基于 Petri 网的批量迹与过程模型校准

田银花[1),2)] 杜玉越[1)] 韩 咚[3)] 刘 伟[1)]

[1)](山东科技大学信息科学与工程学院 山东 青岛 266590)

[2)](山东科技大学信息工程系 山东 泰安 271000)

[3)](山东科技大学矿业与安全工程学院 山东 青岛 266590)

摘 要 校准是事件日志中迹与过程模型之间一致性检查的重要手段,可以精确定位偏差出现位置.但已有校准方法一次只能计算一条迹与过程模型之间的校准,若计算 m 条迹与过程模型之间的校准,需调用 m 次该方法,做大量重复工作.针对该问题,基于 Petri 网提出了一种过程模型与 m 条迹之间的批量校准方法——AoPm(Alignments of Process Model and m Traces)方法,调用 A+或 A++算法同时得到多条迹与模型之间的最优校准.以一个给定完备事件日志集和过程模型为例,基于区域的过程发现算法,挖掘事件日志中所有迹的日志模型;发现日志模型与过程模型的日志移动、模型移动和同步移动,并得到其乘积系统;计算乘积 Petri 网的可达图,得到变迁系统.提出了计算最优校准的 A+算法及 A++算法,可分别得到日志中所有迹与过程模型之间的一个最优校准和所有最优校准.对 AoPm 方法的时间复杂度和空间复杂度进行了理论分析,并与已有校准方法进行比较.当计算 m 条迹与过程模型之间的最优校准时,AoPm 方法计算乘积、变迁系统次数和所占用空间都是传统方法的 $1/m$.给出并验证了变迁系统中必定能找到日志中任意一条迹与过程模型的一个校准、一个最优校准和所有最优校准的定理,并提出了日志同步网的概念,证明了 A+算法和 A++算法的正确性.基于 ProM 平台、人工网上购物模型及生成日志集,对 AoPm 方法进行了仿真实验,并与传统校准方法进行比较分析.实验结果表明,在处理批量迹与过程模型的校准时,AoPm 方法比传统校准方法在计算变迁系统的运行时间和占用空间上,分别有指数级和多项式级的降低.AoPm 方法应用于实际复杂问题的模型与日志,说明了其适应性与健壮性.AoPm 方法突破了以往每次只对一条迹和过程模型进行校准的限制,首次实现了批量迹与模型之间的校准,提高了事件日志中迹与过程模型之间的一致性检查效率.

关键词 Petri 网;过程挖掘;日志模型;过程模型;批量迹;校准

中图法分类号 TP311 **DOI 号** 10.11897/SP.J.1016.2018.00611

Alignments Between Batch Traces and Process Models Based on Petri Nets

TIAN Yin-Hua[1),2)] DU Yu-Yue[1)] HAN Dong[3)] LIU Wei[1)]

[1)](*College of Information Science and Engineering, Shandong University of Science and Technology, Qingdao, Shandong* 266590)

[2)](*Department of Information Engineering, Shandong University of Science and Technology, Taian, Shandong* 271000)

[3)](*College of Mining and Safety Engineering, Shandong University of Science and Technology, Qingdao, Shandong* 266590)

Abstract Alignment is a main method of conformance checking between a trace in the event log and the process model, and can fix the locations of the deviations accurately. But the existing alignment method can obtain the alignments between only one trace and the process model. This method must be applied for m times if the alignments between m traces and the process model are required, and a lot of repetitive tasks have to be done. To resolve such problem, alignments of process models and m traces named AoPm are presented based on Petri nets, and this method can

收稿日期:2016-01-27;在线出版日期:2016-06-20.本课题得到国家自然科学基金(61170078,61472228)、山东省自然科学基金(ZR2014FM009)、山东省优秀中青年科学家科研奖励基金(BS2015DX010)、泰山学者建设工程专项经费资助.**田银花**,女,1982 年生,博士研究生,讲师,主要研究方向为流程挖掘、Petri 网.E-mail:skdxxtyh@163.com.**杜玉越**(通信作者),男,1960 年生,教授,博士生导师,中国计算机学会(CCF)高级会员,主要研究领域为软件工程、形式化技术和 Petri 网.E-mail:yydu001@163.com.**韩 咚**,男,1982 年生,博士研究生,讲师,主要研究方向为流程挖掘与资源管理.**刘 伟**,男,1977 年生,博士,副教授,中国计算机学会(CCF)会员,主要研究方向为 Petri 网、工作流、Web 服务.

achieve the optimal alignments between the batch traces of an event log and the process model at the same time by calling A+ or A++ algorithm. Taking a given complete event log set and a process model for example, the follow tasks were done: all of the traces in the event log were translated into an event model by an iterative algorithm for applying the theory of regions in process mining; moves on logs, moves on models and synchronous moves were found, and a product system of the event net and the process net was built. The reachable graph of the product Petri net was yielded, and the transition system of the product was built. An optimal alignment between every trace in the original event log and the process model could be obtained by A+ algorithm, and all optimal alignments between every trace in the original event log and the process model could be obtained by A++ algorithm. The time complexity and space complexity of AoPm method were analyzed theoretically and compared with the existing alignment method. It was concluded that the iteration times and the memory of the product systems and the transition systems were reduced m-fold using AoPm method than the traditional alignment method when computing the optimal alignments between m traces and the process model. The theorems of finding an alignment, an optimal alignment and all of the optimal alignments were proposed in the transition system, the log-synchronous net was presented, and the correctness of A+ and A++ algorithms could be explained. The simulations of AoPm method were carried out and compared with the traditional alignment methods based on ProM platform, the artificial model on online shopping and the generated event logs. The experiment results showed that the time taken to compute transition systems of AoPm method was reduced exponentially and the space complexity had a polynomial decline than the traditional alignment method. To show the adaptability and robustness of the proposed approach to logs and models with real-life complexity, several real-life logs and models were analyzed. AoPm method breaks through the thought of the alignment between only one trace and the process model, realizes the alignments between batch traces in the event log and the process model, and improves the efficiency of conformance checking between traces in the event log and the process model.

Keywords Petri net; process mining; log model; process model; batch traces; alignment

1 引 言

自美国宣布“大数据的研究和发展计划”以来，大数据引起了世界各国的高度关注.大数据时代的到来，不仅改变了人们的生活工作方式和企业的运作模式，甚至引起了科学研究模式的根本改变[1-2].面对大数据带来的各种挑战[3-4]，业务过程管理(Business Process Management，BPM)必将得到进一步发展[5].BPM 以业务过程和事件日志为基础，将管理技术和信息技术结合起来以提高过程管理的效率，增强企业竞争能力.

随着信息系统支持的业务过程越来越多，过程模型和事件日志的可用性越来越高[6-7].但是当日志在模型上运行时，可能会存在偏差(deviation)[8].校准(alignment)[9-10]是事件日志和过程模型之间进行一致性检查(conformance checking)的一种方法，其能够根据给定的代价函数得到事件日志中迹与过程模型之间的拟合(fitness)情况，确定出现偏差的位置，在此基础上可进一步分析导致偏差的原因.已有计算最优校准方法的步骤主要包括：首先根据给定迹生成一个日志模型，日志模型中变迁之间是全序关系；接下来用该日志模型与过程模型做乘积；然后求乘积的变迁系统；最后利用 A* 算法在变迁系统中搜索出最优校准.

已有的校准方法每次只能处理一条迹，但日志中存在多条迹，如果要计算多条迹与过程模型之间的最优校准，就要反复多次运用该方法.在计算的过程中，需多次求日志模型与过程模型的乘积以及乘积模型的变迁系统，这是一个非常复杂且重复的过程.不仅工作量庞大，而且占用的存储空间较多.

本文基于 Petri 网提出一种同时实现多条迹与

过程模型校准的方法，称作 AoPm(Alignments of Process Model and m Traces，过程模型与 m 条迹之间的批量校准)方法. 本文在已有过程模型和事件日志的基础上进行研究，且过程模型采用标签 Petri 网建模[5]. 首先使用已有过程发现算法得到事件日志中所有迹的日志模型，接下来计算日志模型与过程模型之间的乘积，然后求乘积模型的变迁系统. 一般情况下，如果事件日志中出现的活动均在过程模型中，且事件日志中每条迹长度相当，则该变迁系统与单条迹的变迁系统规模相近. 但是采用该方法，可以在同一个变迁系统中求出所有迹的最优校准. 既简化了计算变迁系统的工作量，也节省了变迁系统占用的存储空间.

目前，过程发现算法种类较多，常见的有 α 算法[11]、启发式挖掘算法[12]、遗传过程挖掘算法[13]和基于区域的挖掘算法[14]等等. 其中，诸如 α 算法和启发式挖掘算法等技术[15]不保证模型能够重演事件日志中的所有案例. 而本文中提出的批量迹与过程模型之间的校准方法要求“事件日志中的所有迹均能够被发现的模型重演”，这是该方法有效必须满足的前提条件. 因此本文采用基于区域的方法，一般情况下基于区域的方法能够表达更加复杂的控制流结构，同时不会欠拟合.

构建实例图(instance graphs)的多阶段(multi-phase)过程挖掘方法[16]，能够保证适应度为 1，即事件日志中出现的任一条迹都是过程模型的一个可能发生序列. 但该方法得到的模型是用实例 EPCs (Event-driven Process Chains)描述的，而本文所提出方法要求日志模型和过程模型采用相同的建模语言，即用标签 Petri 网建模. 应用区域理论的一种迭代算法[17]挖掘出的日志模型能够满足要求：一是事件日志的每条迹都是 Petri 网模型的一个引发序列；二是 Petri 网中任一引发序列都是日志的一条迹. 本文采用该方法从给定的完备事件日志中发现日志模型.

本文第 2 节首先对相关基本概念进行说明；第 3 节给出事件日志利用基于区域挖掘算法得到其日志模型，与给定过程模型求乘积，并计算出乘积的变迁系统；第 4 节给出 A+算法和 A++算法分别计算事件日志中所有迹与过程模型之间的一个最优校准和所有最优校准；第 5 节分析 AoPm 方法的时间和空间复杂度，并与传统方法进行比较，论证 A+算法和 A++算法的正确性；第 6 节基于 ProM 平台实现了 AoPm 方法，分别应用于人工网上购物过程模型及生成日志集和实际复杂问题领域，例证了该方法的可行性与有效性；第 7 节对本文工作进行总结和展望.

2　基本概念

很多形式化语言可以描述过程模型，其中使用较多的为 Petri 网. Petri 网[18]作为分布式系统建模和分析的工具，具有严格数学定义和强大的图形表达能力. Petri 网不仅能够描述过程静态结构，还可以模拟过程运行中动态行为，对于具有并发、异步等性质的信息系统，可以利用 Petri 网对其进行有效描述和分析.

本文基于标签 Petri 网[6]给出一种校准算法. 标签含义在于为每一个变迁分配一个相应的活动名称，如此变迁能够与实际业务中的活动对应起来.

