一种基于诱导规则的 Petri 网完备日志生成算法

靳伟国[1]，闻立杰[2]，王建民[2]，武年华[2]
(1. 北方工业大学 信息中心，北京 100144；2. 清华大学 软件学院，北京 100084)

摘　要： 过程挖掘旨在从事件日志中自动抽取过程模型用于支持过程设计和分析，完备日志是过程挖掘算法设计、测试和分析的前提。提出了基于 Petri 网模型行为仿真的完备日志生成算法，在仿真过程中加入基于发生次数及相继关系的诱导规则，即在有多个任务使能的情况下选择触发发生次数少且覆盖新相继关系的任务，使其在尽可能少的实例下产生完备的日志。诱导规则的应用减少了模拟模型发生的各种情况所需的时间，同时也提高了所有情况发生的概率，使得日志的完备性成为可能。

关键词： 完备日志；过程挖掘；诱导规则；Petri 网；行为仿真
中图分类号： TP312；TP301.6　　**文献标志码：** A　　**文章编号：** 1001-3695(2016)07-2051-05
doi:10.3969/j.issn.1001-3695.2016.07.029

Induced rule based algorithm for complete log generation of Petri nets

Jin Weiguo[1], Wen Lijie[2], Wang Jianmin[2], Wu Nianhua[2]
(1. *Information Center, North China University of Technology, Beijing* 100144, *China*; 2. *School of Software, Tsinghua University, Beijing* 100084, *China*)

Abstract: Process mining aims at automatically deriving process models from event logs to support the process design and analysis. The complete log is the basis of process mining algorithms design, testing and analysis. This paper proposed an algorithm based on the behavioral simulation of Petri net, and employed the induced rules to make choices when two or more tasks were enabled in the process of simulation based on the times of task occurrences and the succession times of two consecutive tasks. It generated as few instances as possible to get a complete log. As a result, it reduces the time for generating all cases, and makes the complete log become feasible.

Key words: complete log; process mining; induced rule; Petri nets; behavioral simulation

0　引言

近十几年工作流技术得到了广泛应用，除了独立的工作流管理系统，工作流技术也被各个企业吸纳和采用，如 ERP、CRM、SCM 和 B2B 等都被用来较好地控制业务流程的变化[1]。工作流管理系统，如 Staffware、IBM MQSeries 和 COSA 等都可以有效支持业务过程的建模、分析、仿真和优化过程实例的执行、监视和控制。然而构造符合业务过程需求的工作流模型是一件复杂、耗时、易错的事情，且需要深厚的领域知识支持。这时就需要利用工作流管理系统日志中包含的各个活动执行信息，运用一定挖掘技术进行过程挖掘来重构业务过程模型。过程挖掘旨在利用事件日志来分析业务活动之间的关系，自动构造确切的过程模型[2,3]。

事件日志是过程挖掘所必需的资源，是过程挖掘的前提，日志数据的质量在很大程度上决定了模型挖掘结果的好坏；另一方面，挖掘算法研究者也急需日志来测试和分析各类挖掘算法，完备日志是绝大多数过程挖掘算法分析和评估的前提条件[4]。直接利用现实生活中的日志看起来是一个不错的选择，然而现实生活中的日志可能源于一个非模型驱动的信息系统，也可能来源于某个未知的过程模型，信息杂乱且格式不规范，更重要的是现实生活中的日志可能不完备且有噪声，而且属于企业保护资产，极难获取。由此看来，通过对模型进行行为仿真获得日志是可行的，这种方法不仅允许研究者对模型日志的特征进行控制，而且原始模型的存在进一步帮助了研究者对挖掘算法的分析。事件日志的产生问题可以归结为求过程模型合法任务执行序列的集合，完备日志即指过程模型所有可能的任务执行序列的集合。然而一个过程模型的执行序列可能是无穷的(如存在循环结构)或者是数量大到难以枚举(如较多并行分支结构)，如果采用单纯的行为仿真方法，即存在多个任务使能的情况下采用完全随机的方法进行选择，会使得状态空间爆炸问题更加突出，很难得到完备日志，同时量化评价日志的完备性也变得几乎不可能。因此，如何有效地控制行为仿真过程，使其在尽可能少的实例下产生尽可能完备的日志是一个研究难题。

