计算机应用
Journal of Computer Applications
ISSN 1001-9081,CN 51-1307/TP

《计算机应用》网络首发论文

题目：基于感知成本的流程模型与事件日志有效对齐
作者：李多芹，方贤文，王丽丽，邵叱风
收稿日期：2021-08-02
网络首发日期：2021-12-13
引用格式：李多芹，方贤文，王丽丽，邵叱风．基于感知成本的流程模型与事件日志有效对齐[J/OL]．计算机应用．
https://kns.cnki.net/kcms/detail/51.1307.tp.20211206.1637.008.html

Journal of Computer Applications 计算机应用 ISSN 1001-9081 CODEN JYIIDU http://www.joca.cn

DOI:10.11772/j.issn.1001-9081.2021081378

基于感知成本的流程模型与事件日志有效对齐

李多芹[1]，方贤文[1*]，王丽丽[1,2]，邵叱风[3]

(1.安徽理工大学 数学与大数据学院，安徽 淮南 232001;

2.嵌入式系统与服务计算教育部重点实验室（同济大学），上海 200000； 3.安徽科技学院 信息与网络工程学院，安徽 蚌埠 233030)

(*通信作者电子邮箱：duoqin_li@126.com)

摘 要: 现存的成本函数没有考虑到业务流程中各活动在现实情境中有着不同的重要程度，于是在模型与日志的对齐过程中可能会导致对齐成本严重偏离感知成本。针对这一问题，基于业务流程中行为的典型流特征提出了重要同步成本函数的概念，并在该函数下给出一种能够提升效率的对齐方法。首先，基于感知成本的概念定义重要同步成本函数；接着，依据日志迹和流程模型中行为的典型流特征确定用以分割流程模型与日志迹的重要匹配子序列；最后，基于重要同步成本函数对齐分割后的子流程和对应的日志迹子序列，并将分段对齐的结果进行合并得到最终的对齐结果。实验部分从准确率和效率两方面进行验证：准确率方面，与现存的标准成本函数和最大同步成本函数相比，所提成本函数下的对齐准确率最高提升了 17.44 个百分点，且当事件日志包含混合噪声时，所提成本函数下的平均对齐准确率最高，为 88.67%；对齐效率方面则通过比较对齐所耗时间来验证，现存两种函数的平均耗时分别为 1.58s 和 2.21s，而所提方法为 0.63s，分别提升了约 150.79%和 250.79%。实验结果表明所提方法能在满足准确率的同时提升对齐的效率。

关键词: 标准成本函数；最大同步成本函数；典型流特征；感知成本；重要同步成本函数；有效对齐

中图分类号:TP391.9 文献标志码： **A**

Effective alignment of process model with event logs based on perceived cost

LI Duoqin[1], FANG Xianwen[1*], WANG Lili[1,2], SHAO Chifeng[3]

(1.School of Mathematics and Big Data, Anhui University of Science and Technology, Huainan Anhui 232001, China;

2.Key Laboratory of Embedded System and Service Computing of Ministry of Education, Tongji University, Shanghai 200000, China;

3.School of Information and Network Engineering, Anhui University of Science and Technology, Bengbu Anhui 233030, China)

Abstract: The different importance of the activities in the business process in real world is not taken into account by the existing cost function, so alignment of model and log may cause that alignment cost deviate significantly from perceived cost. To solve this problem, an important synchronization cost function is proposed based on the typical flow characteristic of the behavior in business processes, and an efficient alignment method is proposed under this function. First, important synchronization cost function is defined based on the concept of perceived cost. Then, important matching sub-sequence is determined according to the log trace and the typical flow characteristic of the behavior in the process model. Finally, based on the important synchronization cost function, segmented subprocess and corresponding log trace subsequence are aligned, and the segmented alignment results are combined to obtain the final alignment result.The experiment verifies the accuracy and efficiency. In terms of accuracy, compare with existing standard cost function and maximum synchronous cost function, the proposed cost function improves the alignment accuracy by up to 17.44 percentage points and when the event log contains mixed noise, the proposed cost function has the highest average alignment accuracy of 88.67%. The efficiency of alignment is verified by comparing the time consumed by alignment. The average time of the existing two functions is 1.58s and 2.21s respectively, while the proposed method is 0.63s, which improves by about 150.79% and 250.79% respectively.Experimental results show that the proposed method can satisfy the accuracy and improve the efficiency of alignment.

Keywords: standard cost function**;** maximum synchronization cost function; typical flow characteristic; perceived cost; important synchronization cost function; effective alignment

收稿日期: 2021-08-02; **修回日期**: 20121-11-22; **录用日期**: 2021-11-25。

基金项目:国家自然科学基金资助项目(61402011,61572035);安徽省自然科学基金资助项目(1508085MF111,1608085QF149);安徽理工大学研究生创新基金资助项目（2019CX2068）。

作者简介: 李多芹(1996—)，女，安徽淮南人，硕士研究生，主要研究方向：Petri 网、过程挖掘； 方贤文（1975—），男，河南信阳人，教授，博士，CCF 会员，主要研究方向：Petri 网、可信软件、业务流程变化域分析； 王丽丽（1982—），女，安徽安庆人，副教授，博士研究生，主要研究方向：Petri 网、业务流程挖掘； 邵叱风（1995—）男，安徽合肥人，硕士，CCF 会员，主要研究方向：Petri 网，过程挖掘、模型修复及模型优化。

0 引言

近年来，事件日志与流程模型之间的对齐已被证明对过程挖掘中的服从性校验问题极为有用[1]。日志迹与流程模型之间可能会存在多种对齐，目前最优的对齐默认为给定成本函数下成本最小的对齐，于是在对齐过程中，成本函数的选取尤为重要。关于对齐及对齐成本函数的选取，很多学者已经做了大量研究。其中，文献[2]表明事件日志与流程模型之间保持适当对齐的重要性，并详细阐述了这种对齐的实现及对齐在服从性校验和性能分析中的应用。文献[3]利用流程模型结构和行为特征的优势，提出一种在寻找最优对齐时能显著缩减搜索空间的方法。文献[4]将对齐的计算问题转化为整数线性规划模型的求解问题，从而显著减小问题的复杂性。文献[5]提出了一种基于 Petri 网的事件日志与过程模型之间的快速对齐方法，即 Rapid Align 方法，以提高过程挖掘中计算最优对齐的效率。文献[6]中为了提升了过程挖掘中计算最优对齐的效率，提出一种基于 Petri 网可达图的业务对齐方法。