定义 1(标签 Petri 网系统). 设 A 是一个活动名称集合. 集合 A 上的标签 Petri 网系统是一个六元组 $N=(P,T,F,\alpha,m_i,m_f)$，其中 P 是库所的有限集合，T 是变迁的有限集合，$F\subseteq(P\times T)\cup(T\times P)$ 是有向弧集合，称为流关系. $\alpha:T\to A^\tau$ 是一个变迁到标签的映射函数. m_i,m_f 分别是 N 的初始标识和结束标识. 其中，τ 标记不可见变迁(invisible transition)，$A^\tau=A\cup\{\tau\}$ 表示活动集合 A 与 $\{\tau\}$ 的并集.

为便于叙述，在不产生混淆的情况下，下文中提到的 Petri 网均指标签 Petri 网.

当前信息系统中记录了数量众多的事件，存储在日志中. 日志中记录的一个事件，称为迹[6](trace).

定义 2(迹). 设 ε 为事件空间. $\sigma\in\varepsilon^*$ 是一个有限事件序列并且每个事件只出现一次，即对于 $1\le i<j\le|\sigma|:\sigma[i]\neq\sigma[j]$，则称 σ 为迹.

迹在 Petri 网模型上重演时，可能会出现偏差，校准能够标记偏差. 给定对偏差的一个度量标准，得到最优校准[9].

定义 3(校准). 设 A 是一个活动名称集合. $\sigma\in A^*$ 是 A 上的一条迹，$N=(P,T,F,\alpha,m_i,m_f)$ 是 A 上的一个 Petri 网. 迹 σ 与网 N 之间的校准 $\gamma\in(A^{\gg}\times T^{\gg})$ 是满足以下条件的移动序列(movements)：

(1) $\pi_1(\gamma)_{\downarrow A}=\sigma$，即迹中的移动序列(忽略 $\gg$)产生该迹；

(2) $m_i\xrightarrow{\pi_2(\gamma)_{\downarrow T}}m_f$，即模型中的移动序列(忽略 $\gg$)产生一个完整引发序列.

其中：$\gg$ 表示无移动，$A^{\gg}=A\cup\{\gg\}$，$\pi_1(\gamma)_{\downarrow A}$ 表示元组序列 γ 第 1 项在 A 上的投影.

对于校准中所有元组 $(a,t)\in\gamma$，对 (a,t) 的定义如下：

(1)若 $a\in A$ 且 $t=\gg$,则为日志移动;

(2)若 $a=\gg$ 且 $t\in T$,则为模型移动;

(3)若 $a\in A$ 且 $t\in T$,则为同步移动;

(4)否则为非法移动.

3 日志模型与过程模型乘积变迁系统

校准考察过程模型与事件日志中迹之间存在偏差的情况.为了更加形象清晰地表达 AoPm 方法的思想,以给定的过程模型和事件日志为例进行说明.过程模型可以是手工建立的,也可以是通过发现得到的.此外,过程模型可以是规范化的,也可以是描述性的.

3.1 日志模型与过程模型

给定过程模型如图 1 所示.该过程模型是一个标签 Petri 网,也是一个合理的工作流网[19].

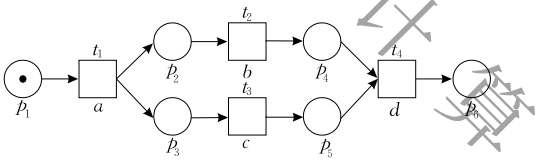


图 1 过程模型 N_p

给定事件日志 $L_1=[(\sigma_1)^3,(\sigma_2)^7]$,其中迹 $\sigma_1=\langle a,b\rangle$,迹 $\sigma_2=\langle a,c\rangle$. L_1 是一个包含 10 个案例的简单事件日志,10 个案例可以被表示为 2 个不同的迹.根据事件日志 L_1,基于区域的过程发现算法能够得到如图 2 所示的工作流网 N_l,称为日志模型.同时,在该图中将标签为 a、b、c 的变迁分别标记为 t_1'、t_2'、t_3'.

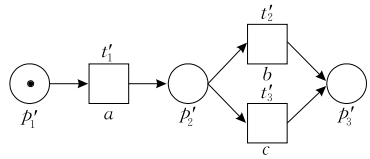


图 2 日志模型 N_l

3.2 日志模型与过程模型的乘积

根据两个 Petri 网乘积的定义[9]可以得到日志模型和过程模型之间的乘积模型.乘积模型由日志模型、过程模型以及同步变迁组成.所谓同步变迁是指在日志模型和过程模型中具有相同标签的变迁,在本例中两个模型都存在标签为 a、b、c 的变迁,因此构建乘积模型时有 3 个与之相应的同步变迁.在乘积模型中,原日志模型中的变迁 t_i' 标记更改为 $(t_i',\gg)$;原过程模型中的变迁 t_j 标记更改为 $(\gg,t_j)$;若同步变迁分别和日志模型中的 t_k'、过程模型中的 t_k 具有相同的标签,则标记为 (t_k',t_k);日志模型与过程模型中原来的库所及弧关系均保持不变;同步变迁应满足 $^\cdot(t_k',t_k)={}^\cdot(t_k',\gg)\cup{}^\cdot(\gg,t_k)$ 及 $(t_k',t_k)^\cdot=(t_k',\gg)^\cdot\cup(\gg,t_k)^\cdot$,根据该条件为同步变迁与日志模型和过程模型中的库所之间添加有向弧.通过一系列运算,得到日志模型 N_l 和过程模型 N_p 的乘积模型 N_{l*p},如图 3 所示.

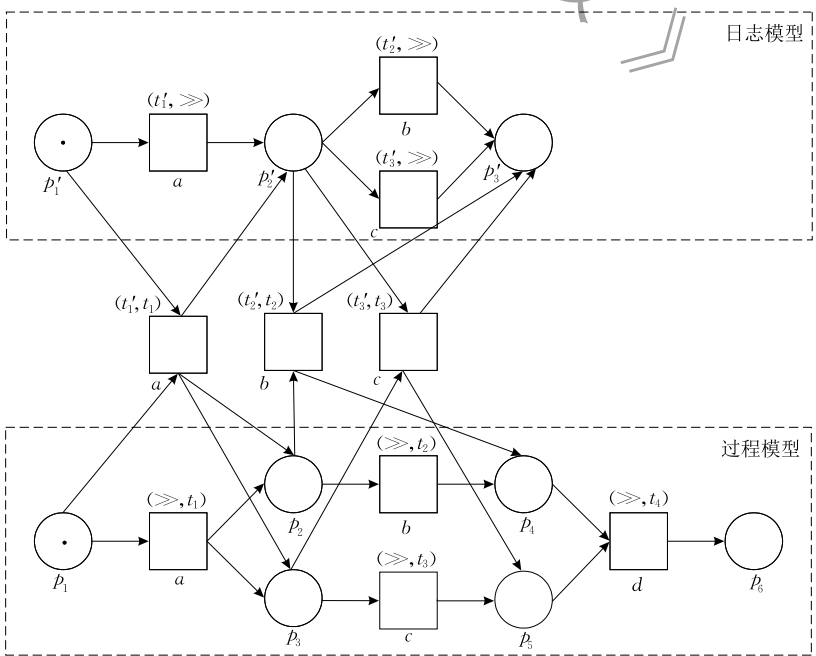


图 3 日志模型 N_l 与过程模型 N_p 的乘积模型 N_{l*p}

3.3　乘积模型的变迁系统

计算日志模型 N_l 与过程模型 N_p 之间乘积模型 N_{l*p} 的可达标识状态，得到变迁系统如图 4 所示. 乘积模型的变迁系统是一个有向图，明确地描述了乘积模型中变迁的引发过程及可达状态. 该图中，双圈结点代表乘积模型的终止标识状态；各条边上的标注为乘积模型中状态之间的引发变迁.

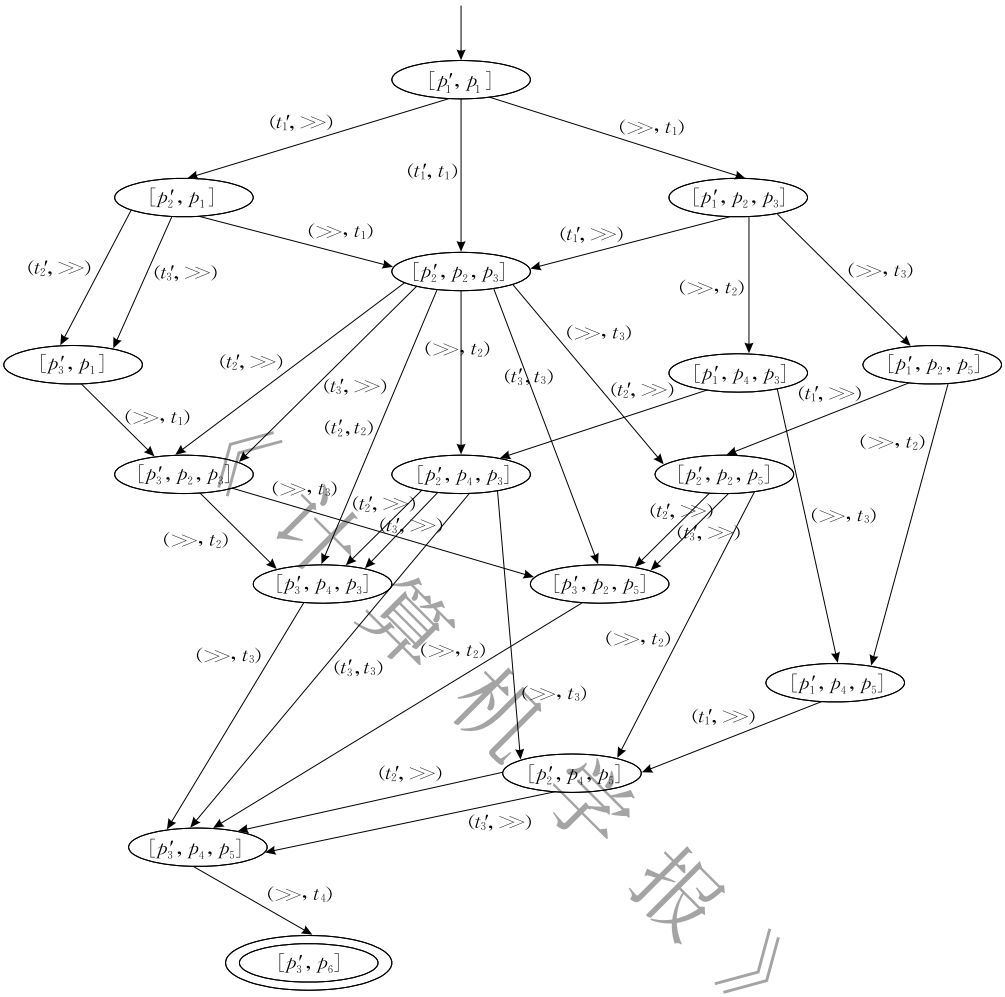


图 4　乘积模型的变迁系统 S_{l*p}

给定事件日志与过程模型，事件日志中的批量迹与过程模型之间的校准就转化为在乘积模型中搜索引发序列的问题，有效解决方法就是状态空间搜索法. 下面描述如何将计算最小代价引发序列问题转化为计算最小路径问题.

