f) = En(M - Pre(t_f) + Post(t_f)) \setminus En(M - Pre(t_f))$ 表示在标识 M 下实施 t_f 后(在新标识)新使能变迁的集合。注意一个变迁在实施 t_f 后丧失使能, 但在新标识又重新使能, 将作为新使能变迁看待。

系统的动态行为可由状态和状态迁移规则描述。

定义 4^[2] 一个时间 Petri 网的状态是一个二元组 $s = (M, f)$, 其中: 1) $M \in N^P$ 是标识; 2) $f: T \rightarrow IR$ 是一个动态实施间隔函数。 $\forall t \in T$, $f(t)$ 表示 t 相对于当前状态的实施时间间隔。

时间 Petri 网的初始状态 $s_0 = (M_0, f_0)$, 其中 M_0 是初始标识; $\forall t \in En(M_0)$, $f_0(t) = SI(t)$ 。

时间 Petri 网的状态迁移规则可以由带标号的迁移系统 (Labeled Transition System) 描述。

定义 5^[4] 一个带标号的迁移系统是一个四元组 $L = (S, S_0, \Sigma, \rightarrow)$, 其中:

- (1) S 是状态集;
- (2) $S_0 \subseteq S$ 是初始状态集;
- (3) Σ 是描述事件的标号集;
- (4) $\rightarrow \subseteq S \times \Sigma \times S$ 是迁移关系。

定义 6^[4] 一个时间 Petri 网 $TPN = (P, T, Pre, Post,$

$M_0; SI)$ 的形式语义定义为一个带标号的变迁系统 $L = (S, S_0, \Sigma, \rightarrow)$, 且使得:

- (1) $S \subseteq N^P \times IR^T$ 是 TPN 的状态集;
- (2) $s_0 = (M_0, f_0)$ 是 TPN 的初始状态;
- (3) $\Sigma \subseteq T \times R^+$ 是标号集;
- (4) $\rightarrow \subseteq S \times \Sigma \times S$ 是迁移关系, 且 $\forall t_f \in T, \forall d \in R^+$,

$$(M, f) \xrightarrow{(t_f, d)} (M', f') \text{ iff}$$

$$\begin{cases} t_f \in En(M) & (a) \\ \downarrow f(t_f) \leq d \leq \uparrow f(t_f) & (b) \\ \forall t \in En(M), d \leq \uparrow f(t) & (c) \\ M' = M - Pre(t_f) + Post(t_f) & (d) \\ \forall t \in En(M') & \\ f'(t) = \begin{cases} SI(t), & \text{if } t \in Newly(M, t_f) \\ [\max(0, \downarrow f(t) - d), \uparrow f(t) - d] \text{ o. w.} & \end{cases} & (e) \end{cases}$$

状态迁移是指从状态 s 流逝一段时间 d , 然后实施变迁 t_f 到达另一个状态 s' 。状态迁移关系 (a) 表明 t_f 在 M 使能; (b) 保证流逝时间 d 位于 t_f 的动态实施间隔内; (c) 确保 d 不超过所有使能变迁的实施时间上界; (d) 计算新状态的标识; (e) 计算新状态下使能变迁的动态实施间隔, 即新使能变迁的动态实施间隔为其静态实施间隔, 其他使能变迁的动态实施间隔则减去流逝时间 d 并截取非零区间。

2.2 随机模拟实验方法

随机数在随机模拟设计中扮演着十分重要的角色。根据时间 Petri 网的定义可知, 时间 Petri 网的变迁实施必须在其实施区间内发生, 即如果变迁 t_i 在时刻 τ 开始使能, 那么它的实施时间一定落在 $[\tau + \downarrow SI(t_i), \tau + \uparrow SI(t_i)]$ 区间内。如何产生在实施区间内的随机数, 是时间 Petri 网随机模拟的关键因素之一。