本文主要研究工作流网(WF-net)完备日志生成算法。WF-net 是 Petri 网的子集。Petri 网以其表达能力、形式化为基础，已有众多研究成果和现有工具的支持，成为表达和分析过程模型的理想语言[5]，而且现有很多算法实现了其他的过程语言模型到 Petri 网模型的转换[6,7]。因此，本文研究基于工作流网生成完备日志的算法具有代表性和普遍意义。

收稿日期：2015-03-02；修回日期：2015-05-20

作者简介：靳伟国(1972-)，男，河北唐山人，实验师，硕士，主要研究方向为网络技术、数据挖掘(jin197612@126.com)；闻立杰(1977-)，男，河北唐山人，副教授，博士，主要研究方向为工作流技术、流程数据管理与挖掘；王建民(1968-)，男，吉林磐石人，副院长，教授，博士，主要研究方向为大数据管理、流程数据管理与挖掘；武年华(1984-)，女，湖南衡阳人，硕士，主要研究方向为流程挖掘.

本文采用的基本方法是对工作流网进行行为仿真，得到模型在实际运行中可能出现的任务执行序列。在仿真过程中，除完全顺序结构模型外都会存在多个任务都可能发生的情况，这时就必须选择其中一个先执行。为此，在仿真过程中加入基于各个任务发生次数这一诱导规则，优先选择具有最少发生次数的任务执行，保证了选择的公平性，并大大提高了各种执行序列的发生概率。鉴于在挖掘算法中日志并不需要全部发生序列均有所体现，而只需要足以反映其结构的那些行为，本文对日志完备性作出了合理的假设，即日志次序关系完备。

本文简要介绍了用到的基础知识，主要包括 Petri 网、WF-net 定义，以及事件日志相关形式化定义，详细介绍了基于诱导规则的完备日志产生算法，包括诱导规则定义、诱导规则实施和案例分析，并给出算法的实验评估情况。

1 相关工作

事件日志是过程挖掘必需的资源，过程挖掘算法研究者要使用事件日志对算法进行验证与分析。现有事件日志算法主要分为两种，一种是基于模型行为，另一种是基于模型结构。基于模型行为的算法主要通过对模型行为进行仿真来获取实例运行时的信息；而基于模型结构的算法首先对模型结构进行分解，通过将基础结构的日志进行有机组合来获得最终事件日志。Westergaard 和 Günther 等人提出利用 CPN tools [8] 和 ProMimport [9] 两个工具生成用于测试挖掘算法的基于着色 Petri 网(CP-nets)的事件日志的方法 [10]。CPN tools 主要用来对 CP-nets 进行行为仿真，作者通过对 CPN tools 进行扩展，加入了一系列可用于记录单个事件轨迹的 ML 函数，最后利用 ProMimport 中的相关插件把所有实例信息按照相关格式组织起来，生成一个事件日志文件。该方法不能保证日志的完备性，只能通过生成尽可能多的实例来实现一定程度上的完备。查海平等人 [11] 提出了基于结构分解的满足日志次序关系的完备日志生成算法。算法分析了基础结构的二元日志次序关系集，通过对结构化简的方法，对模型采用基本结构进行最大化替代，而对复杂结构采用最小化替代的方法逐步递归细化，得到各个结构块的二元日志次序关系集，然后在此基础上构造日志。该算法有效避免了基于行为分析的完备日志生成算法存在的状态空间爆炸问题，但其生成的日志可能不是模型实际的触发序列，且对于复杂的结构不能有效构造出日志。

2 基础知识

本章介绍文中用到的基础知识，主要包括 Petri 网、WF-net 的定义以及事件日志相关形式化定义。如无特殊说明，这些概念均直接引自文献[12]。

2.1 Petri 网

全文使用传统 Petri 网模型的一类变种，即所谓的库所(place)/变迁(transition)网，简称 P/T 网。有关 Petri 网更详尽的阐述，请参见文献[13,14]。

定义 1 P/T 网。库所/变迁网是一个三元组 $N=(P,T,F)$，其中：P 是库所的有穷集合；T 是变迁的有穷集合，满足 $P\cap T=\varnothing$ 且 $P\cup T\neq\varnothing$；$F\subseteq(P\times T)\cup(T\times P)$，是有向弧的集合，称做流关系。