在上述工作中，都选择使用标准成本函数进行对齐。事实上，据所知，现有的绝大多数对齐问题中都默认使用该函数，该函数将模型和日志的同步移动及静默移动的成本记为0，异步移动的成本都记为 1，即将模型中的异步移动和日志中的异步移动产生的影响视为是等价的。然而，文献[7]中通过具体的实例表明，在某些应用情境中使用标准成本函数对齐模型和日志会产生不符合逻辑的对齐结果，于是文献[7]提出了标准成本函数的一个变体：最大同步成本函数，该函数中模型移动的成本值远小于日志移动，进而通过产生额外的模型移动使得同步移动的数量最大化。然而，不论是将异步移动的成本值都记为 1 还是使得模型移动的成本值远小于日志移动的成本值都可能会导致对齐成本与感知成本相差甚远。一个典型的情况是当出现冗余噪声时，最大化同步成本函数会尽可能牺牲模型移动来将该噪声事件与模型中的某一活动形成同步移动，从而导致对齐成本严重偏离感知成本。

基于此，本文根据业务流程中的典型行为流特征定义了一种新的成本函数，即重要同步成本函数，使得在该函数下对齐流程模型与事件日志时的对齐成本更加贴合感知成本。同时，基于该函数给出一种能够提升对齐效率的对齐算法。其中，业务流程中的典型行为流特征用于确定流程中各活动的重要程度。关于业务流程中的典型行为流，已有学者做了相关研究，其中文献[8]中指出许多流程通常以固定的顺序执行初始活动，经过一系列流程走向变化如，并行分支、循环等，又收敛到更结构化的行为，然后再次发散。文献[9]利用行为流的这种特征将流程模型划分为强制性子序列和选择性子序列，通过选择相应的测量公式，切实有效的提升了日志与模型之间的适合度。关于感知成本，本文迁移了文献[10]中提到的感知一致性概念，该文献研究的问题是：流程模型间行为一致性的哪一种正式概念可以最好地近似模型专家的感知一致性。类似地，本文的研究问题是：对齐流程模型和事件日志时，哪一种成本函数下的对齐成本能更好的接近感知成本。

1 知识准备

定义 1[7] 模型与日志的对齐。令 Σ 为活动集，包含静默活动 τ 的活动集记为 Σ_τ，即 $\Sigma_\tau=\Sigma\cup\{\tau\}$；包含跳过符号>>的活动集记为 $\Sigma_{>>}$，即 $\Sigma_{>>}=\Sigma\cup\{>>\};\Sigma_{\tau>>}=\Sigma\cup\{\tau,>>\}$则表示同时包含静默活动和跳过符号的活动集。令 $\sigma\in\Sigma^*$是一条日志迹，N 为一 Petri 网模型，则模型 N 中的执行序列与日志迹 σ 的对齐与对齐 γ 指的是日志-模型对的一个序列，即 $\gamma\in(\Sigma_{>>}\times\Sigma_{>>})^*$。

定义 2[7] 对齐成本。令 $\gamma\in(\Sigma_{>>}\times\Sigma_{>>})^*$是 $\sigma\in\Sigma^*$和 Petri 网 N 间的一组对齐。该对齐的成本函数 c 为对齐对到正实数上的映射，即 $c:(\Sigma_{>>}\times\Sigma\tau_{>>})\rightarrow R_{\geq 0}$，于是 $c(\gamma)=\Sigma_{0\leq i\leq|\gamma|-1}\ c(\gamma_i)$。

若给定对齐成本函数 c，在该成本函数下对齐 γ 为最优的，当且仅当不存在 γ'使得 $c(\gamma')<c(\gamma)$。

定义 3[7] 标准成本函数。对齐对$(l,m)\in(\Sigma_{>>}\times\Sigma\tau_{>>})$的标准成本函数 c_{st} 定义如下：

$$c_{st}(l,m)=\begin{cases}0, & l=>>\ \wedge\ m=\tau\\ 0, & l\in\Sigma\ \wedge\ m\in\Sigma\wedge\ l=m\\ 1, & l\in\Sigma\ \wedge\ m=>>\\ 1, & l=>>\ \wedge\ m\in\Sigma\end{cases}$$

由上式可以看出，对标准的成本函数而言，静默移动和同步移动的成本为 0；日志移动和模型移动的成本都为 1。文献[7]通过真实的案例表明，该成本函数虽然在相关文献中经常使用，但可能会产生不希望得到的，即不符合逻辑的对齐结果，于是提出了最大同步成本函数的概念如下：

定义 4[7] 最大同步成本函数。对齐对$(l,m)\in(\Sigma_{>>}\times\Sigma\tau_{>>})$的最大同步成本函数 $c_{max\text{-}sync}$ 定义如下:

$$c_{max-sync}(l,m)=\begin{cases}0, & l=>>\ \wedge\ m=\tau\\ 0, & l\in\Sigma\ \wedge\ m\in\Sigma\wedge\ l=m\\ 1, & l\in\Sigma\ \wedge\ m=>>\\ \varepsilon, & l=>>\ \wedge\ m\in\Sigma\end{cases}$$

其中 $0<\varepsilon<1$。比较定义 3 和定义 4 可以看出，两个成本函数只在模型移动成本值的设置上有所不同，定义 4 中将模型移动的成本值设置为任意小的数，即 ε，通过这种方式使得对齐结果中同步移动的数量最大化。可以预见，同步移动

数量最大化的同时，会产生很多额外的模型移动。然而，在某些真实情境中，额外产生的模型移动会带来更大的影响。以牺牲模型移动为代价一味的追求同步移动数量的最大化并不是明智的选择，从而使得产生的对齐结果并不可取。

2 动机例子

为了激励消费者，电子商城经常会发放一定额度优惠券，本文以此背景下的商品购买流程作为动机案例，如图 1 用 Petri 网给出了该购买流程的控制流依赖。

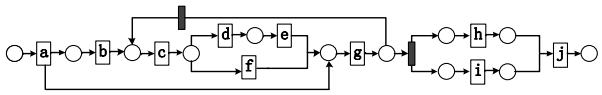


图 1 商品购买流程 *P*

Fig.1 Commodity purchase process

图 1 中各字母所指代的活动如表 1 所示，由图 1 和表 1 可知，顾客一旦选择某一商品后，随时都可以进入付款界面，且进入付款界面前或进入付款界面后都有查看店铺优惠券及选择使用或不使用优惠券的权利，这与当今各大电子商城的购物模式相吻合。

表 1 图 1 中各个字母指代的活动

Tab.1 Activities referred to by each letter in Figure 1

标识符	活动名称
a	选择商品
b	查看店铺优惠
c	领取优惠券
d	选择使用优惠券
e	选定优惠券
f	选择不使用优惠券
g	进入付款界面
h	填写收货地址
i	选择配送方式
j	付款