使用标准似然代价函数 $lc()$ 为变迁系统的每条边分配一个代价值，即同步移动代价值为 0，日志移动和模型移动代价值均为 1[9]. 图 4 中各边权值取值情况如下：$lc((t'_k,t_k))=0(1\leq k\leq 3)$，$lc((t'_i,\gg))=1(1\leq i\leq 3)$，$lc((\gg,t_j))=1(1\leq j\leq 4)$. 根据定义 3，可知$(t'_1,t_1)$是同步移动，而同步移动的代价值为 0. 因此，初始状态$[p'_1,p_1]$到状态$[p'_2,p_2,p_3]$的边长度记作 0；同理，由于$(t'_1,\gg)$是日志移动，初始状态$[p'_1,p_1]$到状态$[p'_2,p_1]$的边长度记作 1. 变迁系统中一条边上的标注对应着乘积模型中的一个引发变迁，故两个状态结点间一条路径的总长度，对应于该路径上引发变迁序列的总代价值. 而移动序列的总代价等价于引发序列的代价值. 因此，获取移动序列的最小代价值，可以转化为计算变迁系统中最短路径的问题.

根据上述分析，为了更清晰地描述相应最短路径查找问题，下面举例进行说明. 例如，图 4 中路径$\langle([p'_1,p_1],(t'_1,t_1),[p'_2,p_2,p_3]),([p'_2,p_2,p_3],(t'_2,t_2),[p'_3,p_4,p_3]),([p'_3,p_4,p_3],(\gg,t_3),[p'_3,p_4,p_5]),([p'_3,p_4,p_5],(\gg,t_4),[p'_3,p_6])\rangle$，提取变迁序列$\langle(t'_1,t_1),(t'_2,t_2),(\gg,t_3),(\gg,t_4)\rangle$，并将其每个变迁的第 1 列映射到相应的活动，可得如图 5(a)所示移动序列. 同理，由图 4 中的路径$\langle([p'_1,p_1],(t'_1,t_1),[p'_2,p_2,p_3]),([p'_2,p_2,p_3],(t'_3,t_3),[p'_3,p_2,p_5]),([p'_3,p_2,p_5],(\gg,t_2),[p'_3,p_4,p_5]),$

$([p_3', p_4, p_5], (\gg, t_4), [p_3', p_6])\rangle$，可得如图 5(b)所示移动序列. 而移动序列也是一个校准.

从变迁系统的初始状态到任何其他状态的任一路径都会产生移动序列，并且每个路径的总长度产生了该路径构成的移动序列的代价值. 因此，从初始状态到终止状态的一个路径产生的移动序列，若其在第一列上的投影与某条迹一致且在该条件下路径最短，则产生该迹与模型之间的一个最优校准. 如图 5(a)所示的校准即为迹 $\sigma_1=\langle a,b\rangle$ 与模型 N_p 之间的一个最优校准. 图 4 所示变迁系统中相应的变迁序列，对应一条从初始状态到终止状态的最短路径. 同理，图 5(b)所示的校准即为迹 $\sigma_2=\langle a,c\rangle$ 与模型 N_p 之间的一个最优校准.

a	b	$\gg$	$\gg$
a	b	c	d
t_1	t_2	t_3	t_4

(a) 迹 $\sigma_1=\langle a,b\rangle$ 产生的移动序列

a	c	$\gg$	$\gg$
a	c	b	d
t_1	t_3	t_2	t_4

(b) 迹 $\sigma_2=\langle a,c\rangle$ 产生的移动序列

图 5 移动序列

下节将给出 A+算法及 A++算法，分别可以得到事件日志中批量迹与过程模型之间的一个最优校准和所有最优校准.

4 计算最优校准算法

得到日志模型与过程模型之间乘积变迁系统之后，在该变迁系统基础上，根据标准似然代价函数，利用查找最短路径的思想，可以找到给定迹与过程模型之间的最优校准. 本节给出 A+算法和 A++算法分别计算事件日志中所有迹与过程模型之间的一个最优校准以及所有最优校准.

4.1 计算一个最优校准算法——A+算法

图 4 给出日志模型和过程模型之间乘积的变迁系统. 下面描述根据该变迁系统得到事件日志中每条迹与过程模型之间一个最优校准的算法. 该算法主要思想是针对所有迹在变迁系统中从源结点开始，做如下搜索工作：第 1 步，将源结点放入优先队列并作为当前结点；第 2 步，得到当前结点的后继结点放入优先队列，并计算它们的代价值以及前缀最优校准，从优先队列中选择代价值最小并且前缀最优校准第 1 列在活动集合上的投影是迹的前缀的一个结点，作为当前结点；第 3 步，重复执行第 2 步，直到当前结点在目标结点集中. 最后一个当前结点上标注的校准就是需要查找的一个最优校准. 重复上述 3 个步骤能够得到所有迹与过程模型的一个最优校准. 下面给出详细执行步骤伪代码. 首先，给出所需数据结构以及相应变量、函数的声明：

sourcenode：源结点；

targetNodesSet：目标结点集；

pqueue：优先队列，存储变迁系统中最优校准路径中的结点；

visitedNodesSet：已访问结点集；

currnode：当前访问结点；

succnode：后继结点；

successorNodesSet(*node*)：*node* 结点的后继结点集；

move(*node*1,*node*2)：结点 *node*1 和 *node*2 之间的移动；

lc(*move*(*node*1,*node*2))：移动 *move*(*node*1,*node*2)的代价值，默认使用标准似然代价函数定义；

cost(*node*)：结点 *node* 的代价值；

alignment(σ_i)(*node*)：源结点到结点 *node* 路径上的移动序列；

prefix(σ_i)：σ_i 的前缀集合.

算法 1(A+算法). 计算每条迹与过程模型之间的一个最优校准.

输入：日志模型与过程模型乘积的变迁系统

输出：事件日志中每条迹与过程模型之间的一个最优校准

1. FOR all $\sigma_i\in\Sigma$ DO
2. {*pqueue*←{*sourcenode*};
3. *visitedNodesSet*←∅;
4. WHILE *pqueue*≠∅ DO
5. {FOR all *node*∈*successorNodesSet*(*pqueue*) DO
6. {IF $\exists\pi_1(alignment(\sigma_i)(node))_{\downarrow A}\in prefix(\sigma_i)$ and *cost*(*node*)≤*cost*(*successorNodesSet*(*pqueue*)) THEN
7. *currnode*←*node*;}
8. IF *currnode*∈*targetNodesSet* THEN
9. RETURN *alignment*(σ_i)(*node*);
10. ELSE
11. FOR all *succnode*∈*successorNodesSet*(*currnode*) DO
12. {IF *succnode*∈*visitedNodesSet* THEN
13. {IF *cost*(*succnode*)>*cost*(*currnode*)+*c*(*move*(*currnode*,*succnode*)) THEN
14. {*cost*(*succnode*)←*cost*(*currnode*)+*lc*(*move*(*currnode*,*succnode*));
15. *pqueue*←*pqueue*∪{*succnode*};
16. *alignment*(σ_i)(*succnode*)←*alignment*(σ_i)(*currnode*)&*move*(*currnode*,*succnode*);}}
17. ELSE
18. {*visitedNodesSet*←*visitedNodesSet*∪{*succnode*};
19. *cost*(*succnode*)←*cost*(*currnode*)+*lc*(*move*(*currnode*,*succnode*));

20. $pqueue \leftarrow pqueue \cup \{succnode\}$；

21. $alignment(\sigma_i)(succnode) \leftarrow alignment(\sigma_i)(currnode) \& move(currnode, succnode)$；}}}}

以迹 $\sigma_1 = \langle a, b \rangle$ 和图 4 所示 S_{l*p} 的变迁系统为例. 算法 1 的具体执行过程是在变迁系统 S_{l*p} 上进行查找，在不产生混淆情况下，查找过程中省略原图中结点和边上的标注. 另外，在每个叶子结点旁给出该结点的最优前缀校准以及代价值. 本例中，对于代价的度量采用标准似然代价函数，即在校准序列中，同步移动的代价值为 0，日志移动和模型移动的代价值各为 1. 整个校准的代价值为序列中每个移动的代价值的累加和. 算法 1 具体执行过程如图 6 所示.