定义 2 标志 P/T 网。它是一个二元组 (N,s)，其中：N 是一个 P/T 网；s 是 P 到自然数的一个函数，表示网的标志，即 $s:P\to\{0,1,2,\cdots\}$。所有标志 P/T 网的集合记做 Ω。

定义 3 节点、前集、后集。令 $N=(P,T,F)$ 是一个 P/T 网，则有：

a) $P\cup T$ 中的元素被称为节点；

b) 若 $x,y\in P\cup T$ 且 $(x,y)\in F$，则 x 是 y 的输入节点；

c) 若 $x,y\in P\cup T$ 且 $(y,x)\in F$，则 x 是 y 的输出节点；

d) 对任意 $x\in P\cup T$，x 的输入集或前集为 $\cdot x=\{y\mid(y,x)\in F\}$；

e) 对任意 $x\in P\cup T$，x 的输出集或后集为 $x\cdot=\{y\mid(x,y)\in F\}$。

图 1 显示了一个由八个库所和七个变迁组成的 P/T 网。变迁 A 有一个输入库所和一个输出库所，变迁 AND-split 有一个输入库所和两个输出库所。变迁 A 的输入库所中的黑点代表托肯(即 token)，变迁 A 的输入库所中的托肯表示网的初始标志。

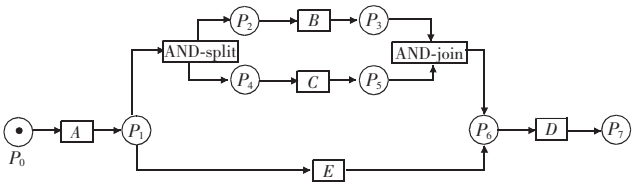

图 1 由八个库所和七个变迁组成的 P/T 网

定义 4 发生规则。令 $(N=(P,T,F),s)$ 是一个标志 P/T 网，则有：变迁 $t\in T$ 可发生，表示为 $(N,s)[t\rangle$，当且仅当 $\cdot t\leqslant s$；发生规则 $[\rangle\subseteq\Omega\times T\times\Omega$ 是满足如下条件的最小关系：

$$\forall(N,s)\in\Omega,t\in T:(N,s)[t\rangle\Rightarrow(N,s)[t\rangle(N,s-\cdot t+t\cdot)$$

在图 1 所示的网标志(即源库所中有一个托肯)下，变迁 A 可发生。变迁 A 发生后，从输入库所移除一个托肯，并在输出库所放入一个托肯。在结果标志下，两个变迁是可发生的：E 和 AND-split。尽管两者均可发生，但最终只能有一个发生，因为两者抢夺公共输入库所 P_1 中的唯一托肯。如果 AND-split 发生，则消耗一个托肯，并产生两个新的托肯。

定义 5 可达标志。令 (N,s_0) 为 Ω 中的一个标志 P/T 网，标志 s 是初始标志 s_0 的可达标志，当且仅当存在一系列可发生的变迁，这些变迁的先后发生使得标志从 s_0 变化到 s。(N,s_0) 的可达标志集合记为 $[N,s_0\rangle$。

2.2 工作流网

建模过程控制流维度的 Petri 网被称做工作流网，即 WF-net，它定义了同一类案例共同的动态行为。工作流网是广泛接受的描述过程控制流维度的 Petri 网，由著名工作流技术专家 Aalst 等人 [15] 提出，它具有良好的表达能力和分析特性。

定义 6 工作流网。令 $N=(P,T,F)$ 为 P/T 网，t 为不属于 $P\cup T$ 的新节点，N 是工作流网(简称 WF-net)，当且仅当：

a) 对象创建。P 包含这样一个输入库所 i，满足 $\cdot i=\varnothing$。

b) 对象完成。P 包含这样一个输出库所 o，满足 $o\cdot=\varnothing$。

c) 连通性。$\bar{N}=(P,T\cup\{t\},F\cup\{(o,t),(t,i)\})$ 是强连通的。

图 1 所示的 P/T 网是一个 WF-net。尽管原始 P/T 网不是强连通的，但其短回路网 $\bar{N}=(P,T\cup\{t\},F\cup\{(o,t),(t,i)\})$