常见的连续型分布有: 均匀分布、指数分布、正态分布等。由于指数分布的区间是 $[0, \infty)$, 正态分布的区间是 $(-\infty, +\infty)$, 根据这些分布产生的随机数不能保证一定落入所需的实施区间, 因此采用均匀分布是一个合理的选择^[16]。当然, 也可以使用截断正态分布 (Truncated normal distribution) 或截断指数分布^[17, 18] 来保证生成的随机数落入所需区间。

本文将采用均匀分布生成实施区间的随机数。设随机变量 X 的概率密度函数为:

$$f_X(x) = \begin{cases} \frac{1}{b-a}, & a \leq x \leq b \\ 0, & x < a \vee x > b \end{cases}$$

则称 X 服从区间 $[a, b]$ 上的均匀分布, 记作 $X \sim U(a, b)$ 。其相应的累积分布函数为:

$$F_X(x) = \int_{-\infty}^x f_X(u) du = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & x \geq b \end{cases}$$

假设时间 Petri 网中每个变迁的实施时间遵循均匀分布。用 $X_1, X_2, \dots, X_n$ 分别表示变迁 $t_1, t_2, \dots, t_n$ 的实施时间随机变量。因此, 变迁 t_i 的概率密度函数为:

$$f_{X_i}(x_i) = \begin{cases} \frac{1}{\uparrow SI(t_i) - \downarrow SI(t_i)}, & x_i \in SI(t_i) \\ 0, & \text{otherwise} \end{cases}$$

基于均匀分布的时间 Petri 网随机模拟步骤如下。

(1)在某一状态 s , 计算状态 s 的使能变迁集(根据定义 2), 采用均匀分布对每个新使能变迁 t_i 随机生成一个随机数 $u_i \in SI(t_i)$, 以表示该变迁(相对于使能时刻)的实施时间。

(2)每个变迁的实施随机数 u_i 随着时间的流逝同步减小。一旦 u_i 到达 0, 就立刻实施变迁 t_i 。因此, 具有最小实施时间的变迁总是优先实施(时间竞争策略)。

(3)变迁实施后, 计算下一个状态 s' (根据定义 6)。

(4)在状态 s' , 若变迁 t_i 丧失使能, t_i 将失去实施权, 抛弃与 t_i 关联的随机数 u_i (也可设置 $u_i = \infty$)。

(5)继续重复步骤(1)–(5), 直到使能变迁集为空(终止状态)。

实验案例选取了一个简单的时间 Petri 网模型, 如图 1 所示。尽管该模型简单, 但包含了顺序、选择、并发等不同关系, 具有一定代表性。

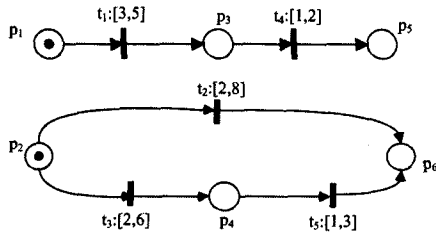


图 1 一个简单时间 Petri 网模型

实验数据为一系列带时间的变迁实施序列。例如：

($t_2, 2.3868$) ($t_1, 3.4636$) ($t_4, 1.486$)
 ($t_3, 3.3694$) ($t_5, 1.0370$) ($t_1, 1.2675$)
 ($t_4, 1.8214$)

每条时间序列数据可表示为 $(t_{r1}, tm_{r1})(t_{r2}, tm_{r2}) \cdots (t_m, tm_m)$ 的形式, 它表明 t_{r1} 在 tm_{r1} 时刻实施, t_{r2} 在 $tm_{r1} + tm_{r2}$ 时刻实施, $\cdots, t_m$ 在全局时间 $\sum_{i=1}^m tm_{ri}$ 实施。

3 随机模拟统计分析

对随机模拟实验得到的数据进行统计分析, 构造时间 Petri 网状态类图, 计算变迁序列实施区间和变迁序列实施概率。

3.1 状态类构造

对图 1 所示时间 Petri 网进行随机模拟, 可以获得 Petri 网的变迁实施序列集合。如果模拟次数足够多, 可获得该网的变迁实施序列完全集。对图 1 所示时间 Petri 网进行 10^4 次模拟实验, 统计得到全部的 9 条变迁实施序列, 如表 1 所列。