对迹 σ=<a,b,g,c,f,h,i,j>，若使用标准的成本函数将其与图 1 中的流程 P 进行对齐，可得到图 2 中的最优对齐 γ^1，易知在标准成本函数下 γ^1 的成本为 2；若使用最大同步成本函数则可得到最优对齐 γ^2 且在最大同步成本函数下 γ^2 的成本为 3ε。然而，代入图 1 中所给出的真实问题情境，可以明显感知到 γ^1 和 γ^2 的感知成本与两种函数下计算出的成本有很大差距，因为在网络购物这一问题情境中，顾客是否愿意进入到付款界面代表着对该商品购买意愿，是商家十分关注的步骤。尤其在 γ^2 中，跳过活动 g(进入付款界面)的成本用一个任意小的 ε 来衡量是不合理的。此外，把跳过活动 g 和跳过活动 f(或 c)的成本设置为相同的单位值也会使得对齐成本偏离感知成本。

γ^1=	a	b	g	c	f	>>	h	i	j

	a	b	>>	c	f	g	h	i	j

γ^2=	a	b	>>	>>	g	c	f	>>	h	i	j
	a	b	c	f	g	c	f	g	h	i	j

图 2 迹 σ 分别在标准成本函数(γ^1)和最大同步成本函数(γ^2)下的最优对齐

Fig.2 Optimal alignment of trace σ under standard cost function (γ^1) and maximum synchronous cost function (γ^2), respectively

知道，对齐模型和日志迹的意义在于反映模型流程和真实执行流程间的偏差。于是对于这类真实的问题情景，γ^1 和 γ^2 产生的对齐结果是不可取的，因为对齐成本严重偏离感知成本，也就是说对齐结果无法反映真实偏差情况。为了解决这个问题，下面将基于感知成本的概念提出标准成本函数的另外一个变体，即重要同步成本函数。

3 基于感知成本的有效对齐方法

3.1 重要同步成本函数

通过对动机例子的分析知道，跳过或插入不同活动的感知成本是有很大差距的。于是，为了使对齐成本更加贴合感知成本，在对齐前将活动集 Σ 按重要程度进行划分，并将划分后的不同的活动类分配不同的成本是必要的。为了划分活动集 Σ，不失一般性，引入流程模型中行为的典型流特征作为重要程度的划分依据。

文献[8]中指出许多流程以独特的或类似的行为开始，也就是说，通常一个初始活动(组)是以固定的顺序执行的。随后，根据流程的具体实例，流程中可能出现更多的变化，例如并行分支、循环等。在流程的某些节点上，行为再次收敛为更结构化的，即以固定的顺序执行的行为，然后再次发散。图 3 给出了该流程事实的示意图。

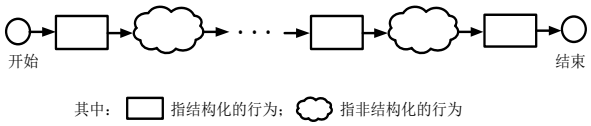


图 3 流程中行为的典型流示意图

Fig.3 A typical flow diagram of behavior in a process

此外，文献[9]所提方法同样基于该流程事实，将流程模型划分为强制性活动路径和选择性活动路径，并通过实验证明划分后的准确检测偏差能够明显提高适合度测量值。

基于上述事实，本文将流程模型中强制性活动路径上的所有活动记作强制性活动或重要活动，所有强制性活动组成的集合记为 Σ^+；将选择性活动路径中的所有活动记作选择性活动，所有选择性活动组成的集合记为 Σ^-，并默认活动集中的活动是按照流程执行的先后顺序排列的。这里

$\Sigma=\Sigma^{+}\cup\Sigma^{-}$且 $\Sigma^{+}\cap\Sigma^{-}=\emptyset$，即将活动集 Σ 按重要程度划分为两个互不相交子活动集 Σ^{+}和 Σ^{-}的并。具体地，对图 1 中的流程有：Σ={a,b,c,d,e,f,g,h,i,j};Σ^{+}={a,g,j};Σ^{-}={b,c,d,e,f,h,i}。基于上述活动集的分类，可将重要同步成本函数形式化定义如下：

定义 5 重要同步成本函数。结合第一节中对相关符号的定义，定义一个对齐对的重要同步成本函数 $c_{\text{imp-sync}}$ 如下：

$$c_{\text{imp-sync}}(l,m)=\begin{cases}0, & l=>>\ \wedge\ m=\tau\\ 0, & l\in\Sigma\ \wedge\ m\in\Sigma\wedge\ l=m\\ 1, & l\in\Sigma^{+}\ \wedge\ m=>>\\ \varepsilon, & l\in\Sigma^{-}\ \wedge\ m=>>\\ 1, & l=>>\ \wedge\ m\in\Sigma^{+}\\ \varepsilon, & l=>>\ \wedge\ m\in\Sigma^{-}\end{cases}$$

其中 0<ε<1。定义 5 中活动集被分 Σ 为强制性活动集 Σ^{+}和选择性活动集 Σ^{-}，且 Σ^{+}上日志移动和模型移动的成本为 1;Σ^{-}上日志移动和模型移动的成本都为 ε。通过为两类活动集赋予差距较大的单位成本值，使得在对齐过程中优先考虑重要活动的对齐，从而得到更加贴近感知成本的对齐结果。

根据定义 5 中的成本函数，可以得到迹 σ=<a,b,g,c,f,h,i,j>和图 1 中所示流程的最优对齐如下：

γ^{3}=

a	b	g	c	f	h	i	j
a	>>	g	>>	>>	h	i	j

图 4 迹 σ 在重要同步成本函数下的最优对齐 γ^{3}

Fig.4 The optimal alignment of trace σ γ^{3} under important synchronization cost functions

易知 γ^{3} 在重要同步成本函数下的对齐成本为 3ε，虽然 γ^{2} 的对齐成本也为 3ε，但正如 2.1 中所述，γ^{2} 中为了使同步移动的数量最大化添加了额外的模型移动(如活动 g)，这导致对齐成本与感知成本有很大差距。相反地，γ^{3} 中的对齐因为优先考虑了重要活动的同步(已用阴影突出显示)，所以使得对齐后的成本与感知成本比较贴近。于是对迹 σ=<a,b,g,c,f,h,i,j>和图 1 中的流程来说，重要同步成本函数下的最优对齐 γ^{3} 更能反映流程模型和真实执行迹间的偏差。