(即用新变迁 t 连接 o 和 i) 是强连通的。即使一个 P/T 网满足定义 6 中描述的所有性质,其执行过程仍然可能出错,如死锁(再没有任务能够发生) 或活锁(过程结束后有托肯残留) 等。

定义 7　合理性。令 $N=(P,T,F)$ 为 WF-net,输入库所为 i,输出库所为 o,则 N 是合理的,当且仅当:

a) 安全性。$(N,[i])$ 是安全的,即所有可达标志的库所中最多只包含一个托肯。

b) 恰当完成。$\forall s\in[N,[i]\rangle: o\in s\Rightarrow s=[o]$。

c) 可完成。$\forall s\in[N,[i]\rangle: o\in[N,s\rangle$。

d) 无死任务。$[N,[i]\rangle$不包含任何死任务。

图 1 中的 WF-net 是合理的。合理性工作流网是从行为上来定义的 WF-net 子集,大多数的业务过程模型满足合理性的要求。合理性对于日志产生算法具有重要的意义。

2.3　事件日志

过程挖掘的目标就是在对事件日志进行分析的基础上构造过程模型。事件轨迹、事件日志以及日志次序关系的形式化定义如下。

定义 8　事件轨迹、事件日志。令 T 为任务集合,$\sigma\in T^*$ 是一条事件轨迹,$W\subseteq T^*$ 为一个事件日志。

定义 9　相继。W 是任务集 T 上的事件日志,即 $W\subseteq T^*$,令 $a,b\in T$,则 b 在 W 中紧随 a 发生,记做 $a>_W b$,当且仅当 $\exists\sigma=t_1t_2\cdots t_n\wedge 1\leqslant i\leqslant n-1$,使得 $\sigma\in W\wedge t_i=a\wedge t_{i+1}=b$。

两个任务 a、b 相继的概念就是说在某个事件轨迹当中,这两个任务至少有机会相继出现,而且 a 与 b 之间不存在其他任务发生。相继是两个任务间最基本的日志次序关系,在此基础上可推导出以下关系:

a) $a\rightarrow_W b$,当且仅当 $a>_W b\wedge b\not>_W a$;

b) $a\parallel_W b$,当且仅当 $a>_W b\wedge b>_W a$;

c) $a\#_W b$,当且仅当 $a\not>_W b\wedge b\not>_W a$。

令事件日志 $W=\{AED,ABCD,ACBD\}$,"$>_W$"关系描述了事件日志中任务之间基础的紧邻关系,日志 W 中存在的 $>$ 关系有 $A>_W E$、$E>_W D$、$A>_W B$、$B>_W C$、$C>_W D$、$A>_W C$、$C>_W B$、$B>_W D$,"$\rightarrow_W$"则表示 a、b 之间的因果关系,"$\parallel_W$"描述了任务之间的并行关系,"$\#_W$"表示两个任务之间不存在紧邻,这样也就不存在因果或是并行关系了。鉴于其他三种关系都是由"$>_W$"关系推导出来的,在这里引入 α 算法[8] 中基于相继关系的完备性定义。

定义 10　完备工作流日志。设 $N=(P,T,F)$ 是一个合理的 WF-net,W 是 N 的完备日志,当且仅当:

a) $W'\subseteq T^*$,满足 $W'\subseteq W$;

b) $\forall t\in T,\exists\sigma\in W$,使得 $t\in\sigma$。

为有效挖掘短循环结构与非自由选择结构 α^+ 算法[16]、α^{++} 算法[17],分别对该定义进行了相应的扩充。

3　基于诱导规则的完备日志生成算法

本章介绍基于诱导规则的完备日志生成算法,主要包括工作流网相继关系的获取、诱导规则的定义与实施,以及如何利用诱导规则和相继关系生成完备日志。

3.1　获取工作流网的相继关系

要产生满足相继关系的完备日志,首先必须得到模型任务所有可能的相继关系。最直接的方法是遍历模型所有的可能路径,然后从路径中提取相继关系。但该算法的时间开销会随着模型的规模呈指数增长。为得到所有可能相继关系而又避免状态空间爆炸,本文直接利用工作流网的可达图[18] 的求解过程来获得相继关系集合。假设工作流网每个库所最多只能容纳一个托肯,这样工作流网就是安全的,其对应的可达标志集是一个有限集。以可达标志集作为顶点集,以标志之间的直接可达关系为边集构成一个有向图,即工作流网的可达图,记为 RG(N)。可达图记录了工作网的状态变化和变迁发生序列的情况,进而得知工作流网中任务之间的相继关系。