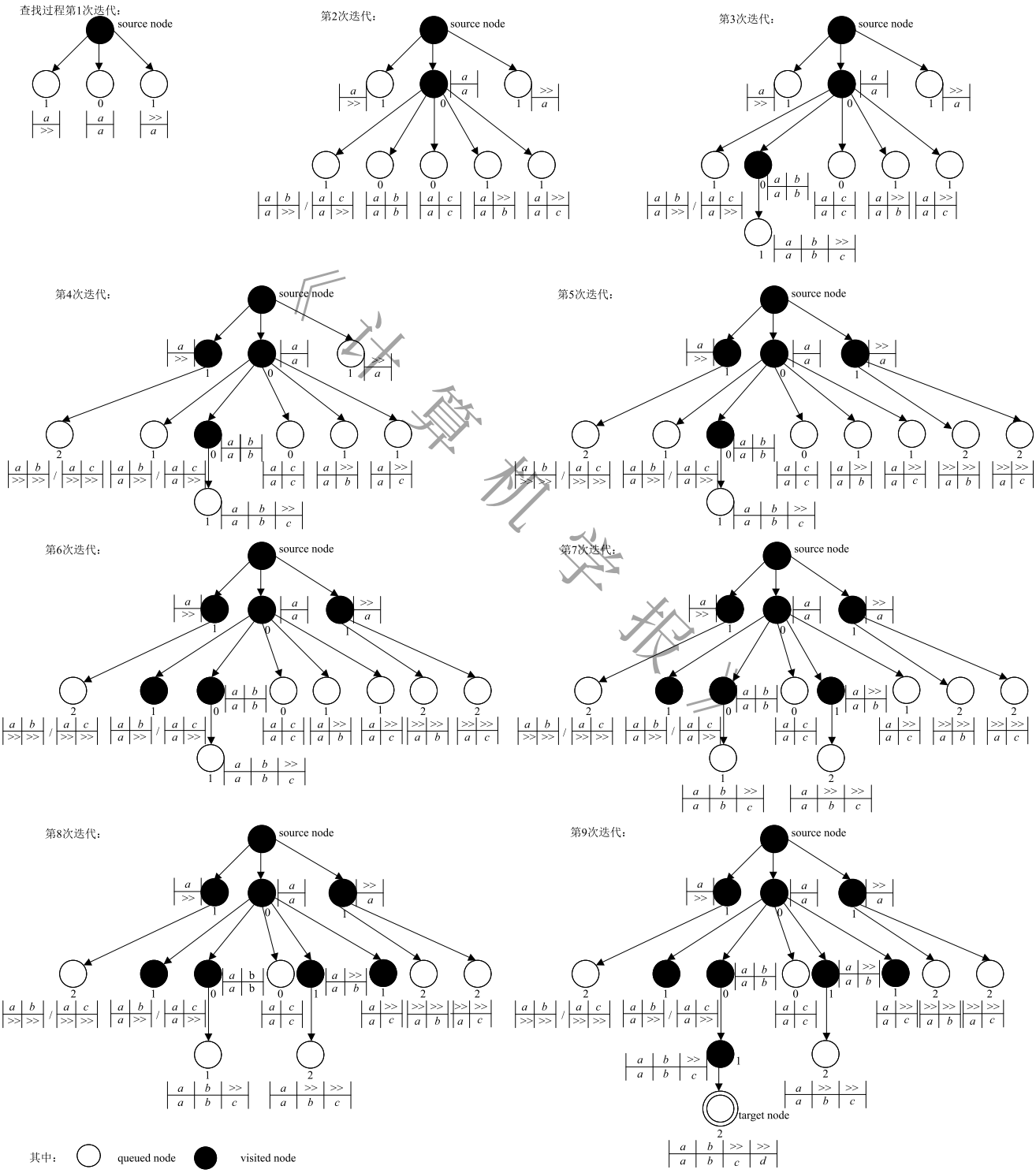


图 6　一个最优校准查找过程

根据算法可知结点$[p_1', p_1]$为源结点，结点$[p_3', p_6]$为目标节点.首先将源结点放入优先队列，取源结点作为当前结点进行访问，将源结点的后继结点放入队列中，并计算出后继结点的代价值及相应的前缀校准；这是第1次迭代，如图6所示.接下来选择代价值最小的结点$[p_2', p_2, p_3]$作为当前结点进行访问，将其后继结点放入队列中，并计算出后继结点的代价值及相应的前缀校准，这是第2次迭代.在队列中取代价值最小的结点$[p_3', p_4, p_3]$作为当前结点进行计算，这是第3次迭代，计算结果如图6所示.下一次迭代时，在队列中，虽然结点$[p_3', p_2, p_5]$的代价值最小，但是其前缀校准序对第1列在A上的投影，不是迹σ_1的前缀，因此不能作为当前结点.其余结点的代价值相同，按照队列先进先出的特点，结点$[p_2', p_1]$为当前结点.在计算结点$[p_2', p_1]$的后继结点时，发现结点$[p_2', p_2, p_3]$亦为其后继结点，但因为通过结点$[p_2', p_1]$的代价值和两结点之间边的代价值之和大于结点$[p_2', p_2, p_3]$原来的代价值，因此结点$[p_2', p_2, p_3]$的代价值和前缀校准保留原来值，不被更新.然后，依次选择结点$[p_1, p_2, p_3]$、$[p_3', p_2, p_3]$、$[p_2', p_4, p_3]$、$[p_2', p_2, p_5]$作为当前节点进行迭代，具体执行过程参照图6.最后，以结点$[p_3', p_4, p_5]$作为当前节点，查找其后继结点，找到后继结点$[p_3', p_6]$为目标结点.查找过程结束，求得的目标结点$[p_3', p_6]$上标注的校准就是要查找的一个最优校准.

同理，通过该算法也可以得到$\sigma_2=\langle a, c\rangle$的一个最优校准.

4.2 计算所有最优校准算法——A++算法

算法2计算事件日志中每条迹与过程模型之间的所有最优校准.该算法的主要思想是对每条迹在变迁系统中从源结点开始，做如下搜索工作：第1步，将源结点放入优先队列并作为当前结点；第2步，得到当前结点的后继结点放入优先队列，并计算它们的代价值以及前缀最优校准，从优先队列中选择代价值最小并且前缀最优校准第1列在活动集合上的投影是迹的前缀的所有结点，作为当前结点；第3步，重复执行第2步，直到当前结点在目标结点集中.最后当前结点上标注的校准就是需要查找的最优校准.重复上述3个步骤能够得到事件日志中全部迹与过程模型之间的所有最优校准.下面给出详细执行步骤伪代码.数据结构以及相应的变量、函数的声明和算法1基本一致，只是每个结点对应的前缀最优校准不只有一个，因此将算法1中$alignment(\sigma_i)(node)$改为数组$alignment(\sigma_i)(node)(j)$，其中$1\leq j$.

算法2(A++算法). 计算每条迹与过程模型的所有最优校准.

输入：日志模型与过程模型乘积的变迁系统

输出：事件日志中每条迹与过程模型之间的所有最优校准

```
1.  FOR all σ_i ∈ Σ DO
2.  {pqueue ← {sourcenode};
3.  visitedNodesSet ← ∅;
4.  solutionNodesSet ← ∅;
5.  solutionFound ← false;
6.  distanceLim ← +∞;
7.  WHILE pqueue ≠ ∅ DO
8.  {FOR all node ∈ successorNodesSet(pqueue) DO
9.   {IF ∃π_1(alignment(σ_i)(node)(k))↓_A ∈ prefix(σ_i)
     and cost(node) ≤ cost(successorNodesSet(pqueue))
     THEN
10.   currnode ← node;}
11.  IF π_1(alignment(σ_i)(node)(k))↓_A ≤ distanceLim
     THEN
12.   {IF currnode ∈ targetNodesSet THEN
13.   {solutionFound ← true;
14.    distanceLim ← π_1(alignment(σ_i)(node)(k))↓_A;
15.    solutionNodesSet ← solutionNodesSet ∪
                          {currnode};}
16.   FOR all succnode ∈ successorNodesSet(currnode)
      DO
17.   {IF succnode ∈ visitedNodesSet THEN
18.    IF cost(succnode) > cost(currnode) +
       lc(move(currnode, succnode)) THEN
19.    {cost(succnode) ← cost(currnode) +
       lc(move(currnode, succnode));
20.    pqueue ← pqueue ∪ {succnode};
21.    alignment(σ_i)(succnode)(k) ← alignment(curr-
       node)(k) & move(currnode, succnode);}
22.   ELSE
23.    {IF cost(succnode) = cost(currnode) +
       lc(move(currnode, succnode)) THEN
24.     {visitedNodesSet ← visitedNodesSet ∪
                           {succnode};
25.      cost(succnode) ← cost(currnode) +
         lc(move(currnode, succnode));
26.      pqueue ← pqueue ∪ {succnode};
27.      alignment(σ_i)t(succnode)(k) ← alignment(σ_i)
         (currnode)(k) & move(currnode, succnode);}
28.     ELSE
29.     {visitedNodesSet ← visitedNodesSet ∪
                           {succnode};
```

30. $cost(succnode) \leftarrow cost(currnode) +$ $lc(move(currnode, succnode))$;
31. $pqueue \leftarrow pqueue \cup \{succnode\}$;
32. $alignment(\sigma_i)(succnode)(j) \leftarrow alignment(\sigma_i)(currnode)(j) \& move(currnode, succnode)$; }}
33. ELSE
34. BREAK WHILE; }}
35. IF $solutionFound$ = true THEN
36. FOR all $solutionNode \in solutionNodesSet$ and $solutionNode \in targetNodesSet$ DO
37. RETURN $alignment(\sigma_i)(solutionNode)$;}

对上述两个算法进行分析，如果将算法 1 中存储后继结点的数据结构由队列改为栈，那么算法 1 的效率将会有一定程度的提高. 算法 1 查找迹的一个最优校准，类似图的深度优先遍历；算法 2 查找迹的所有最优校准，类似图的广度优先遍历. 因此两个算法的时间复杂度都和图的规模有关系，且其最坏时间复杂度分别和图的深度优先遍历和广度优先遍历复杂度一致.

5　AoPm 方法性能分析

已知 m 条迹与过程模型，计算它们之间的校准需要生成 m 条迹的日志模型、求日志模型与过程模型的乘积及变迁系统、运行 A+算法及 A++算法等步骤. 整个求解过程，称为 AoPm 方法. 接下来，分析该方法的复杂度，并对其正确性给予说明. A++算法实际为 A+算法的扩展，算法的基本思想一致. 重点分析 A+算法的复杂度及正确性.

5.1　复杂度分析

传统校准方法每次只计算一条迹和过程模型之间的最优校准. 假设事件日志共有 m 条迹，每条迹包含的活动数为 t_i 个($1 \leq i \leq m$)，过程模型中有 p 个变迁. 计算 m 条迹与过程模型之间的最优校准时，首先要生成 m 个日志模型，每个模型对应一条迹，日志模型的变迁数和迹的活动数相同，分别为 t_i 个. 然后，计算每个日志模型与过程模型的乘积模型. 根据两个 Petri 网乘积的定义[9]，假设每个日志模型与过程模型有 s_i 个同步变迁($0 \leq s_i \leq \min(t_i, p)$)，则日志模型和过程模型的乘积模型共 $t_i + p + s_i$ 个变迁. 最后，分别得到 m 个乘积模型的变迁系统，对每个变迁系统运用 A* 算法，即可得到每条迹与过程模型之间的一个最优校准.

经分析，求解过程中，共得到 m 个乘积模型，每个乘积模型的变迁数为 $t_i + p + s_i$. 生成 m 个变迁系统，运用 A* 算法 m 次.

依据上述假设，分析 AoPm 方法的性能. 首先采用挖掘算法生成日志模型，该模型至多有 t_s 个变迁($\max(t_i) \leq t_s \leq mt_i, 1 \leq i \leq m$). 然后计算乘积模型，假设日志模型与过程模型有 s_s 个同步变迁($0 \leq s_s \leq \min(t_s, p)$)，则乘积模型共 $t_s + p + s_s$ 个变迁. 最后，生成该乘积模型的变迁系统，在变迁系统中运用 A+算法，即可得到 m 条迹各自与过程模型之间的一个最优校准.

分析可知，AoPm 方法求解过程中，只得到一个乘积模型，乘积模型的变迁数为 $t_s + p + s_s$. 只需生成一个变迁系统，运用 A+算法一次.

一般情况下，事件日志中出现的不同活动数和过程模型中变迁数相当，不会有太多偏差. 因此，由一条迹生成的日志模型与过程模型所得的乘积，和多条迹挖掘出的日志模型与过程模型所得的乘积具有相同数量级的变迁数，即 $t_i + p + s_i \approx t_s + p + s_s$. 由乘积得到的变迁系统占用的存储空间也具有相同的数量级. 且变迁系统中结点个数随着乘积模型中变迁个数的增加而增加，但二者之间的函数关系难以确定，假定变迁系统中结点个数为 n 个. 可见，批量迹与过程模型的校准比单条迹与过程模型的校准在计算乘积及变迁系统时节约了大量计算时间和存储空间.

另外，A+算法其实是在 A* 算法基础上进行的扩展. A* 算法执行一次只能计算出一条迹与过程模型之间的校准，而 A+算法一次可计算出 m 条迹与过程模型之间的最优校准. 计算一条迹与过程模型的最优校准时，与 A* 算法相比，A+算法需对当前校准结果第 1 列在活动集合 A 上的投影是否是当前迹的最优前缀校准进行判断. 此操作对于计算一条迹与过程模型的最优校准时间复杂度所造成影响可忽略不计. 因此，A+算法的时间复杂度是 A* 算法的 m 倍. 在上述分析基础上可知，计算 m 条迹与过程模型的最优校准时，传统方法要调用 A* 算法 m 次，而 AoPm 方法仅需调用一次 A+算法. 因此，传统方法和 AoPm 方法计算一个最优校准所花费时间的数量级相同.

虽然计算 m 条迹与同一过程模型的一个最优校准时，分别使用 1 次 A+算法和 m 次 A* 算法时间复杂度相同，但 AoPm 方法主要节省了日志模型与过程模型计算乘积及变迁系统时所花费时间及所占用空间.

由此可知，计算 m 条迹与过程模型的一个最优

校准时，比较传统方法和 AoPm 方法的复杂度，结果如表 1 所示.