值得注意的是，上述活动类的划分只是依据流程模型中行为的典型流特征分为两类，在实际执行中，操作者可根据现实意义自定义的将流程中活动划分为若干类。此外，不同活动类上单位成本值的分配也可以依据实际需求自定义设置。当按照上述操作划分活动类时，下面将基于该成本函数给出一种可以提升效率的对齐计算方法。

3.2 重要同步成本函数下的有效对齐方法

基于所提成本函数有效对齐方法的框架图如图 5 所示，主要可以被分为 3 步：首先，根据流程模型的典型流特征划分活动类，并基于日志迹和强制性活动集确定重要匹配子序列；然后，依据重要匹配子序列中的活动分别将模型与日志做对应的分割；最后，基于重要同步成本函数对齐每一个子流程和对应的日志迹子序列，并将分段对齐的结果进行合并以得到最终的对齐结果。

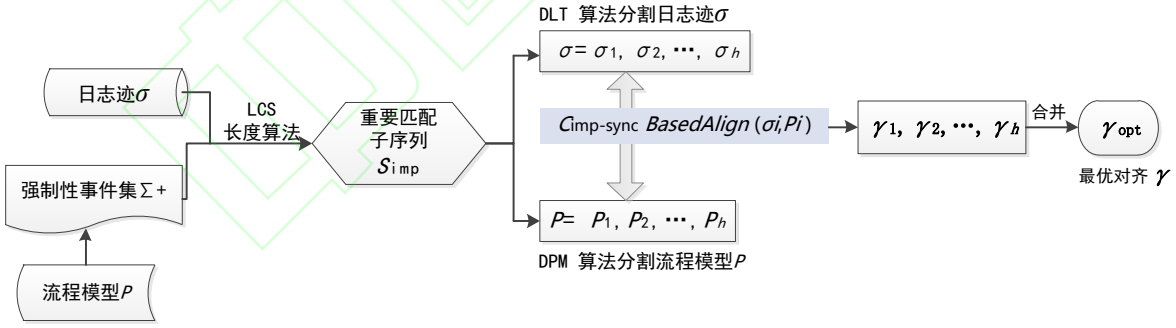


图 5 本文方法框架图

Fig.5 Frame diagram of the proposed method

3.3 重要匹配子序列的确定

重要匹配子序列指的是在日志迹和强制性活动集中都按一定顺序出现的活动组成的集合，子序列中的活动在对齐过程中会优先产出同步移动。依据流程模型中行为的典型流特征划分活动类后，利用文献[11]中给出的最长公共子序列概念从日志迹和 Σ^{+}中确定重要匹配子序列。最长公共子序列的定义如下：

定义 6[11] 最长公共子序列。序列 $Z_r=(z_1,z_2,\ldots,z_r)$为序列 $X_m=(x_1,x_2,\ldots,x_m)$和序列 $Y_n=(y_1,y_2,\ldots,y_n)$的最长公共子序列，当且仅当序列满足：

1) $z_s=x_{i_s}=y_{j_s},1\le s\le r$；

2) $1 \le i_1 \le i_2 \le \cdots \le i_r \le m \wedge 1 \le j_1 \le j_2 \le \cdots \le j_r \le n$；

3)对于任意一个 X_m 和 Y_n 的子序列 Z'，都有$|Z'| \le |Z_r|$。

文献[11]根据最长公共子序列定义，展示了求最长公共子序列长度的算法(Longest Common Subsequence, LCS)。该算法以两个有限序列为输入，根据递推计算公式输出这两个序列的最长公共子序列定义。以动机案例中所给的日志迹和模型为例，输入 σ=<a,b,g,c,f,h,i,j>和 Σ^+={a,g,j}，可得到的 σ 和 Σ^+的最长公共子序列，也就是日志迹 σ 和图 1 中模型的重要匹配子序列为 S_{imp}=(a,g,j)。为表示方便，可将该算法视为一个函数，即 $LCS(\sigma,\Sigma^+)=S_{imp}$。确定了重要匹配子序列 S_{imp} 后，下面将根据 S_{imp} 中的活动分别对日志迹和流程模型进行分割。

3.4 日志迹和流程模型的分割

在 3.3 中，通过 *LCS* 函数可以确定对齐流程模型和日志迹的重要匹配子序列 S_{imp}，为了提升对齐流程模型和事件日志的效率，下面将依据 S_{imp} 中的活动分别对流程模型和事件日志进行分割，使得分割后的子模型和日志迹子序列能够一一对应并对齐，利用这种分而治之的思想旨在缩减对齐整条日志迹和流程模型时需要搜索的状态空间。下面给出以日志迹和 S_{imp} 为输入的日志分割算法(Log Trace Devide,LTD)。该算法通过 If-else 语句区分出四种不同的日志分割情况。

算法 **1** 日志迹分割算法。

输入 σ=<$e_1,e_2,\ldots,e_n$>; S_{imp}=($e_{i1},e_{i2},\ldots,e_{ik}$);

输出 $\sigma_1,\sigma_2,\ldots,\sigma_h$;$h$=$k$-1 or k or k+1。

1. Initialize $\sigma_1,\sigma_2,\ldots,\sigma_h \leftarrow \emptyset$
2. If $e_1=e_{i1} \wedge e_n=e_{ik}$:
3. $h=k-1$
4. $\exists e_{j2},e_{j3},\ldots,e_{j(k-1)} \in \sigma$ s.t. $e_{i2}=e_{j2},\ldots,e_{i(k-1)}=e_{j(k-1)}$
5. <$e_{i1},e_2,\ldots,e_{i2}$>$\leftarrow\sigma_1$;<$e_{i2},\ldots,e_{i3}$>$\leftarrow\sigma_2$;…;<$e_{i(k-1)},\ldots,e_{n-1},e_{ik}$>$\leftarrow\sigma_{k-1}$
6. else If $e_1=e_{i1} \wedge e_n \ne e_{ik}$:
7. $h=k$
8. $\exists e_{j2},e_{j3},\ldots,e_{jk} \in \sigma$ s.t. $e_{i2}=e_{j2},\ldots,e_{ik}=e_{jk}$
9. <$e_{i1},e_2,\ldots,e_{i2}$>$\leftarrow\sigma_1$;<$e_{i2},\ldots,e_{i3}$>$\leftarrow\sigma_2$;…;<$e_{i(k-1)},\ldots,e_{ik}$>$\leftarrow\sigma_{k-1}$;<$e_{ik},\ldots,e_{n-1},e_n$>$\leftarrow\sigma_k$
10. elseIf $e_1 \ne e_{i1} \wedge e_n=e_{ik}$:
11. $h=k$
12. $\exists e_{j1},\ldots,e_{j(k-1)} \in \sigma$ s.t. $e_{i1}=e_{j1},\ldots,e_{i(k-1)}=e_{j(k-1)}$
13. <$e_1,e_2,\ldots,e_{i1}$>$\leftarrow\sigma_1$;<$e_{i1},\ldots,e_{i2}$>$\leftarrow\sigma_2$;…;<$e_{i(k-1)},\ldots,e_{n-1},e_{ik}$>$\leftarrow\sigma_k$
14. else:
15. $h=k+1$
16. $\exists e_{j1},\ldots,e_{jk} \in \sigma$ s.t. $e_{i1}=e_{j1},\ldots,e_{ik}=e_{jk}$
17. <$e_1,e_2,\ldots,e_{i1}$>$\leftarrow\sigma_1$;<$e_{i1},\ldots,e_{i2}$>$\leftarrow\sigma_2$;…;<$e_{i(k-1)},\ldots,e_{ik}$>$\leftarrow\sigma_k$;<$e_{ik},\ldots,e_{n-1},e_n$>$\leftarrow\sigma_{k+1}$
18. 返回 $\sigma_1,\sigma_2,\ldots,\sigma_h$;$h$=$k$-1 or k or k+1