图 2 为图 1 中工作流网的可达图。文献[18] 详细描述了 Petri 网可达图定义及构造方法。图 2 中每条弧上都有一个任务作为旁标,如果两条有向边通过一个标志直接相连,则弧上对应的任务间存在相继关系。记工作流网 N 的相继关系集为 C_N。

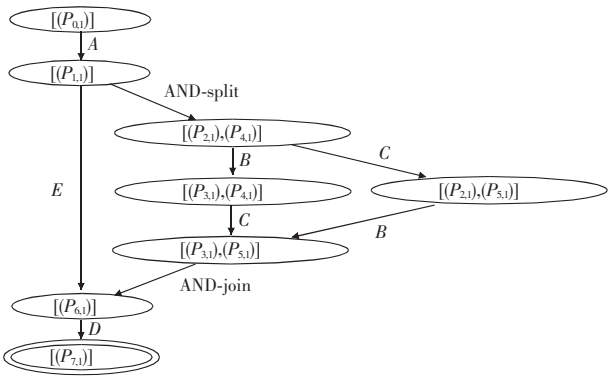

图 2　图 1 中工作流网的可达图

从图 2 所示的可达图可以看出,该模型从 P_0 中包含一个托肯的初始标志,沿着有向弧的方向不断前行,直到抵达 P_7 中包含一个托肯的终止标志,其任务之间的全部相继关系集合为 $\{A>E,E>D,A>\text{AND-split},\text{AND-split}>B,\text{AND-split}>C,B>C,C>B,B>\text{AND-join},C>\text{AND-join},\text{AND-join}>D\}$。

3.2　基于发生次数的诱导规则

在介绍诱导规则之前,首先给出发生次数的形式化定义。

定义 11　发生次数。令 $N=(P,T,F)$ 为 WF-net,发生次数函数 h 是 N 的任务集 T 到自然数的一个函数,表示在行为仿真过程中存在多个任务使能情况下任务被选择发生的次数,即 $h: T\rightarrow\{0,1,2,\cdots\}$。

在对过程模型进行仿真之前,需要对发生次数进行初始化,即 $\forall a\in T,h(a)=0$。在对过程模型进行仿真时,如果有多个任务使能,将采用如下诱导规定选择任务发生。

定义 12　诱导规则。令 $(N=(P,T,F),s_0)$ 为标志 WF-net,h 为任务集 T 上的发生次数函数,C_N 为 N 的相继关系集合,C_S 为事件日志中已有实例覆盖的相继关系的集合,a 为当前实例最近发生的任务,s 为当前标志,诱导规则定义如下:

a) 获得集合 $T_C=\{x\in T\mid\cdot x\leqslant s\}$;

b) 获得集合 $Tr=\{x\in T_C\mid a>_W x\notin C_S\}$;

c) 选择发生任务 t:

若 $Tr=\varnothing$,则有 $\forall x\in T_C,h(x)\geqslant h(t)$;

若 $Tr\neq\varnothing$,则有 $\forall x\in Tr,h(x)\geqslant h(t)$。

根据诱导规则,仿真时总是优先选择触发发生次数较少且覆盖新的相继关系的任务,保证了日志在模型各个分支上的相

对公平性，优先产生不同的事件轨迹，能减少产生完备日志所需要的运行实例个数，同时减少生成日志的时间。

3.3 算法实现

算法的核心思想就是在仿真过程中，在出现多个任务使能的情况下，根据诱导规则选择触发发生次数最少且覆盖新的相继关系的任务。发生次数少与覆盖新的相继关系这两个条件是相辅相成的，因为发生次数少就在一定程度上蕴涵着其相对应的任务相关的相继关系还未得到覆盖。若所有的相继关系都已被覆盖，则直接选择触发发生次数最少的任务。

对一个工作流网 $N=(P,T,F)$，其可达图记为 $RG(N)$，其所有相继关系集合记为 C_N，h 为任务集 T 上的发生次数函数。C_S 为事件日志中已生成的实例覆盖的相继关系的集合；a 为当前实例最近发生的任务；T_C 为存储了当前使能任务的优先队列，队列中的元素按其发生次数由小到大进行排序；caseNumber 表示当前为第多少个实例。算法详细过程如下：