表 1 传统方法和 AoPm 方法复杂度对比

各项复杂度比较	传统方法	AoPm 方法
乘积模型中变迁个数	$O(t+p+s)$	$O(t+p+s)$
变迁系统中结点的个数	$O(n)$	$O(n)$
乘积模型的个数	m	1
变迁系统的个数	m	1
查找一个最优校准算法的时间复杂度	$O(n^2)\times m$	$O(mn^2)\times 1$

5.2 正确性分析

本节对 AoPm 方法能否正确实现计算事件日志中所有迹与过程模型的最优校准进行分析. 在本文的引言部分已经论述了本方法所使用的过程发现算法保证挖掘得到的日志模型能够重演生成该模型的事件日志的任意一条迹. 即根据事件日志 $L_1=\langle\sigma_1,\sigma_2,\sigma_3,\cdots,\sigma_n\rangle$ 挖掘出日志模型 N_l，N_l 能够完全正确重演迹 $\sigma_1,\sigma_2,\sigma_3,\cdots,\sigma_n$，也就是迹 $\sigma_1,\sigma_2,\sigma_3,\cdots,\sigma_n$ 中每个活动在 N_l 中对应的变迁组成的序列分别是 N_l 中的一个完整引发序列.

假定事件日志 $L_1=\langle\sigma_1,\sigma_2,\sigma_3,\cdots,\sigma_n\rangle$，根据基于区域的过程发现算法得到日志模型记作 $N_1=(P_1,T_1,F_1,\alpha_1,m_{i,1},m_{f,1})$，已有过程模型为 $N_2=(P_2,T_2,F_2,\alpha_2,m_{i,2},m_{f,2})$，二者乘积模型为 $N_3=N_1\otimes N_2=(P_3,T_3,F_3,\alpha_3,m_{i,3},m_{f,3})$，乘积的变迁系统为 S_3. 为便于描述，对 S_3 每条路径上的标记内容做如下运算：设原标记为 (x,y)，若 $x=\gg$，定义新标记为 (x,y)；否则，新标记为 $(\alpha(x),y)$. S_3 中从源结点到目标结点的所有路径采用其所遍历边的新标记来表示，所有路径组成的集合记作 Λ. Γ_{L_1,N_2} 记作日志 L_1 与模型 N_2 之间所有校准的集合. $\Gamma^{o}_{L_1,N_2,lc}$ 记作日志 L_1 与模型 N_2 之间基于标准似然代价函数 $lc()$ 的所有最优校准的集合.

定理 1. 对 $\gamma_i\in\Gamma_{L_1,N_2}$，$\exists\lambda_j\in\Lambda$，有 $\lambda_j=\gamma_i$，其中 $1\leqslant i\leqslant|\Gamma_{L_1,N_2}|\wedge 1\leqslant j\leqslant|\Lambda|$（$|\Gamma_{L_1,N_2}|$ 记录校准集 Γ_{L_1,N_2} 的长度，$|\Lambda|$ 记录路径集合 Λ 的长度）.

证明. 对 $\forall\sigma_k\in L_1\wedge 1\leqslant i\leqslant|\Gamma_{L_1,N_2}|$，$N_2$ 中 $\exists$ 变迁引发序列 $t_1't_2'\cdots t_m'$，使得 $m_{i,1}[t_1't_2'\cdots t_m'\rangle m_{f,1}$ 且 $\sigma_k=\langle\alpha(t_1'),\alpha(t_2'),\cdots,\alpha(t_m')\rangle$. 因为 N_1 是合理的，即 N_1 具有可正确完成性，N_1 中必 $\exists$ 变迁引发序列 $t_1t_2\cdots t_k$，使得 $m_{i,2}[t_1t_2\cdots t_k\rangle m_{f,2}$. 所以 N_3 中 $\exists$ 变迁引发序列 $(t_1',\gg)(t_2',\gg)\cdots(t_m',\gg)\cdots(\gg,t_1)(\gg,t_2)\cdots(\gg,t_k)$，使得 $m_{i,3}[(t_1',\gg)(t_2',\gg)\cdots(t_m',\gg)\cdots(\gg,t_1)(\gg,t_2)\cdots(\gg,t_k)\rangle m_{f,3}$. 即 $\exists\lambda_j\in\Lambda$，$\lambda_j=\langle(\alpha(t_1'),\gg),(\alpha(t_2'),\gg),\cdots,(\alpha(t_m'),\gg),\cdots,(\gg,t_1),(\gg,t_2),\cdots,(\gg,t_k)\rangle$. λ_j 满足：(1) $\pi_1(\lambda_j)_{\downarrow A}=\sigma_k$；(2) $m_{i,2}\xrightarrow{\pi_2(\lambda_j)_{\downarrow T}}m_{f,2}$. 所以 λ_j 为迹 σ_k 与模型 N_2 之间的一个校准，即 $\lambda_j\in\Gamma_{L_1,N_2}$. 因此，$\exists\gamma_i\in\Gamma_{L_1,N_2}$，有 $\lambda_j\in\Lambda$，使得 $\lambda_j=\gamma_i$. 证毕.

定理 1 说明乘积变迁系统 S_3 中至少存在一条路径，路径边上的标记是日志 L_1 指定迹 σ_k 与 N_2 之间的一个校准 γ_i. 上述证明过程中，找到的校准是情况最坏的一个校准，即存在偏差最多的一类校准. 该校准由两部分组成，一部分是由只引发日志模型中的变迁得到，一部分是由只引发过程模型中的变迁得到. 由定理 1 可知，变迁系统中至少存在一条从源结点到目标结点的路径是日志中某条迹与过程模型之间的校准.

例如 $\sigma_1=\langle a,b\rangle$ 与过程模型 N_p 在变迁系统图中能找到一条如图 7 所示路径，且可得如图 8 所示的一个校准.

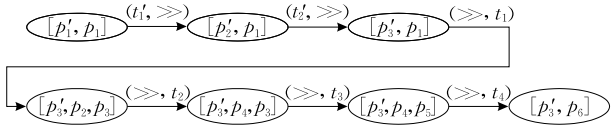


图 7 查找校准路径图

a	b	$\gg$	$\gg$	$\gg$	$\gg$
$\gg$	$\gg$	a	b	c	d
		t_1	t_2	t_3	t_4

图 8 迹 $\sigma_1=\langle a,b\rangle$ 与过程模型 N_p 的一个校准

定义 4（日志同步网）. 设 A 是一个活动名称集合. $N_1=(P_1,T_1,F_1,\alpha_1,m_{i,1},m_{f,1})$ 和 $N_2=(P_2,T_2,F_2,\alpha_2,m_{i,2},m_{f,2})$ 是 A 上的两个 Petri 网. N_1 和 N_2 的日志同步网记作 Petri 网 $N_4=(P_4,T_4,F_4,\alpha_4,m_{i,4},m_{f,4})$，其中：

(1) $P_4=P_1$；

(2) $T_4=\{(t_1,\gg)\mid t_1\in T_4\}\cup\{(t_1,t_2)\in T_1\times T_2\mid\alpha(t_1)=\alpha(t_2)\neq\tau\}$；

(3) $F_4:(P_4\times T_4)\cup(T_4\times P_4)\to IN$，其中：

① $F_4(p_1,(t_1,\gg))=F_1(p_1,t_1)\cup F_4((t_1,\gg),p_1)=F_1(t_1,p_1)$，若 $p_1\in P_1\wedge t_1\in T_1$；

② $F_4(p_1,(t_1,t_2))=F_1(p_1,t_1)\cup F_4((t_1,t_2),p_1)=F_1(t_1,p_1)$，若 $p_1\in P_1\wedge(t_1,t_2)\in T_4\cap(T_1\times T_2)$；

③ $F_4(x,y)=0$，其他情况；

(4) $\alpha_4:T_4\to A^{\tau}$，若 $\alpha_4((t_1,t_2))\in T_4$，则 $\alpha_4((t_1,t_2))=\alpha(t_1)$；

(5) $m_{i,4}=m_{i,1}$；

(6) $m_{f,4}=m_{f,1}$.

定理 2.　对 $\forall\lambda_i\in\Lambda$，有 $\lambda_i\in\Gamma_{L_1,N_2}$，其中 $1\leq i\leq|\Lambda|$（$|\Lambda|$ 记录路径集合 Λ 的长度）.

证明.　$\forall\lambda_i\in\Lambda_1$，设 $\pi_1(\lambda_i)_{\downarrow A}=\sigma_x$，其中 $\sigma_x=\langle a_1,a_2,\cdots,a_m\rangle$. 若 $\sigma_x\in L_1$，则 λ_i 满足：$\pi_1(\lambda_i)_{\downarrow A}=\sigma_x\in L_1$；又因为 $\lambda_i\in\Lambda$，所以 $m_{i,2}\xrightarrow{\pi_2(\lambda_i)_{\downarrow T}}m_{f,2}$. 因此 $\lambda_i\in\Gamma_{L_1,N_2}$，问题得证.

假设 $\sigma_x\notin L_1$，因为 $\lambda_i\in\Lambda$，所以 λ_i 可对应 N_3 的一个变迁引发序列，记作 $(t'_{31},t_{31})(t'_{32},t_{32})\cdots(t'_{3k},t_{3k})$. 取其中 $t'_{3i}\neq\gg$ 的变迁组成一个新序列 $(t'_{41},t_{41})'(t'_{42},t_{42})'\cdots(t'_{4j-1},t_{4j})'$，则为 N_4 的一个变迁引发序列，且 $\sigma_x=\pi_1((\alpha((t'_{41},t_{41})'),\alpha((t'_{42},t_{42})'),\cdots,\alpha((t'_{4j},t_{4j})')))_{\downarrow A}$. 设 $t'_{11}\in T_1\wedge(t'_{41},t_{41})\in T_4$，若 $\alpha_1(t'_{11})=\alpha_4(t'_{41},t_{41})$，则 $t'_{11}=(t'_{41},t_{41})\wedge t'_{11}=(t'_{41},t_{41})'$. 即 $\forall a\in\sigma_x$，若 $(t'_i,\gg)\in T_4\wedge\alpha((t'_i,\gg))=a$ 或 $(t'_i,t_i)\in T_4\wedge\alpha((t'_i,t_i))=a$，$\exists t'_i\in T_1\wedge\alpha(t'_i)=a$. 所以 N_4 中 $\forall m_{i,4}[(t'_{41},t_{41})'(t'_{42},t_{42})'\cdots(t'_{4j},t_{4j})'\rangle m_{f,4}$，$N_1$ 中 $\exists m_{i,1}[t'_{11}t'_{12}\cdots t'_{1j}\rangle m_{f,1}$，使得 $\langle\alpha((t'_{41},t_{41})'),\alpha((t'_{42},t_{42})'),\cdots,\alpha((t'_{4j},t_{4j})')\rangle=\langle\alpha(t'_{11}),\alpha(t'_{12}),\cdots,\alpha(t'_{1j})\rangle$. 这与假设 $\sigma_x\notin L_1$ 是相矛盾的. 显然假设不成立. 即 $\lambda_i\in\Gamma_{L_1,N_2}$.　　证毕.

定理 2 说明 T_1 中任选一条从源结点到目标结点的路径，该路径边上的标记组成的序列 λ_i 是 L_1 中某条迹 σ_x 与 N_2 之间的一个校准. 例如第 3 节中示例，其只保留原日志模型 N_l 中的库所、变迁、同步变迁以及它们之间的弧，得到子网模型如图 9 所示.