从算法 1 中可以看出，日志迹的分割可以依据日志迹与 S_{imp} 的 1 号位活动、n 号位活动是否相同分为 4 种情况，进而导出 3 种可能的子模型/日志子序列的个数，即 h=k-1 or k or k+1。此外，可以观察到，每两个相邻的子序列会有一个共享的活动，即上一个子序列的结束活动与下一个子序列的开始活动相同。算法最后输出日志迹 σ 的 h 个子序列(算法 1 中：18)。为表示方便，同样将该算法视为一个函数，即 $LTD(\sigma,S_{imp})=\sigma_1,\sigma_2,\ldots,\sigma_h$。延续 3.2 中的例子可对日志迹 σ=<a,b,g,c,f,h,i,j>进行分割：因为 S_{imp}=(a,g,j),$|S_{imp}|$=3 且日志迹与 S_{imp} 的 1 号位活动、n 号位活动均相同，于是可根据算法 1 的 4-5 将日志迹 σ=<a,b,g,c,f,h,i,j>分成 2 个子序列：σ_1=<a,b,g>,σ_2=<g,c,f,h,i,j>，可以看到活动 g 为 σ_1 和 σ_2 的共享活动。

类似的，可以依据 S_{imp} 中的活动将流程模型分成若干个个子流程，即流程分割算法(Process Model Devide,PMD)。为表示方便，同样将该算法视为一个函数，于是有 $PMD(P,S_{imp})=P_1,P_2,\ldots,P_h$。此时，每两个相邻的子流程也会有一个共享的活动，即上一个子流程的结束活动与下一个子流程的开始活动相同。根据重要匹配子序列 S_{imp} 可将图 1 中的流程分割为两个子流程 P_1 和 P_2，如图 6 所示。

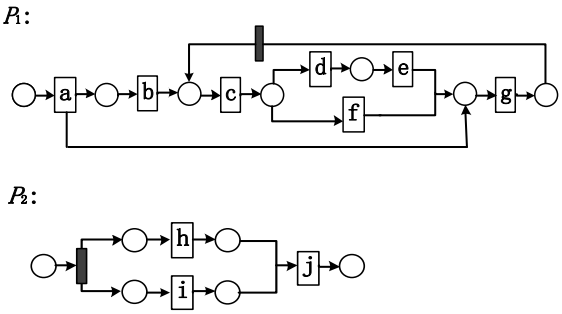


图 6 图 1 中流程模型根据 S_{imp} 分割后的子流程

Fig.6 The sub-process after the segmentation of the process model in Figure 1 based on S_{imp}

注意到，根据同一个重要匹配子序列对流程模型和日志迹进行分割后，得到的分割结果是对应的，即子流程与子序列个数相同，且存在一一对应的关系。于是，可先对齐分割

后得到的子流程与子日志序列，再依据共享活动将各部分的对齐结果合并以得到完整的日志迹和流程模型的对齐。较之于传统的对齐操作，这种分而治之的对齐策略能大幅提升对齐效率。

通过 *LTD* 和 *PMD* 函数可以将流程模型和事件日志分割成一一对应的子流程和日志迹子序列，接着基于重要同步成本函数对齐每一个子流程和对应的日志迹子序列，将分段对齐的结果进行合并就可以得到最终的对齐结果。

3.5 基于重要同步成本函数的有效对齐算法

到目前为止，前面的内容分别详细阐述了所提方法的理论基础，以及重要步骤的具体操作，下面将给出完整的算法实现。该算法以日志迹 σ 和流程模型 P 为输入，经过确定重要匹配子序列 S_{imp}、基于 S_{imp} 分割流程模型与事件日志、在重要同步成本函数下对齐子流程和日志迹子序列及依据共享活动合并对齐结果等步骤，最后输出重要成本函数下日志迹与流程模型的最优对齐 γ_{opt}。

算法 **2** 基于重要同步成本函数 $c_{\text{imp-sync}}$ 的对齐算法。

输入 $\sigma=<e_1,e_2,\ldots,e_n>$; $P=(S,T;F,M_0)$;

输出 γ_{opt}。

1. $\Sigma^{+}\leftarrow GetMandatoryEvent(P);\Sigma^{-}\leftarrow GetSelectiveEvent(P)$
2. $S_{\text{imp}}\leftarrow LCS(\sigma,\Sigma^{+})$
3. $\sigma_1,\sigma_2,\ldots,\sigma_h\leftarrow LTD(\sigma,S_{\text{imp}})$
4. $P_1,P_2,\ldots,P_h\leftarrow PMD(P,S_{\text{imp}})$
5. $Es=GetShareEvent(\sigma_1,\sigma_2,\ldots,\sigma_h)$
6. For σ_i,P_i $i\le h$:
7. $\gamma_{i,\text{opt}}\leftarrow c_{\text{imp-sync}}BasedAlign(\sigma_i,P_i)$
8. $\gamma_{\text{opt}}\leftarrow \text{Join}(\gamma_{1,\text{opt}},\gamma_{2,\text{opt}},\ldots,\gamma_{h,\text{opt}},Es)$
9. 返回 γ_{opt}