算法 1 基于诱导规则的完备日志生成算法

输入：工作流网 $N=(P,T,F)$、日志中最少实例数 minNumber。

输出：事件日志文件的路径以及 caseNumber，即实际事件日志中实例个数，算法结束。

a) 获取相继关系集合。根据已有算法构造模型的可达图 $RG(N)$，在 $RG(N)$ 构造过程中直接抽取边与边之间的信息，获得所有的任务相继关系 C_N。

b) 初始化。对发生次数函数 h 进行初始化，$\forall a \in T$，$h(a)=0$，即在开始仿真之前，所有任务的发生次数均为 0；令 caseNumber = 0，$C_S=\varnothing$。

c) 仿真。清空 N 的库所 P 中的所有托肯，caseNumber 增加 1，在源库所 i 中放入一个托肯，开始该实例的仿真，根据定义 4 的发生规则以及发生次数函数计算出使能的任务优先队列 T_C。

Case 1 T_C 为空，表示当前没有任务使能，该实例结束。

(a) caseNumber < minNumber 或 $C_S \subset C_N$，即日志相继完备性未满足或还未达到用户指定的最小实例数，重新执行 c)；

(b) caseNumber ≥ minNumber 且 $C_S=C_N$，即日志相继完备性已满足且达到用户指定的最小实例数，执行 d)。

Case 2 T_C 只含有一个元素，则直接触发该元素。

Case 3 T_C 中含有多个元素，则：

(a) $C_S=C_N$，直接选择 T_C 中第一个元素进行触发；

(b) $C_S \subset C_N$，按顺序遍历 T_C 中的元素，查找第一个覆盖新的相继关系的任务。如果找不到，则直接触发 T_C 的第一个任务；否则触发该任务。

不妨设触发的任务为 x，更新 x 的发生次数，即 $h(x)$ 增加 1，更新 C_S。如果有新的相继关系被覆盖，将其加入 C_S。重新计算使能任务队列 T_C，继续执行 c)。

d) 形成事件日志文件。将所有实例的执行信息按照 MXML 格式[14] 写入文件。

3.4 案例分析

通过一个案例来进一步了解算法的过程。如图 3 所示，工作流网模型 N 包含选择、并行、循环等结构，令用户输入的最小实例数 minCase = 5，模型 N 的相继关系集为 $C_N=\{A>_W B, A>_W C, B>_W D, B>_W E, D>_W E, E>_W D, D>_W F, E>_W F, F>_W I, C>_W G, G>_W H, G>_W J, J>_W G, H>_W I\}$。仿真的具体情况如表 1 所示。其中 T_C 表示使能任务的优先队列，h 表示发生次数的变化情况，x 表示选择触发的任务。

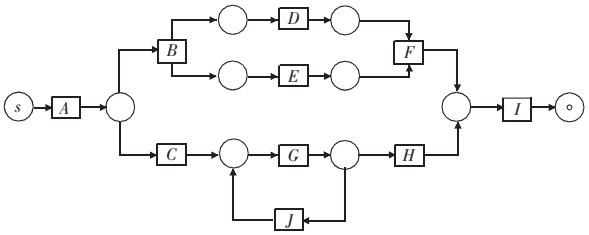

图 3 包含选择、并行、循环结构的工作流网

表 1 五个实例的具体任务触发情况

	第一步	第二步	第三步	第四步	第五步	第六步	第七步
caseNumber = 1							
T_C	{A}	{B,C}	{D,E}	{E}	{F}	{I}	
x	A	B	D	E	F	I	
h	无	h(B) = 1	h(D) = 1	无	无	无	
caseNumber = 2							
T_C	{A}	{C,B}	{G}	{H,J}	{I}		
x	A	C	G	H	I		
h	无	h(C) = 1	无	h(H) = 1	无		
caseNumber = 3							
T_C	{A}	{B,C}	{E,D}	{D}	{F}	{I}	
x	A	B	E	D	F	I	
h	无	h(B) = 2	h(E) = 1	无	无	无	
caseNumber = 4							
T_C	{A}	{C,B}	{G}	{J,H}	{G}	{H,J}	{I}
x	A	C	G	J	G	H	I
h	无	h(C) = 2	无	h(J) = 1	无	h(H) = 2	无
caseNumber = 5							
T_C	{A}	{B,C}	{D,E}	{E}	{F}	{I}	
x	A	B	D	E	F	I	
h	无	h(B) = 3	h(D) = 2	无	无	无	