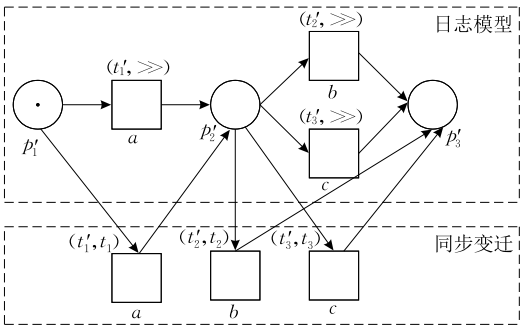


图 9　子网模型 N_4

子网模型 N_4 虽与日志模型 N_l 的变迁引发序列不同，但是由它们的变迁引发序列生成的事件日志集是完全相同的. 由定理 2 可知，变迁系统中任选一条路径可得到日志中某条迹与过程模型之间的一个校准.

定理 3.　设 Λ 是变迁系统所有路径组成的集合，Γ_{L_1,N_2} 是日志 L_1 与模型 N_2 之间所有校准的集合，有 $\Lambda=\Gamma_{L_1,N_2}$.

证明.　根据定理 2，$\Lambda\subseteq\Gamma_{L_1,N_2}$ 成立.

假设 $\exists\sigma_i$，$\exists\gamma_1\in\Gamma_{L_1,N_2}$ 但 $\gamma_1\notin\Lambda$. 即 γ_1 对应的序列不是 N_3 的一个变迁引发序列. γ_1 是 σ_i 与 N_2 之间的一个校准，所以 γ_1 满足 (1) $\pi_1(\gamma_1)_{\downarrow A}=\sigma$；(2) $m_{i,2}\xrightarrow{\pi_2(\gamma_1)_{\downarrow T}}m_{f,2}$. γ_1 为 N_3 的一个变迁引发序列. 即 $\gamma_1\in\Lambda_1$，因此假设不成立. 故 $\Gamma_{L_1,N_2}\subseteq\Lambda$.　　证毕.

定理 3 说明 T_1 中包含 L_1 任一条迹 σ_i 与 N_2 之间的所有校准. 由定理 3 可知，通过搜索变迁系统中从源结点到目标结点的路径，可以找到日志中所有迹与过程模型之间的所有最优校准.

推论 1.　对于 $\forall\sigma_i\in L_1$，A+ 算法计算结果 $alignment(\sigma_i)\in\Gamma^{o}_{L_1,N_2,lc}$.

A+算法能找到 L_1 中所有迹与 N_2 之间的一个最优校准. 根据定理 3，T_1 包含了任一条迹 σ_i 与 N_2 之间的所有校准. 其中肯定也包含了 L_1 中所有迹与 N_2 的最优校准. A+算法是在改进 A* 算法的基础上得到的，在查找的过程中，计算了每个结点的代价值和最优校准. A+算法既保证当前搜索的结点代价值最小又保证该结点对应的前缀校准是迹 σ_i 的前缀，因此 A+算法能找到 L_1 中迹 σ_i 与 N_2 之间的一个最优校准. 对每条迹都如此进行搜索最优校准的工作，最终 A+算法能找到 L_1 中所有迹与 N_2 之间的一个最优校准.

推论 2.　对于 $\forall\sigma_i\in L_1$，A++ 算法计算结果 $\sum_{i=1}^{|L_1|}alignment(\sigma_i)=\Gamma^{o}_{L_1,N_2,lc}$.

A++算法能找到 L_1 中全部迹与 N_2 之间的所有最优校准. 同理，根据定理 3，变迁系统中包含了事件日志中全部迹与过程模型之间的所有校准. A++ 算法记录了所有代价值最小且前缀校准是迹 σ_i 的前缀的结点，因此肯定能找到所有的最优校准.

6　仿真实验

给定事件日志中批量迹与过程模型，本文第 4 节提出 A+算法和 A++算法计算它们之间的最优校准，是 AoPm 方法的核心思想. 本节给出一些实验结果来评价 AoPm 方法，并与传统校准方法进行比较. 本节所做实验均基于 ProM 平台[20]. 运行 ProM 平台的计算机至少应该具有 Intel Core 3.20 GHz 处理器，1 GB 的 Java 虚拟内存.

ProM 是一个完全插件式环境，可通过添加插

件来扩展其功能. 在 ProM 平台上实施 AoPm 方法，将该方法用工具包“Alignments of Process Model and *m* Traces”实现. 该工具包能够实现 AoPm 方法所述全部功能，即计算过程模型与批量迹之间的最优校准. 给定过程模型及事件日志，AoPm 方法的具体实现步骤包括：(1) 基于区域的过程发现算法，挖掘事件日志中所有迹的日志模型；(2) 得到日志模型与过程模型的乘积系统；(3) 计算乘积系统的可达图，得到其变迁系统；(4) 运用 A＋算法及 A＋＋算法，分别得到日志中所有迹与过程模型之间的一个最优校准和所有最优校准.

本节对两组实验进行了分析. 在第 1 组实验中，使用人工模型与日志进行仿真，显示 AoPm 方法的可行性，及较传统校准方法的优越性；第 2 组实验显示了 AoPm 方法处理现实生活模型与日志的可用性和适应性. 人工仿真实验在 6.1 节进行说明，而实际案例分析在 6.2 节中进行解释.

6.1 人工日志与模型

本组实验的目的在于评价 AoPm 方法计算批量迹与人工过程模型之间校准的健壮性. 仿真实验考察的重点在于：内存效率和计算时间.

分析现在较为流行的网上购物模式，人工创建一个过程模型，如图 10 所示. 该网上购物流程 Petri 网模型中，首先用户登录网上购物平台(login)选购商品，选中商品后可以立即购买(buy now)；也可以先将商品放入购物车(add to cart)，而且往购物车中放入商品的过程可以反复执行，以便同时购买多样商品，选择结束后进入购物车(go to cart)进行结算(settle accounts). 无论是立即购买还是在购物车中进行结算，均会生成订单(generate order)，之后需要用户确认收货地址信息(confirm address)，与此同时用户还需确认账单的商品信息(confirm order). 信息确认后提交订单(submit order)，此时可以放弃购买(cancel order)，交易结束；也可以进入付款环节，而实施真正的付款(pay)之前要先进行付款方式的选择(choose method). 如果付款成功(succeed)，卖家会发货，用户收到商品后，要进行收货确认(confirm receipt)，之后应对商品给予评价(estimate)，交易结束；如果付款失败(fail)，交易直接结束.

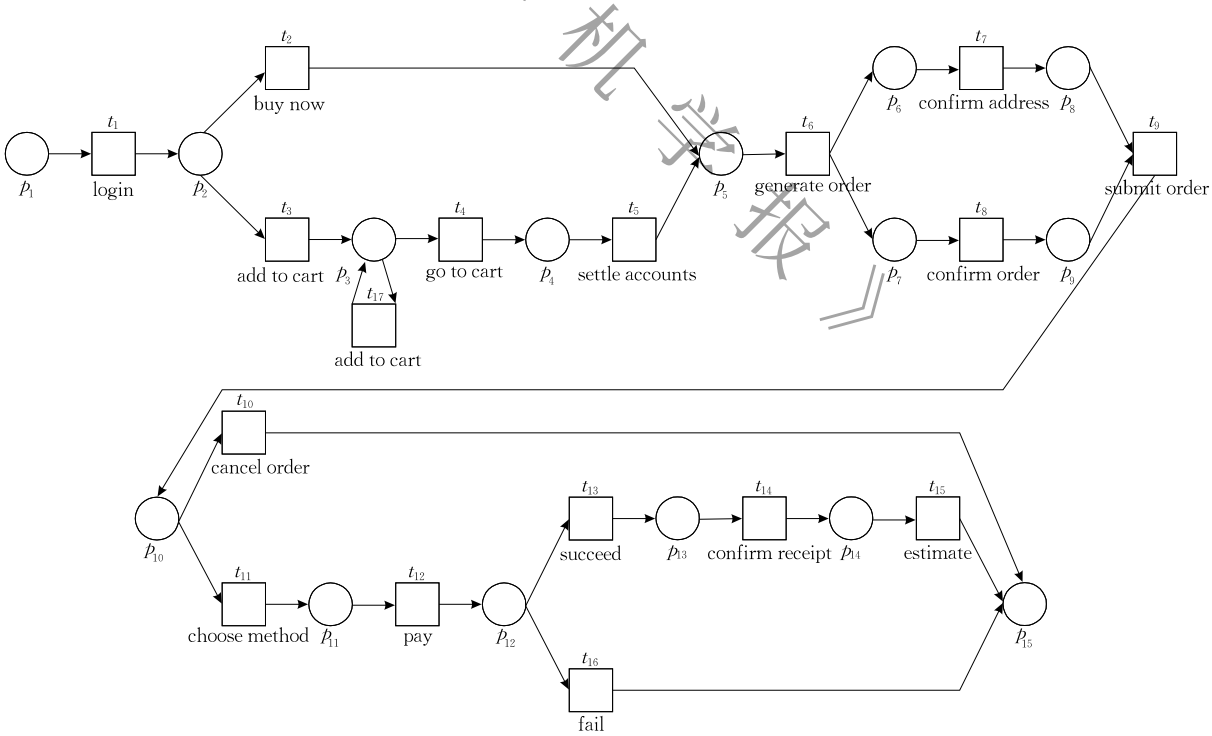
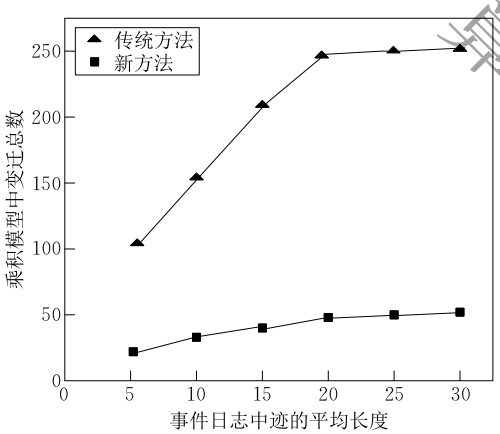


图 10 网上购物流程的 Petri 网模型 N_{os}

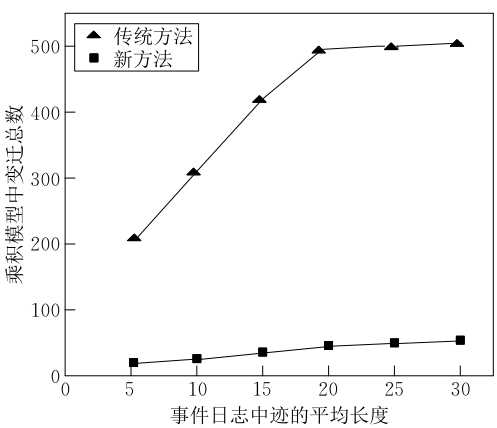
由该模型随机生成事件日志与已知过程模型进行分析. 根据过程模型生成完全拟合且长度不相同的迹，每条迹大约包含 6 到 30 个活动，并通过随机删除和增加活动在迹中制造噪音. 然后，根据标准似然代价函数计算所有迹与模型的一个最优校准. 在计算过程中统计乘积模型中变迁个数、变迁系统中结点个数、生成变迁系统所需时间、查找最优校准花费时间等信息，以比较传统方法和 AoPm 方法在处

理批量迹和过程模型之间的校准时的优劣. 上述需要考察的项目中，从一定程度上，乘积模型中变迁的个数影响了变迁系统中结点个数，二者反映了占用的存储空间. 查找最优校准所需时间从实验中记录的数据可看出 A+算法和执行 m 次 A* 算法所需时间相近，不再讨论. 两种方法生成变迁系统所需时间的不同就体现了二者在执行时间上的差异. 因此，实验重点研究两种方法分别在乘积模型中变迁总数及生成变迁系统所需时间两个方面的对比.