算法首先获取强制性活动集和选择性活动集(算法 1 中：1)，接着利用 LCS 长度算法获取重要匹配子序列 S_{imp}(算法 1 中：2)，然后分别通过 *LTD* 和 *PMD* 函数获取日志迹子序列和子流程(算法 1 中：3-4)，同时根据分割后的日志迹提取共享活动集 Es(算法 1 中：5)。对每一组对应的子序列和子流程，在 $c_{\text{imp-sync}}$ 下进行对齐(算法 1 中：6-7)，最后利用共享活动将各部分的对齐结果进行合并，并返回最优对齐 γ_{opt}。

4 实验评估

为了验证上述所提方法，实验部分将先介绍实验数据的来源；接着，详细阐述实验步骤；最后，分别通过三种函数下对齐准确率和效率的对比来评估所提方法的性能，并在实验结果对比图的支撑下得出结论，即本文方法能在满足准确率的同时提升对齐的效率。

4.1 实验数据

JOUCK T 等人提出的 PTandLogGenerator[12-13]可构造同时包含序列、并发、选择和循环结构的业务流程模型，该构造器已在 Prom 中实现，操作者通过输入模型大小、各个操作符出现的概率等参数可直接批量生成符合要求的流程模型，部分学者将其用于过程发现算法的评估[14-15]。为了验证所提方法，实验部分将利用该构造器随机生成 10 条业务流程模型，流程中活动数量为 11 到 37 的均匀分布。通过对每条业务流程进行下述实验步骤中一系列的处理，最终将对 6000 条对齐结果的数据进行分析。所有的实验数据均可在线访问：https://github.com/duoqinLi/experimenta-data.git。

4.2 实验步骤

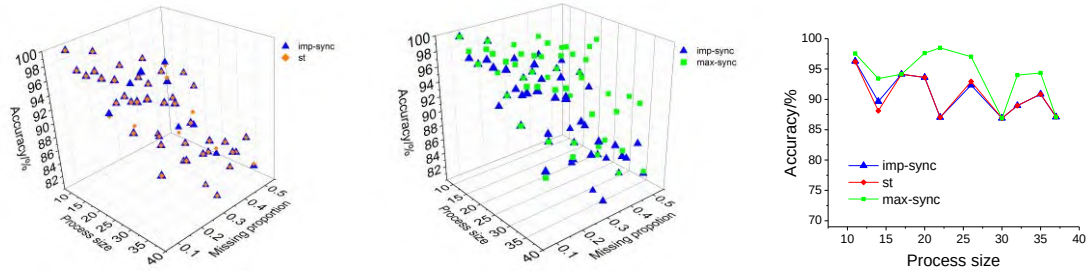
为了评估三种成本函数下的对齐准确率和对齐效率，设计如下实验步骤：首先，对每一个用 Petri 网表示的流程模型，通过 Prom 中的插件随机运行出 10 条完全服从的流程执行序列，称这些服从的日志迹为参考迹，用 σ 表示。然后，为了获取包含各种可能情况的不服从日志，对 σ 分别添加不同形式且不同比例的噪声。噪声的形式包括缺失、错位、冗余及三种噪声的混合，噪声的比例分别为 10%,20%,…,50%，这样每一条完全服从的执行序列都可以导出 20 条不同的不服从序列，加噪后不服从的日志迹用 σ'表示，于是每个流程对应 200 条不服从的日志迹。接着，先直接利用现有 Prom 插件计算 c_{st} 和 $c_{\text{max-sync}}$ 下 σ'与流程模型的对齐；再基于算法 2 计算 $c_{\text{imp-sync}}$ 下 σ'与流程模型的对齐。对齐结果中的模型执行序列用 σ''表示，于是每个流程对应 600 条相互独立的对齐结果。最后，依据这些对齐结果，分别比较三种成本函数下对齐的准确率和效率，并用可视化的三维图直观的展示对比结果。

4.3 准确性评估

对每一个流程模型，将最初获得的完全服从执行序列 σ，作为评估不同成本函数下对齐结果所返回的模型执行序列 σ''是否准确的标准。也就是说如果 σ 和 σ''完全相同，则准确率记为 1，于是当 σ 和 σ''不完全相同时，就需要通过判断两个序列的相似度来获取不同成本函数下对齐结果的准确率。在下面的实验中，本文利用 python 官方库 difflib 中的 SequenceMatcher 类来计算两序列的相似度，其思想是找到不包含无用元素的最长连续匹配子序列，然后递归地将相同的思想应用于匹配子序列的左边和右边的序列片段。

根据上述的实验数据和评估标准，可以分别对不同噪声类型下三种函数得出的对齐结果进行准确率的比较。当不服

从的日志仅包含缺失噪声时，重要同步成本函数分别与标准成本函数和最大同步成本函数的对齐结果对比如图 7 所示。

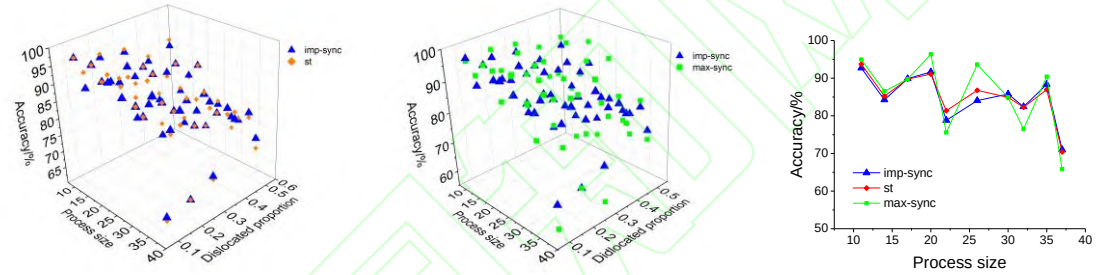


(a)$c_{\text{imp-sync}}$ 与 c_{st} 的比较　(b)$c_{\text{imp-sync}}$ 与 $c_{\text{max-sync}}$ 的比较　(c)各比例下准确率的平均值

图 7　不服从日志中仅包含缺失噪声时的对齐结果比较

Fig.7　The alignment results comparison when non-conformance log trace only contain missing noise

从图 7 中可以得出，当不服从的日志仅包含缺失噪声时，$c_{\text{imp-sync}}$ 下的平均对齐准确率为 90.69%，结果略优于 c_{st} 的 90.59%，但是与 $c_{\text{max-sync}}$ 下的对齐结果相比准确率偏低。另一方面，用标准差衡量图 7(c)中各准确率的离散程度时，可得 $c_{\text{imp-sync}}$、c_{st} 及 $c_{\text{max-sync}}$ 下的标准差分别为 3.15、3.25 和 3.92。这表明，与其他两种成本函数相比，$c_{\text{imp-sync}}$ 下的对齐结果更加稳定。下面考虑当不服从的日志仅包含错位噪声时三种成本函数的对齐结果，结果对比如图 8 所示。