表 1 图 1 中 TPN 的全部变迁实施序列

序号	变迁实施序列	序号	变迁实施序列
1	t1 t2 t4	6	t2 t1 t4
2	t1 t3 t4 t5	7	t3 t1 t4 t5
3	t1 t3 t5 t4	8	t3 t1 t5 t4
4	t1 t4 t2	9	t3 t5 t1 t4
5	t1 t4 t3 t5		

从变迁实施序列完全集出发, 可以构造出时间 Petri 网的状态类树。记 $S = \{s_1, s_2, \cdots, s_n\}$ 为全部 n 条变迁实施序列的集合, s_i 为第 i 条实施序列。状态类树以初始状态类 C_0 为根节点。若实施变迁 t_k , 使状态类 C_i 转换到状态类 C_j , 则标记节点 C_i 到节点 C_j 的边为 t_k 。状态类树构造算法 Seq2Ctree 如算法 1 所示。

算法 1 Seq2Ctree 构造算法

```

input: S
begin
  初始化状态类树  $CT = (V, E), V = \{C_0\}, E = \emptyset // C_0$  是根节点
  for each  $s_i \in S$ 
     $C = C_0$  // 当前状态类初始化为  $C_0$ 
    for each  $t \in s_i$  // 按  $s_i$  中实施序列的顺序取变迁  $t$ 
      if  $t$  是  $CT$  中  $C$  到下一个状态类  $C'$  的边
         $C = C'$  // 转移到下一状态类
      else
        生成新的状态类节点  $C'$ 
         $V = V \cup \{C'\}$ 
        标记  $t$  为  $C$  到  $C'$  的边
         $E = E \cup \{t\}$ 
         $C = C'$  // 转移到下一状态类
    end
  end
end
return 状态类树  $CT$ 
end
  
```

Seq2Ctree 算法从实施序列集 S 构造状态类树 CT 。对 S 中的每条实施序列 s_i , 以根节点 C_0 为当前节点, 依次取 s_i 中的实施变迁 t 。若在 CT 中 t 是当前节点 C 的实施边, 则 t 在类 C 的实施已经标记在 CT 中, 转移到下一状态类继续; 否则, t 在 C 的实施没有标记在 CT 中, 则生成新的状态类 C' , 并标记 t 为从 C 指向 C' 的边, 直到读取完实施序列 s_i 的所有变迁。继续读取 S 中的下条实施序列, 如此重复, 直至所有实施序列读取完毕。

假设变迁实施序列的数目为 m , 变迁实施序列中变迁的数目为 n , 则 Seq2Ctree 算法的时间复杂度为 $O(mn)$ 。以图 1 中 TPN 的变迁实施序列集为输入, 使用 Seq2Ctree 算法, 构造获得如图 2 所示的状态类树。

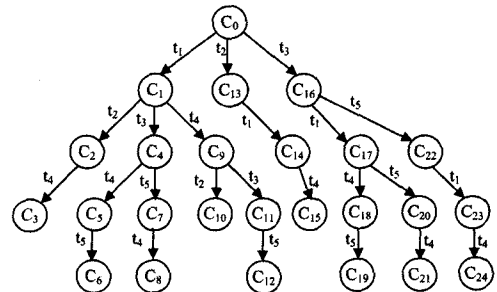


图 2 图 1 中 TPN 的状态类树

3.2 实施区间计算

在时间 Petri 网中, 变迁的实施选择采用时间竞争策略, 即在所有可实施变迁中选择实施时间最小的变迁实施。而变迁实施时间点的设置是在变迁开始使能时根据某种随机分布产生的。在每一次模拟的过程中, 可以记录每个变迁的实施时间点。最后通过对实施点数据的统计分析, 确定该变迁的实施区间范围。