所有实验均采用标准似然代价函数对校准中出现的偏差进行度量，假设事件日志中出现噪音的比例平均为 25%. 每次实验的结果数据都是相同实验做 10 次的平均性能. 每次随机产生的事件日志中包含 m 条迹. 实现 m 条迹与过程模型之间的校准，AoPm 方法只需执行一次，而传统方法需执行 m 次. 为了查看整体比较效果，传统方法考察 m 条迹最终累加结果. 实验统计结果分别如图 11 与图 12 所示.



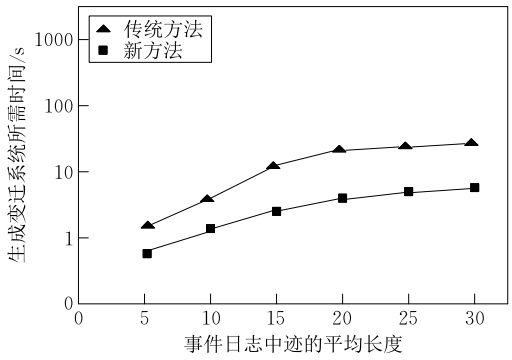
(a) m=5



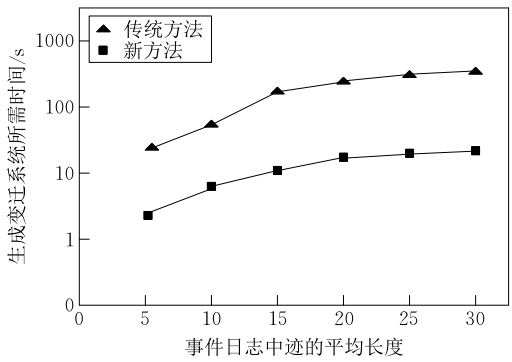
(b) m=10

图 11　乘积模型中变迁总数比较

图 11 中两个子图均显示随着事件日志中迹平均长度的增加，所建立乘积模型的变迁总数会不断增加. 该结论与 5.1 节的分析结果是一致的. 当迹的长度增加时，说明迹中包含的活动增多，根据迹建立的日志模型中变迁数就会增加，因此乘积模型的变迁总数也会增加. 另外，不论事件日志所包含迹的数目为多少，使用 AoPm 算法，只需建立一个日志模型，相应的只需生成一个乘积系统；而使用传统校准方法，所建立日志模型与所生成乘积系统的数目和迹的数目是相同的，即当事件日志中迹的数目 m=5 时，需建立 5 个乘积系统，而当 m=10 时，需建立 10 个乘积系统. 因此，传统校准方法建立乘积系统中包含的变迁总数是 AoPm 方法的 m 倍.



(a) m=5



(b) m=10

图 12　生成变迁系统所需时间比较

图 10 所示网上购物流程 Petri 网模型 N_{os} 共包含 17 个变迁，对应着 17 个活动. 假设事件日志出现噪音的比例平均为 25%左右，如此事件日志中出现过程模型中不存在的活动最多约 4 个. 图 11 可以看出，当迹的长度为 20 时，即使事件日志中迹平均长度继续增加，所得乘积系统的变迁数基本保持不变. 这是因为，事件日志中可能会出现的合理活动有 20 个左右，当迹平均长度达到此值时，即使迹长度再增加，迹中活动只会重复出现，而不会有新活动出现. 因此此时日志模型的变迁数稳定不变，同步变迁数保持不变，而乘积系统的变迁数也基本保持不变.

图 11 的纵坐标取值范围有较大不同，从图中可

以看出，事件日志中迹的数目 $m=10$ 时，传统校准方法所得乘积模型的变迁数是 $m=5$ 时乘积模型变迁数的 2 倍左右. 其原因为采用传统校准方法生成的乘积模型的个数与事件日志中迹的数目是一致的，而每个乘积模型包含的变迁数和给定迹相关. 但是，当事件日志中迹的数目发生变化时，AoPm 方法生成的乘积模型包含的变迁数基本保持不变，是因为出现在迹中的活动是一定的，当迹的数目增加到一定时，即使出现新迹，也几乎没有新活动出现. 此时，生成模型时，增加的只是变迁之间的库所和弧，而不会增加新变迁.

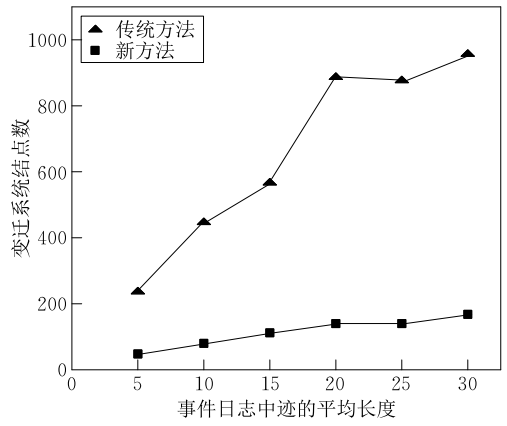
图 12 中，y 轴以指数级规模增长. 从图 12 中可以看出，传统校准方法生成变迁系统所花费的时间是 AoPm 方法的倍数且与事件日志中迹的数目有关系. 当 $m=5$ 时，传统校准方法生成变迁系统花费的时间是 AoPm 方法的 5 倍左右；当 $m=10$ 时，传统校准方法生成变迁系统花费的时间是 AoPm 方法的 10 倍左右. 其原因为若迹的平均长度相同，当事件日志中有 m 条迹时，传统校准方法要生成 m 个变迁系统，而 AoPm 方法只需生成 1 个变迁系统. 二者生成的每个变迁系统规模相近且花费的时间亦相似.

比较图 12 中两个子图可以发现，当事件日志中迹的数目增加时，无论是传统校准方法还是 AoPm 方法生成变迁系统所花费时间均有一定增加. 其原因为当迹的数目增加时，使用传统校准方法生成的变迁系统个数会相应增加；而 AoPm 方法虽然只生成一个乘积模型且模型的变迁数相近，但模型的库所和弧会有一定程度增加，因此生成的变迁系统会更为复杂，花费时间更多.

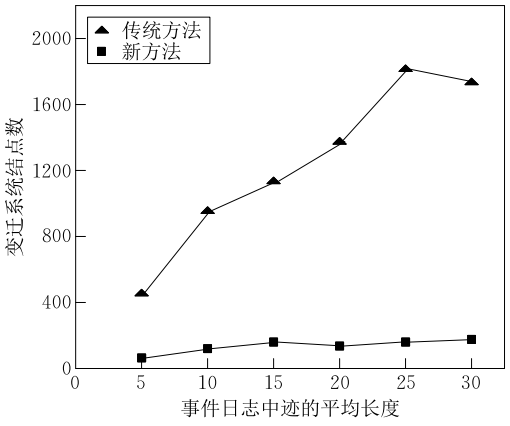
无论是 AoPm 方法还是传统校准方法占用内存空间主要是由变迁系统结点数造成的. 实验比较结果如图 13 所示.

A+算法和 A++算法均是在变迁系统上进行结点搜索，其空间复杂度主要是考察变迁系统的结点数，而变迁系统的结点数是随着乘积模型的变迁数的增加而递增的. 但是对比图 11 和图 13 可以看出变迁系统的结点数和乘积模型的变迁数之间既非线性关系也非指数关系. 这也符合 Petri 网的可达图与其变迁数之间的关系.

综上所述，计算事件日志中批量迹与过程模型之间的最优校准时，AoPm 方法无论在占用内存方面还是计算时间方面都比传统方法更好一些.



(a) $m=5$



(b) $m=10$

图 13 变迁系统结点数比较

6.2 实例分析

校准是观察行为和建模行为之间进行各种分析的起点. 而批量迹与过程模型之间的校准可以提高校准效率，包括计算校准花费的时间和占用的内存空间. 6.1 节中通过仿真实验详细地分析介绍了 AoPm 方法较传统校准方法的优越性. 本节将采用现实生活中的日志与模型作为研究案例来说明 AoPm 方法在处理更为复杂的实际案例时所具有的洞察力及健壮性. 本实验所采用日志与模型来自于某三甲医院，主要包括门诊系统业务和住院系统业务案例及流程. 为便于描述，门诊系统业务和住院系统业务在本实验中分别用 OPSB(Outpatient System Business)与 IPSB(Inpatient System Business)标记. 两个业务流程相关事件日志与模型的详细描述如表 2 所示.

表 2 真实案例相关数据

序号	事件日志			过程模型	
	名称	迹条数	活动数	变迁数	库所数
1	OPSB	1831	50541	46	43
2	IPSB	723	39356	70	72

根据给定真实日志和模型使用 A+ 算法和 A++ 算法分别计算日志中每条迹与过程模型之间的一个最优校准和所有最优校准，统计所需时间及对偏差的诊断情况，来例证该方法的可行性与适应性. 其实验统计结果如表 3 与表 4 所示.

表 3　真实案例所需时间

案例名称	A+算法	A++算法		
	所需时间/s	最优校准数/迹		所需时间/s
		最大	平均	
OPSB	<1	29	1.30	<1
IPSB	<1	54	1.91	<1

表 4　真实案例偏差统计

序号	OPSB 案例		IPSB 案例	
	迹条数	偏差数	迹条数	偏差数
1	241	9	63	13
2	173	8	39	31
3	56	12	24	10
4	49	17	9	3
5	21	—	12	—

表 3 记录使用案例 OPSB 和 IPSB 分别运行 A+ 算法和 A++ 算法所需花费时间. 从统计结果来看，其运行时间均小于 1 s. 说明算法在处理实际复杂问题时，能够在有限且较短时间内完成，因此算法的时间复杂度是可以接受的且效率较高. 另外，使用 A++ 计算事件日志中迹与过程模型之间的所有最优校准时，案例 OPSB 中每条迹的平均最优校准个数为 1.30，所有迹中最优校准个数最多为 29；案例 IPSB 中每条迹的平均最优校准个数为 1.91，所有迹中最优校准个数最多为 54.

统计数据可以看出，虽然真实案例中，个别迹与过程模型之间存在最优校准，但是大部分迹与过程模型之间的最优校准个数较少. 说明真实案例在实际运行 A++ 算法时，占用内存空间有限，即本算法的空间复杂度亦在可接受范围内. 该系列实验数据表明应用 A+ 算法和 A++ 算法计算复杂现实案例中批量迹与过程模型之间的最优校准时具有一定的健壮性.