在第一个实例(caseNumber = 1)的第二步 T_C 中有 B、C 两个任务，两个任务的发生次数相同且都覆盖新的相继关系，则选择第一个元素 B 进行触发，同理在第三步选择 D 进行触发。而在第二个实例的第二步中，因为 $h(C)<h(B)$，且 C 任务的发生将覆盖新的相继关系，所以选择任务 C 进行触发。第四个实例结束后有 $C_S=C_N$，实例已覆盖所有任务之间的相继关系，所以在第五个实例中直接选择触发发生次数最少的任务。

从该案例中可以看到，在仿真过程中诱导规则的运用使得选择、并行、循环等分支结构上的任务都可以公平有效地发生，从概率上来说缩短了发生各种情况所需要的时间，减少了覆盖全部相继关系的实例，使得算法可在尽可能少的实例下得到尽可能完备的日志。虽然说算法在最坏情况下的时间复杂度达到 $O((n-2)!)$，但在一般情况下，算法所用时间远远小于该值，具有较高的效率。

4 实验与分析

本章将通过实验对算法进行评估与分析。实验机器配置如下：CPU 为 Intel 酷睿双核 2.53 GB；内存为 2.0 GB；操作系统为 Windows XP Professional。实验程序用 Java 编写，Java 虚拟机堆内存为 1.0 GB。

实验所用的数据是根据文献[19]实现的过程模型生成器算法产生，其产生的模型满足合理性和安全性的要求。实验共采用了五组数据，每组数据为 100 个模型，五组数据的特性如

表 2 所示。

表 2　实验用模型数据集特征

数据集	平均任务数	平均选择分支数	平均并行分支数
1	12	3	2
2	20	4	4
3	29	6	6
4	41	6	8
5	54	8	10

接下来分别阐述基于上述五个数据集的三组实验。

a) 测试算法在具有不同分支数的五组模型上,产生完备日志平均所用时间(为降低空间复杂度,一个任务触发后,立即将相关信息写入了日志文件,故这里的时间包括写入日志文件所用时间),结果如图 4 所示。从图上可以清晰地观察到算法所用时间随着模型分支数的增加而增加,然而并行分支数量的增多并未使算法时间呈阶乘式增长,证实了诱导规则运用在一定程序上解决了行为空间的爆炸问题。

b) 比较已有算法与本算法产生完备日志所用时间。因为利用 CPN tools 产生日志的算法不能全自动地进行,所以在这里主要与相关工作中基于结构的算法进行比较,结果如图 5 所示。基于结构的算法在小规模数据上执行效率优于本算法,但在大规模数据上因为结构复杂,使其产生许多并非模型实际触发序列的日志,其执行效率低于本算法。

c) 从每组数据中抽取两个模型,共 10 个模型。比较对于同一个模型,本文中提到的三种算法生成完备日志所需的实例个数,结果如图 6 所示。图 6 中数据显示,本算法在绝大多数情况下,能在比其他两种算法少的实例下产生完备日志,日志完备性更容易得到满足,质量得到提高。

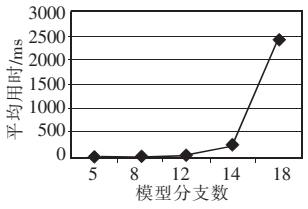

图 4　算法在五个数据集上平均用时

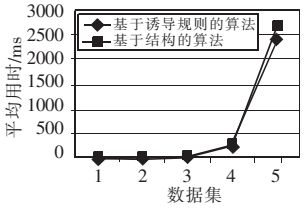

图 5　两种算法产生完备日志平均用时

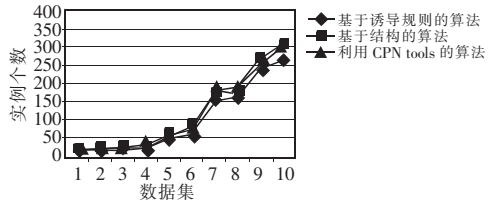

图 6　三种算法产生完备日志所需实例数

5　结束语

完备日志是过程挖掘算法设计、测试、分析的前提。本文提出了一个基于模型行为仿真且满足任务相继关系完备性的日志生成算法。通过对模型进行仿真,并且在仿真过程应用基于发生次数和任务相继关系的诱导规则,使得在各个分支上的任务都能公平有效地触发。与现在方法相比,算法可在较短时间内、较少的实例下生成模型,尽可能全面地执行轨迹。