变迁实施会使 Petri 网从一个状态类转移到另一个状态类, 因此状态类 C_i 的时间区间对应于从 C_0 到 C_i 的变迁序列的实施区间。对图 1 中 TPN 进行了 10^4 次至 10^8 次模拟, 记录所有变迁的实施时间点, 并进行数据统计, 所有状态类的实施区间如表 2 所列。

表2 状态类时间区间

状态类	时间区间 (10^4 次)	时间区间 (10^6 次)	时间区间 (10^8 次)	理论值	绝对误差(%) (10^8 次)	相对误差(%) (10^8 次)
C_1	[3.0000, 4.9988]	[3.0000, 5.0000]	[3.0000, 5.0000]	[3,5]	0	0
C_2	[3.0314, 5.8970]	[3.0032, 5.9846]	[3.0008, 5.9990]	[3,6]	0.0009	0.030
C_3	[4.0561, 6.8644]	[4.0029, 6.9935]	[4.0003, 6.9986]	[4,7]	0.0017	0.057
C_4	[3.0272, 5.9956]	[3.0048, 6.0000]	[3.0000, 6.0000]	[3,6]	0	0
C_5	[4.0229, 6.9270]	[4.0037, 6.9899]	[4.0007, 6.9995]	[4,7]	0.0012	0.040
C_6	[4.4678, 8.9346]	[4.1360, 8.9786]	[4.0570, 8.9992]	[4,9]	0.0578	1.156
C_7	[4.2218, 6.6902]	[4.0756, 6.9490]	[4.0058, 6.9932]	[4,7]	0.0126	0.420
C_8	[4.3985, 6.7164]	[4.1552, 6.9910]	[4.0331, 6.9981]	[4,7]	0.0350	1.167
C_9	[4.0070, 5.9710]	[4.0031, 5.9965]	[4.0000, 5.9997]	[4,6]	0.0003	0.015
C_{10}	[4.2830, 5.9414]	[4.0592, 5.9978]	[4.0046, 5.9996]	[4,6]	0.0050	0.250
C_{11}	[4.1360, 5.9992]	[4.0364, 6.0000]	[4.0040, 6.0000]	[4,6]	0.0040	0.200
C_{12}	[5.6124, 8.9740]	[5.1494, 8.9956]	[5.0424, 8.9990]	[5,9]	0.0434	1.085
C_{13}	[2.0006, 4.8530]	[2.0000, 4.9820]	[2.0000, 4.9994]	[2,5]	0.0006	0.020
C_{14}	[3.0040, 5.0000]	[3.0000, 5.0000]	[3.0000, 5.0000]	[3,5]	0	0
C_{15}	[4.0496, 6.9505]	[4.0038, 6.9956]	[4.0006, 6.9998]	[4,7]	0.0008	0.027
C_{16}	[2.0000, 4.9636]	[2.0000, 4.9920]	[2.0000, 4.9996]	[2,5]	0.0004	0.013
C_{17}	[3.0004, 5.0000]	[3.0000, 5.0000]	[3.0000, 5.0000]	[3,5]	0	0
C_{18}	[4.0332, 6.8176]	[4.0037, 6.9840]	[4.0003, 6.9992]	[4,7]	0.0011	0.037
C_{19}	[4.2302, 7.8266]	[4.0626, 7.9726]	[4.0076, 7.9868]	[4,8]	0.0208	0.520
C_{20}	[3.0572, 6.9364]	[3.0242, 6.9364]	[3.0028, 6.9926]	[3,7]	0.0102	0.255
C_{21}	[4.1222, 6.9639]	[4.0083, 6.9989]	[4.0005, 6.9997]	[4,7]	0.0008	0.027
C_{22}	[3.0074, 4.9818]	[3.0018, 4.9942]	[3.0002, 4.9998]	[3,5]	0.0004	0.020
C_{23}	[3.0684, 4.9996]	[3.0266, 5.0000]	[3.0054, 5.0000]	[3,5]	0.0054	0.270
C_{24}	[4.3147, 6.9973]	[4.0626, 6.9992]	[4.0356, 6.9998]	[4,7]	0.0358	1.193