表 4 记录案例 OPSB 和 IPSB 所有迹与过程模型之间最优校准包含偏差的情况. 其中偏差数为“—”表示非固定值，如 OPSB 案例中 21 条迹的偏差数为非固定值，即 21 条迹的最优校准中存在偏差，但是偏差数并非全部相同，且和上述各值不同. 其中，7 条迹偏差数为 18，2 条迹偏差数为 2，1 条迹偏差数为 21 等等. 该记录涵盖了偏差数的小概率情况. 不再详细列举.

根据似然标准代价函数定义，迹与过程模型之间的偏差即为最优校准包含的日志移动或模型移动. 本实验统计了每个案例其偏差数不为零且相同的迹条数. 表 4 中数据显示案例 OPSB 的 1831 条迹与过程模型的最优校准中，有 241 条迹存在 9 个偏差，173 条迹存在 8 个偏差，56 条迹存在 12 个偏差，49 条迹存在 17 个偏差，21 条迹的偏差非固定数，其他迹不存在偏差；显示案例 IPSB 的 723 条迹与过程模型的最优校准中，有 63 条迹存在 13 个偏差，39 条迹存在 31 个偏差，24 条迹存在 10 个偏差，9 条迹存在 3 个偏差，12 条迹的偏差非固定数，其他迹不存在偏差.

表 4 所示诊断结果显示案例 OPSB 中 241+173+56+49+21=540 条迹与过程模型之间存在偏差. 因此，事件日志中大约 30% 的迹不能完全在过程模型上重演. 也就意味着，如果使用完整日志进行分析，大约有 30% 的迹是不符合要求的，应该过滤掉. 案例 IPSB 中 63+39+24+9+12=147 条迹与过程模型之间存在偏差，表示大约 20% 的迹不能完全在过程模型上重演. 实验结果表明住院系统记录的事件日志比门诊系统更加符合过程模型，因此住院系统所记录数据差错率低，数据处理的吞吐率更高，有更高的严谨性.

从实验结果可看出，应用 AoPm 方法来处理现实生活中复杂问题模型与日志集时，其执行效率较高且占用内存空间有限，时间复杂度和空间复杂度均在可接受范围内，说明了该方法的可用性与健壮性.

7　结束语

传统计算事件日志中迹与过程模型之间校准的方法，每次只能计算出一条迹与过程模型之间的校准. 该方法主要思想为首先将该迹转化为变迁之间完全是顺序关系的日志模型，接着计算日志模型与过程模型的乘积，然后生成乘积模型的变迁系统，最后利用 A* 算法及 A* 算法的改进算法得到迹与模型之间的最优校准. 计算日志模型与过程模型的乘积系统及其变迁系统工作量比较大，且占用的存储空间较多. 如果要求多条迹与过程模型的最优校准，就要不断重复该过程.

因此，基于标签 Petri 网提出一种新的校准方法——AoPm 方法，可以同时实现事件日志中多条迹与过程模型之间的校准. 首先，将要与过程模型进

行校准的迹作为一个完备事件日志集,由该事件日志中所有迹通过基于区域的过程发现算法挖掘出日志模型.然后,得到日志模型和过程模型的乘积及其变迁系统.最后,给出 A+算法和 A++算法分别计算得到事件日志中每条迹和过程模型之间的一个最优校准和所有最优校准.当计算多条迹与过程模型之间的校准时,该方法只需生成一个乘积模型及相应的变迁系统,大大节省了相关的工作量及存储空间.

当变迁系统中结点增多时,A+算法和 A++算法作为最优校准查找算法,其所需内存和时间将会有较大的消耗.虽然通过现实生活案例已经例证了算法的有效性及可行性,但仍可做一些工作来提高最优校准的查找效率.A+算法和 A++算法的主要算法思想借鉴于 A* 算法,而 A* 算法在寻找最短路径问题领域是一种高效、先进的搜索算法.因此,A+算法和 A++算法无需继续优化.在进一步研究中,将考虑变迁系统的化简.对变迁系统进行剪枝,即将变迁系统中不在最优校准路径上的结点删除.处理后的变迁系统中所有从源结点到目标结点的路径都对应着一个最优校准.新生成的变迁系统不仅节省了占用的内存空间,甚至不必使用 A+算法或者 A++算法,很容易便可得到所需最优校准.

另外,该方法虽然同时实现了多条迹与过程模型之间的校准,但是每条迹与过程模型之间的校准仍是相对独立的.在接下来的研究中,将给出一种新方法,实现多条迹与过程模型的同步校准.并提出针对该校准方法的一致性检查标准.

致 谢 感谢为本文提出宝贵意见的审稿专家及责任编辑!

参 考 文 献

[1] Manyika J, Chui M, Brown B, et al. Big Data: The Next Frontier for Innovation, Competition and Productivity. USA: Mckinsey Global Institute, White Paper, 2011

[2] Li Guo-Jie. The scientific value of big data research. Communications of the CCF, 2012, 8(9): 8-15(in Chinese)
(李国杰. 大数据研究的科学价值. 中国计算机学会通讯, 2012, 8(9): 8-15)

[3] Wang Yuan-Zhuo, Jin Xiao-Long, Cheng Xue-Qi. Network big data: Present and future. Chinese Journal of Computers, 2013, 36(6): 1125-1138(in Chinese)
(王元卓, 靳小龙, 程学旗. 网络大数据:现状与展望. 计算机学报, 2013, 36(6): 1125-1138)

[4] Feng Deng-Guo, Zhang Min, Li Hao. Big data security and privacy protection. Chinese Journal of Computers, 2014, 37(1): 246-258(in Chinese)
(冯登国, 张敏, 李昊. 大数据安全与隐私保护. 计算机学报, 2014, 37(1): 246-258)

[5] van der Aalst W M P, Stahl C. Modeling Business Processes: A Petri Net Oriented Approach. Cambridge, USA: MIT Press, 2011

[6] van der Aalst W M P. Process Mining: Discovery, Conformance and Enhancement of Business Processes. Berlin, Germany: Springer-Verlag, 2011

[7] Hu Hesuan, Li Zhiwu, Wang Anrong. Mining of flexible manufacturing system using work event logs and petri nets// Proceedings of the Advanced Data Mining and Applications. Xi'an, China, 2006: 380-387

[8] van der Aalst W M P, Adriansyah A, Dongen B F. Replaying history on process models for conformance checking and performance analysis. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2012, 2(2): 182-192

[9] Adriansyah A. Aligning Observed and Modeled Behavior [Ph. D. dissertation]. Eindhoven University of Technology, Eindhoven, the Netherlands, 2014

[10] Tian Yin-Hua, Du Yu-Yue. A grouping algorithm of optimal alignments. Journal of Shandong University of Science and Technology (Natural Science), 2015, 34(1): 29-34(in Chinese)
(田银花, 杜玉越. 一种最优校准的分组算法. 山东科技大学学报(自然科学版), 2015, 34(1): 29-34)

[11] van der Aalst W M P, Weijters A J M M, Maruster L. Workflow mining: Discovering process models from event logs. IEEE Transactions on Knowledge and Data Engineering, 2004, 16(9): 1128-1142

[12] Weijters A J M M, Ribeiro J T S. Flexible heuristics miner (FHM)//Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining (CIDM). Paris, France, 2011: 310-317

[13] de Medeiros A K A, Weijters A J M M, van der Aalst W M P. Genetic process mining: An experimental evaluation. Data Mining and Knowledge Discovery, 2007, 14(2): 245-304

[14] van Dongen B F. Process Mining and Verification [Ph. D. dissertation]. Eindhoven, the Netherlands: Eindhoven University of Technology, 2007

[15] Zha Hai-Ping, Wang Jian-Min, Sun Jia-Guang. Incremental algorithm for process mining based on sliding window. Computer Integrated Manufacturing System, 2008, 14(1): 203-208(in Chinese)
(查海平, 王建民, 孙家广. 一种基于滑窗的增量式过程挖掘算法. 计算机集成制造系统, 2008, 14(1): 203-208)

[16] van Dongen B F, van der Aalst W M P. Multi-phase process mining: Building instance graphs//Proceedings of the International Conference on Conceptual Modeling. Berlin, Germany, 2004: 362-376

[17] van Dongen B F, Busi N, Pinna G M, van der Aalst W M P. An iterative algorithm for applying the theory of regions in process mining//Proceedings of the Workshop on Formal Approaches to Business Processes and Web Services (FABP-WS'07). Siedlce, Podlasie, 2007: 36-55

[18] Murata T. Petri nets: Properties, analysis and applications. Proceedings of the IEEE, 1989, 77(4): 541-580

[19] Lin Chuang, Tian Li-Qin, Wei Ya-Ya. Performance equivalent analysis of workflow systems. Journal of Software, 2002, 13(8): 1472-1480(in Chinese)
(林闯，田立勤，魏丫丫. 工作流系统模型的性能等价分析. 软件学报，2002，13(8)：1472-1480)

[20] Verbeek H M W, Buijs J C A M, van Dongen B F, van der Aalst W M P. ProM 6: The process mining toolkit// Proceedings of the BPM Demonstration Track 2010. Hoboken, USA, 2010: 34-39

TIAN Yin-Hua, born in 1982, Ph. D. candidate, lecturer. Her main research interests include process mining and Petri nets.

DU Yu-Yue, born in 1960, professor, Ph. D. supervisor. His main research interests include software engineering, formal modeling and Petri nets.

HAN Dong, born in 1982, Ph. D. candidate, lecturer. His main research interests include process mining and resource management.

LIU Wei, born in 1977, Ph. D., associate professor. His main research interests include Petri nets, workflows and web services.

Background

Big data processing is a hot topic for researchers and concerned by the governments all over the world. Along with the development of big data, BPM enters a new era. More and more event logs are recorded which need to be analyzed and dealt with, meanwhile the process models are mined by a series of process mining algorithms or constituted by hand. But usually there are many deviations between the event logs and the corresponding process models. So the conformance checking between the event logs and the process models must be carried on, which is a main task to improve the efficiency of BPM. Alignment is a prevalent method of conformance checking, which can fix the deviations accurately in a trace of the event log and the process model.

The existing alignment method was presented by Adriansyah Arya systematically. The alignment method can achieve the alignments between only one trace and the process model at a time. This method must be applied for the equivalent times if the alignments between many traces and the process model are required, which means a lot of repetitive jobs. To resolve such problems, a new alignment method named AoPm method was presented based on Petri net, which could obtain the optimal alignments between a batch of traces in the event log and the process model at the same time. AoPm method breaks through the thought of the alignment between only one trace and the process model once, realizes the alignments between a batch of traces in the event log and the process model for the first time, and improves the efficiency of conformance checking between traces in the event log and the process model.

The work is supported by the National Natural Science Foundation of China under Grant Nos. 61170078, 61472228, the Natural Science Foundation of Shandong Province under Grant No. ZR2014FM009, the Promotive Research Fund for Young and Middle-Aged Scientists of Shandong Province under Grant No. BS2015DX010, and the "Taishan Scholar" Construction Project of Shandong Province. The team has published a lot of papers on the study of logic Petri nets, service composition, BPM and process mining.

The projects mentioned above need to do some research on business process. As part of the projects, the achievement of this paper realized the conformance checking between the event log and the process model. Usually, the process model is a model based on Petri net, which must be sound and conforms to the definition of logical workflow. Conformance checking is very important and indispensable for business process mining.