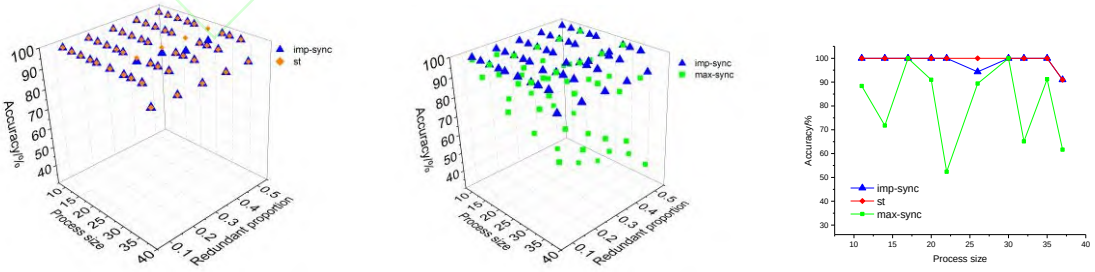


(a)$c_{\text{imp-sync}}$ 与 c_{st} 的比较　(b)$c_{\text{imp-sync}}$ 与 $c_{\text{max-sync}}$ 的比较　(c)各比例下准确率的平均值

图 8　不服从日志中仅包含错位噪声时的对齐结果比较

Fig.8　The alignment results comparison when non-conformance log trace only contain dislocated noise

从图 8 中可以看出，当不服从的日志仅包含错位噪声时，三种成本函数下的对齐结果均有较大的波动。其中 $c_{\text{max-sync}}$ 下的对齐结果起伏波度最大，标准差为 9.40，相比之下 $c_{\text{imp-sync}}$ 与 c_{st} 的波度较小且起伏轨迹十分接近，标准差分别为 6.18 和 6.13。下面考虑当不服从的日志仅包含冗余噪声时，三种成本函数的对齐结果，结果对比如图 9 所示。



(a)$c_{\text{imp-sync}}$ 与 c_{st} 的比较　(b)$c_{\text{imp-sync}}$ 与 $c_{\text{max-sync}}$ 的比较　(c)各比例下准确率的平均值

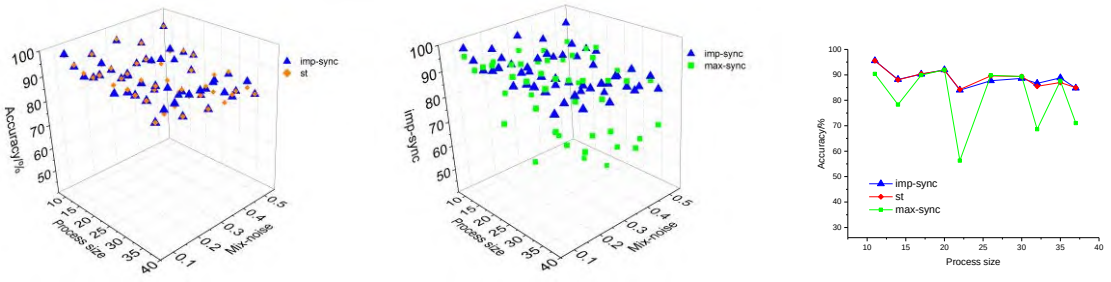
图 9　不服从日志中仅包含冗余噪声时的对齐结果比较

Fig.9　The alignment results comparison when non-conformance log trace only contain redundant noise

从图 9 中可以看出，当不服从的日志仅包含冗余噪声时，各流程在 $c_{\text{imp-sync}}$ 和 c_{st} 下对齐结果的准确率十分平稳，平均准确率分别为 98.53%和 99.10%。相比之下，$c_{\text{max-sync}}$ 下的对齐结果波度较大且平均准确率最低，仅为 81.09%，与 $c_{\text{imp-sync}}$ 和 c_{st} 下的对齐准确率分别相差 17.44%和 18.01%。这是因为 $c_{\text{max-sync}}$ 下在对齐过程中过分追求同步移动的数量的最大化，

于是当出现冗余噪声时，会尽可能牺牲模型移动来将该噪声事件与模型中的某一活动形成同步移动，从而降低准确率。

下面考虑当不服从的日志包含缺失、错位和冗余三种类型的混合噪声时，对齐结果对比如图 10 所示。



(a)$c_{\text{imp-sync}}$ 与 c_{st} 的比较　　(b)$c_{\text{imp-sync}}$ 与 $c_{\text{max-sync}}$ 的比较　　(c)各比例下准确率的平均值

图 10　不服从日志中包含混合噪声时的对齐结果比较

Fig.10 Alignment results comparison when non-conformance log trace contain mixed noise

从图 10 中可以得出，当不服从的日志包含混合噪声时，$c_{\text{imp-sync}}$ 下的平均对齐准确率最高，为 88.67%；c_{st} 其次，为 88.65%，均大幅度高于 $c_{\text{max-sync}}$ 下的 81.34%。此外，$c_{\text{imp-sync}}$ 下的对齐结果也最为稳定，标准差为 3.21；c_{st} 其次，标准差为 3.29；$c_{\text{max-sync}}$ 下的对齐结果最不稳定，标准差高达 11.61。事实上，知道，在实际的业务流程执行过程中，不服从日志的噪声在绝大多数情况下并不是单一类型的。而图 10 表明，在包含混合噪声的情况下，$c_{\text{imp-sync}}$ 能够得出准确率更高且更稳定的对齐结果。

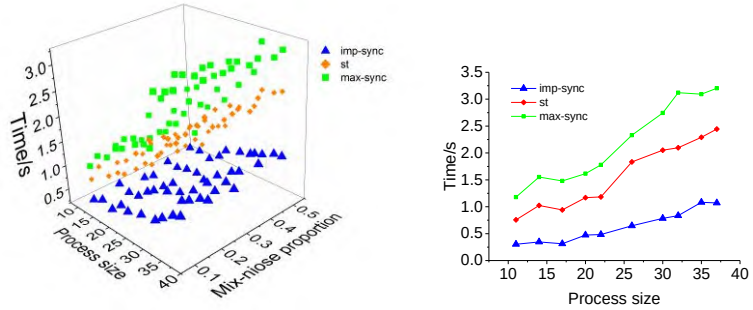
综上，可以得出结论，较之于标准成本函数和最大同步成本函数，重要同步成本函数下的对齐结果满足准确率的需求，且在某些情况下能够产出更高准确率、更稳定的对齐结果。