从表2容易看出,随着模拟次数的增加,模拟统计所得的状态类时间区间与理论值越来越接近。当模拟次数达到 10^8 时,统计值与理论值的绝对误差均控制在0.06以下,其中70%的状态类控制到千分位;相对误差均控制在1.2%以下,其中80%的状态类控制在0.6%以下。

有4个末端状态类节点的相对误差高于1.0%,这4个节点分别是 C_6, C_8, C_{12} 和 C_{24} ,其主要原因是它们受到顺序变迁概率累积效应的影响。例如, t_1 和 t_4 是顺序关系,序列 $t_1 t_4$ 取上下界值的概率是两个变迁同时取界值的概率乘积;而 t_1 和 t_2 不是顺序关系,序列 $t_1 t_2$ 或 $t_2 t_1$ 取上下界值的概率可能只是其中一个变迁取界值的概率。

3.3 变迁实施概率计算

在模拟过程中,可以记录变迁实施序列的实施次数,如表3所列。通过变迁实施序列的实施次数,可以求解在某一状态类下,各变迁的实施频率,获得变迁实施概率。

用 $\Pr(C_i, t_k)$ 表示在 C_i 实施 t_k 的概率,用 $Tm(C_i, t_k)$ 表示 t_k 在 C_i 的实施次数。则 $\Pr(C_i, t_k)$ 可由下式计算得到。

$$\Pr(C_i, t_k) = \frac{Tm(C_i, t_k)}{\sum_{t_j \in Fr(C_i)} Tm(C_i, t_j)}$$

例如,在状态类 C_0 ,变迁 t_1, t_2, t_3 均可实施, t_1 的实施概率为:

$$\Pr(C_0, t_1) = (77095 + 182183 + 13034 + 12953 + 61254) / 10^8 = 0.34719373$$

在状态类 C_4 ,变迁 t_5 的实施概率为:

$$\Pr(C_4, t_5) = 1302976 / (18228625 + 1302976) = 0.066711172$$

表3 变迁实施序列的实施次数

变迁实施序列	实施次数
$t_1 t_2 t_4$	7723443
$t_1 t_3 t_4 t_5$	18228625
$t_1 t_3 t_5 t_4$	1302976
$t_1 t_4 t_2$	1302531
$t_1 t_4 t_3 t_5$	6161798
$t_2 t_1 t_4$	24310307
$t_3 t_1 t_4 t_5$	10752995
$t_3 t_1 t_5 t_4$	22574931
$t_3 t_5 t_1 t_4$	7642394

将第3.2节和3.3节的计算结果附加到状态类树,得到如图3所示的带时间区间和概率信息的状态类树。通过构造扩展状态类树,可以很方便地获取任意变迁序列的实施区间和实施概率。

例如,对于变迁序列 $\sigma = t_3 t_1 t_5 t_4$,从图3可达树中可以获得序列 σ 的实施区间为 $[4, 7]$ 。该序列的实施概率可以通过序列路径上的各条边的实施概率的乘积来计算,即:

$$\Pr(\sigma) = 0.41 * 0.81 * 0.68 * 1 = 0.23$$

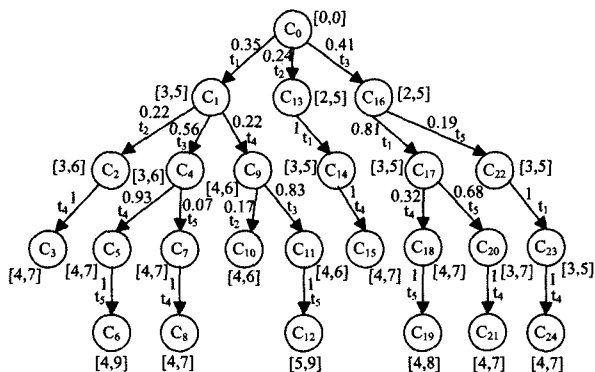


图3 带时间区间和概率信息的状态类树

结束语 本文的主要贡献包括:1)提出时间 Petri 网的随机模拟方法,通过某种随机分布生成实施区间内的实施时间点,为时间 Petri 网的模拟实现提供了可行的方案;2)提出基于随机模拟数据的统计分析方法,给出了状态类树、变迁实施区间及概率的构造算法和计算方法,为时间 Petri 网的系统模拟提供了有效的分析方法。