各种挖掘算法对日志完备性要求不甚相同,目前算法还无法满足 Alpha +、Alpha ++ 算法对日志完备性的要求,这将在以后的工作中加以解决。

参考文献:

[1] Stohr E A, Zhao J L. Workflow automation: overview and research issues [J]. Information Systems Frontiers, 2001, 3(3): 281-296.

[2] Van Der Aalst W M P, Van Dongen B F, Herbst J, *et al*. Workflow mining: a survey of issues and approaches [J]. Data and Knowledge Engineering, 2003, 47(2): 237-267.

[3] De Leoni M, Maggi F M, Van Der Aalst W M P. An alignment-based framework to check the conformance of declarative process models and to preprocess event-log data [J]. Information Systems, 2015, 47(1): 258-277.

[4] Yang Hedong, Wen Lijie, Wang Jianmin, *et al*. CPL +: an improved approach for evaluating the local completeness of event logs [J]. Information Processing Letters, 2014, 114(11): 607-610.

[5] Van Der Aalst W M P. The application of Petri nets to workflow management [J]. Journal of Circuits, System and Computer, 1998, 8(1): 21-26.

[6] Hinz S, Schmidt K, Stahl C. Transforming BPEL to Petri nets [C] //Proc of the 3rd International Conference on Business Process Management. Berlin: Springer, 2005: 220-235.

[7] Zha Haiping, Yang Yun, Wang Jianming, *et al*. Transforming XPDL to Petri nets [C] //Proc of BPM International Workshops on Business Process Management. Berlin: Springer, 2008: 197-207.

[8] Westergaard M. CPN tools 4: multi-formalism and extensibility [C] //Proc of the 34th International Conference on Application and Theory of Petri Nets and Concurrency. Berlin: Springer, 2013: 400-409.

[9] Günther C W, Van Der Aalst W M P. A generic import framework for process event logs [C] //Proc of BPM International Workshops on Business Process Management. Berlin: Springer, 2006: 81-92.

[10] De Medeiros A K A, Günther C W. Process mining: using CPN tools to create test logs for mining algorithms [C] //Proc of the 6th Workshop on Practical Use of Colored Petri Nets and CPN Tools. 2005: 177-190.

[11] 查海平, 王建民, 闻立杰. 一种 Petri 网模型完备日志生成算法 [J]. 系统仿真学报, 2007, 17(1): 271-274.

[12] Van Der Aalst W M P, Weijters A J M M, Maruster L. Workflow ming: discovering process models from event log [J]. IEEE Trans on Knowledge and Data Engineering, 2004, 16(9): 1128-1142.

[13] Desel J, Esparza J. Free choice Petri nets, volume 40 of Cambridge tracts in theoretical computer science [M]. Cambridge: Cambridge University Press, 1995.

[14] Murata T. Petri nets: properties, analysis and applications [J]. Proceeding of the IEEE, 1989, 77(4): 541-580.

[15] Van Der Aalst W M P, Van Hee K. Workflow management: models, methods and systems [M]. Boston: MIT Press, 2004.

[16] De Medeiros A K A, Van Dongen B F, Van Der Aalst W M P, *et al*. Process mining: extending the α-algorithm to mine short loops, WP113 [R]. Eindhoven, The Netherlands: Eindhoven University of Technology, 2004.

[17] Wen Lijie, Van Der Aalst W M P, Wang Jianming, *et al*. Mining process models with non-free-choice constructs [J]. Data Mining and Knowledge Discovery, 2007, 15(2): 145-180.

[18] 吴哲辉. Petri 网导论 [M]. 北京: 机械工业出版社, 2006.

[19] Chrzastowski-Wachtel P, Benatalla B, Hamadi R, *et al*. A top-down Petri net-based approach for dynamic workflow modeling [C] //Proc of International Conference on Business Process Management. Berlin: Springer, 2003: 336-353.