4.4 效率评估

为了评估算法 2 中所提对齐算法的效率，同样与另外两种函数在对齐所耗时间上进行比较。这里 σ'与流程模型在 c_{st} 和 $c_{\text{max-sync}}$ 下的对齐时间可直接通过对齐插件获取并记录，$c_{\text{imp-sync}}$ 下 σ'与流程模型的对齐时间则通过累加各个子序列和子模型的对齐时间得到。

根据上述的实验数据和评估标准，当不服从的日志包含包含缺失、错位和冗余三种类型的混合噪声时，重要同步成本函数分别与标准成本函数和最大同步成本函数的对齐所耗时间对比如图 11 所示。

从图 11 中可以得出，$c_{\text{imp-sync}}$、c_{st} 和 $c_{\text{max-sync}}$ 下对齐流程模型和事件日志的平均耗时分别为 0.63s、1.58s 和 2.21s，即较之于现存的两种成本函数，所提成本函数下的对齐效率分别提升了约 150.79%和 250.79%。这得益于对模型和不服从日志的分割，因为分段后的对齐能够大大缩减对齐时的搜索空间。相比之下，$c_{\text{max-sync}}$ 下对齐的平均耗时最高，主要因为在对齐过程中产出了大量的同步移动。上述结果说明算法 2 中所提方法能够很好的提升模型与不服从日志的对齐效率。



(a)$c_{\text{imp-sync}}$、c_{st} 与 $c_{\text{max-sync}}$ 的比较　　(b)各比例下准确率的平均值

图 11　不服从日志中包含混合噪声时的对齐耗时对比

Fig.11　Comparison of alignment time when the non-conformance log trace contain mixed noise

5 结语

本文针对现存成本函数存在的问题给出了一种新的成本函数，在该成本函数下能够优先同步操作者自定义的活动类，进而得出更贴近感知成本的对齐结果。同时，基于该函数提出了一种能够提升对齐效率的算法。实验部分通过对大量对齐结果数据的分析，验证了所提函数及对应算法能够在满足准确率需求的同时提升对齐的效率。在未来的工作中，将考虑进一步提升算法的效率，以便所提方法能更好的应用于实际的问题情境中。

参考文献

[1] VAN ZELST S J, BOLT A, VAN DONGEN B F. Tuning Alignment Computation: An Experimental Evaluation[J]. ATAED@ Petri Nets/ACSD, 2017, 1847: 6-20.

[2] VAN DER AALST W M P, ADRIANSYAH A, VAN DONGEN B. Replaying history on process models for conformance checking and performance analysis[J]. WIREs Data Mining Knowl Discov,2012, 2(2): 182-192.

[3] SONG W, XIA X X, HANS ARON J, et al. Efficient Alignment Between Event Logs and Process Models[J]. IEEE Transactions on Services Computing, 2017,10, (1): 136-149.

[4] TAYMOURI F, CARMONA J. A recursive paradigm for aligning observed behavior of large structured process models [C]// Proceedings of the 2016 International Conference on Business Process Management. Cham: Springer, 2016: 197-214.

[5] 田银花，杜玉越，韩咚，等.基于 Petri 网的事件日志与过程模型对齐方法[J].计算机集成制造系统，2019,25(4):809-829.(TIAN Y H, DU Y Y, HAN D, et al. A Method of Aligning Event Log and Process Model Based on Petri Nets [J]. Computer Integrated Manufacturing Systems,2019,25(4):809-829.)

[6] 韩咚，田银花，杜玉越，等.基于 Petri 网可达图的业务对齐方法[J].计算机集成制造系统，2020,26(06):1589-1606.(HAN D, TIAN Y H, DU Y Y, et al. A Business Alignment Method Based on Petri Net Accessibility Graph [J]. Computer Integrated Manufacturing Systems,2020,26(6):1589-1606.)

[7] BLOEMEN V, VAN ZELST S J, VAN DER AALST W M P, et al. Maximizing Synchronization for Aligning Observed and Modelled Behaviour [C]// Proceedings of the 2018International Conference on Business Process Management. Cham: Springer, 2018: 233-249.

[8] SANI M F, VAN ZELST S J, VAN DER AALST W M P. Repairing outlier behaviour in event logs using contextual behaviour[J]. Enterprise Modelling and Information Systems Architectures, 2019, 14: 5: 1-24.

[9] 张力雯，方贤文.基于对齐处理与偏差检测的业务流程适合度分析[J].计算机集成制造系统，2020,26(6):1573-1581.(ZHANG L W, FANG X W. Business process fitness analysis based on alignment processing and deviation detection [J]. Computer Integrated Manufacturing Systems,2020,26(6):1573-1581.)

[10] MATTHIAS W, JAN M. Perceived consistency between process models[J].Information Systems， 2010,37(2):80-98.

[11] 王前东.一种带匹配路径约束的最长公共子序列长度算法[J].电子与信息学报， 2017,39(11):2615-2619.(WANG Q D. A longest common subsequence length algorithm with matching path constraint [J].Journal of Electronics & Information Technology,2017,39(11):2615-2619.)

[12] JOUCK T, DEPAIRE B. PTandLogGenerator: A generator for artificial event data[J]. BPM (Demos), 2016, 1789: 23-27.

[13] JOUCK T, DEPAIRE B. Generating artificial data for empirical analysis of control-flow discovery algorithms[J]. Bus Inf Syst Eng,2019, 61:695–712..

[14] DE LEONI M, VAN DER AALST W M P. Data-aware process mining: discovering decisions in processes using alignments[C]// Proceedings of the 28th Annual ACM Symposium on Applied Computing. New York: ACM,2013:1454–1461.

[15] BAZHENOVA E, BUELOW S, WESKE M. Discovering decision models from event logs[C]// Proceedings of the 2016 International Conference on Business Information Systems. Cham: Springer, 2016: 237–251.

This work is partially supported by the National Natural Science Foundation of China (61402011,61572035), the Anhui Natural Science Foundation of China (1508085MF111, 1608085QF149), the Graduate Innovation Fund project of Anhui University of Science and Technology (2019CX2068).

LI Duoqin,born in 1996, M. S. candidate. Her main research interests include Petri nets, process mining.

FANG Xianwen,born in 1975, Ph.D., professor, PhD supervisor, is a member of CCF(20851M). His main research interests include Petri nets, trusted software.

WANG Lili,born in1982,Ph. D candidate., associate professor,Her main research interests include Petri nets, business process mining.

SHAO Chifeng, born in1995,M. S. candidate,is a member of CCF(B6740G).His main research interests include Petri nets, process mining, model repair and model optimization.