本文提出的随机模拟实验和统计分析方法同样适用实施区间为其它概率分布的情况,但模拟实验数据特征、统计分析结果很可能发生变化。下阶段工作将研究其他概率分布下(例如三角形分布、截断正态分布等)的时间 Petri 网随机模拟实验及统计分析。

参考文献

- [1] Merlin P, Farber D J. Recoverability of communication protocols: Implication of a theoretical study[J]. IEEE Trans. Commun., 1976, 24(9): 1036-1043

(下转第159页)

对本协议的吞吐量和延迟进行了仿真,并对性能进行了验证,从仿真结果可以看出本协议与 CAMAC 协议相比,吞吐量增加了 10%,延迟减少了 60ms,提高了网络的吞吐量,减少了节点间传输数据的延迟。

参 考 文 献

- [1] Oyman O, Singh S. Quality of experience for HTTP adaptive streaming services[J]. Communications Magazine, 2012, 50(4): 20-27
 - [2] Du Wei. Multi-Channel MAC Protocol in Ad Hoc Research[D]. Harbin: Harbin Institute of Technology, 2009(in Chinese)
杜威. Ad Hoc 网络多信道 MAC 协议研究[D]. 哈尔滨: 哈尔滨工业大学, 2009
 - [3] Maamar H, Boukerche A, Petriu E. Streaming 3d meshes over thin mobile devices[J]. Wireless Communications, 2013, 20(3): 136-142
 - [4] Qureshi F F, Baksh A G. Reliable Cognitive Radio Multi-Channel MAC Protocol for Ad Hoc Networks[C]//ICCSA. 2015
 - [5] Wen Jing-rong. Research on The MAC Layer And Related Technology of Wireless Ad Hoc Networks[D]. Beijing: Beijing University of Posts and Telecommunications, 2013(in Chinese)
温景容. 无线自组网 MAC 层及相关技术研究[D]. 北京: 北京邮电大学, 2013
 - [6] Bai Ke. Design and Simulation of A Novel Ad Hoc Multi-Channel Access Protocol[D]. Chengdu: University of Electronic Science and Technology of China, 2008(in Chinese)
白科. 一种新的 Ad Hoc 多信道接入协议的设计与仿真[D]. 成都: 电子科技大学, 2008
 - [7] Rezende C, Mammeri A, Boukerche A, et al. Areceiver-based video dissemination solution for vehicular networks with content transmissions decoupled from relay node selection[J]. Ad Hoc Networks, 2014, 17(6): 1-17
 - [8] Vella J, Zammit S. A survey of multicasting over wireless access networks[J]. Communications Surveys Tutorials, IEEE, 2012, 15(2): 1817-1827
 - [9] Run Xi. Research on QoS-Guaranteed MAC Protocol in Ad Hoc Network[J]. Modern Navigation, 2013(3): 212-215(in Chinese)
闰哲. 确保无线 AdHoc 网络服务质量的 MAC 协议研究[J]. 现代导航, 2013(3): 212-215
 - [10] Wang Fu-cai. Protocol Design and Research in Multi-Channel Wireless Ad Hoc Networks[D]. Changsha: National University of Defense Technology, 2011(in Chinese)
王付彩. 多信道无线 Ad hoc 网络协议设计与研究[D]. 长沙: 国防科学技术大学, 2011
 - [11] Li Jia. Multi-Channel MAC Protocol Research and Performance Analysis for Mobile Ad Hoc Network[D]. Beijing: Beijing University of Posts and Telecommunications, 2013(in Chinese)
李佳. 移动 Ad Hoc 网络多信道 MAC 协议研究及性能分析[D]. 北京: 北京邮电大学, 2013
 - [12] 柯志恒. NS2 仿真实验[M]. 北京: 电子工业出版社, 2009
-
- (上接第 129 页)
- [2] Berthomieu B, Diaz M. Modeling and Verification of Time Dependent Systems Using Time Petri Nets [J]. IEEE Trans. Softw. Eng., 1991, 17(3): 259-273
 - [3] Wang J, Xu G, Deng Y. Reachability analysis of real-time systems using time Petri nets[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 2000, 30(5): 725-736
 - [4] Pan Li, Ding Zhi-jun, Zhou Meng-chu. A Configurable State Class Method for Temporal Analysis of Time Petri Nets[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2014, 44(4): 482-493
 - [5] Boucheneb H, Gardey G, Roux O H. TCTL model checking of time Petri nets[J]. Journal of Logic and Computation, 2009, 19(6): 1509-1540
 - [6] Berard B, Cassez F, Haddad S, et al. The expressive power of time Petri nets[J]. Theoretical Computer Science, 2013, 474(N/A): 1-20
 - [7] Boel R K, Jiroveanu G. The on-line diagnosis of time Petri nets [M]. Control of Discrete-Event Systems. Springer London, 2013: 343-364
 - [8] Wu N Q, Chu F, Chu C, et al. Net Modeling and Cycle-Time Analysis of Dual-Arm Cluster Tools With Wafer Revisiting[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2013, 43(1): 196-207
 - [9] Du Y, Tan W, Zhou M C. Timed compatibility analysis of web service composition; A modular approach based on Petri nets [J]. IEEE Transactions on Automation Science and Engineering, 2014, 11(2): 594-606
 - [10] Gardey G, Lime D, Magnin M, Romeo A. A tool for analyzing time Petri nets[C]//Proceedings of 17th International Conference on Computer Aided Verification. LNCS 3576, Springer Berlin Heidelberg, 2005: 418-423
 - [11] Berthomieu B, Ribet P, Vernadat F. The tool TINA-Construction of abstract state spaces for petri nets and time petri nets [J]. International Journal of Production Research, 2007, 42(14): 2741-2756
 - [12] Lin Chuang, Qu Yang, Li Ya-juan. Modeling, Consistency and Inference of Extended Interval Temporal Logic [J]. Chinese Journal of Computers, 2002, 25(12): 1338-1347(in Chinese)
林闯, 曲扬, 李雅娟. 扩展时段时序逻辑的模型、一致性和推理 [J]. 计算机学报, 2002, 25(12): 1338-1347
 - [13] Lei D M. Interval job shop scheduling problems[J]. The International Journal of Advanced Manufacturing Technology, 2012, 60(1-4): 291-301
 - [14] Sotskov Y N, Lai T C, Werner F. Measures of problem uncertainty for scheduling with interval processing times [J]. OR Spectrum, 2013, 35(3): 659-689
 - [15] Lei D M, Guo X P. An effective neighborhood search for scheduling in dual-resource constrained interval job shop with environmental objective [J]. International Journal of Production Economics, 2015, 159(1): 296-303
 - [16] Pan Li, Ding Zhi-jun, Chen Gang. A Time Petri Net Model Based on Uniform Distribution [J]. Chinese Journal of Computers, 2010, 33(5): 900-907(in Chinese)
潘理, 丁志军, 陈刚. 基于一致分布的时间 Petri 网模型 [J]. 计算机学报, 2010, 33(5): 900-907
 - [17] Donald R B, Sherrill E T. Mean and variance of truncated normal distributions[J]. The American Statistician, 1999, 53(4): 357-361
 - [18] Jawitz J W. Moments of truncated continuous univariate distributions[J]. Advances in Water Resources, 2004, 27(3): 269-281