

基于 Petri 网的并发事件流程模型修复分析^{*}

杨慧慧, 方贤文, 邵叱风

(安徽理工大学数学与大数据学院, 安徽 淮南 232001)

摘要:目前, 流程模型可以从大量的事件日志中挖掘出来, 以重放大多数的日志。但是, 少数偏离流程模型的日志亦是有效的, 为了使事件日志与流程模型更加拟合, 模型修复是一个很好的方法。提出了基于 Petri 网的并发事件流程模型修复分析方法。首先, 找到事件日志与流程模型的最优对齐, 筛选出用于修复的并发事件; 其次, 利用提出的重构子流程的修复方法, 对筛选得到的并发事件进行重构; 最后, 根据算法嵌入到原始模型中以实现模型修复, 并通过一个具体实例说明了该方法的合理有效性。修复后的模型可以完全重放给定的事件日志, 并且能够避免因循环造成的多余行为的发生, 同时也在最大程度上保留了原始模型的使用价值。

关键词: Petri 网; 模型修复; 对齐; 块结构; 并发事件

中图分类号: TP311.5

文献标志码: A

doi: 10.3969/j.issn.1007-130X.2021.10.009

Repair analysis of concurrent event process model based on Petri net

YANG Hui-hui, FANG Xian-wen, SHAO Chi-feng

(College of Mathematics and Big Data, Anhui University of Science and Technology, Huainan 232001, China)

Abstract: At present, the process model can be mined from a large number of event logs to replay most of the logs. However, a few logs that deviate from the process model are also valid, and model repair is a good way to make the event log fit more with the process model. This paper proposes a repair analysis method of concurrent event process model based on Petri net. Firstly, the optimal alignment between the event log and the process model is found to filter out the concurrent events for repair. Secondly, the filtered concurrent events are reconstructed by using the proposed reconstruction sub-process repair method. Finally, the algorithm is embedded into the original model to realize the model repair, and the reasonable effectiveness of the method is illustrated through a concrete example. The repaired model can completely replay the given event log, and it can avoid the occurrence of redundant behavior caused by the loop, while also preserving the use value of the original model to the maximum extent.

Key words: Petri net; model repair; alignment; block structure; concurrent event

1 引言

流程挖掘是工作流建模中最有用的工具, 它可以从事件日志中获取客观的、有价值的信息, 建立

流程模型^[1]。流程挖掘主要包括 3 种类型: 流程发现、一致性检查和流程增强。但是, 通过流程发现得到的模型有时会出现与原始模型没有任何相似之处的情况, 这样会丢失原始模型的使用价值, 因此为了避免这种损失, 许多学者对模型修复的研究

^{*} 收稿日期: 2020-05-13; 修回日期: 2020-09-01

基金项目: 国家自然科学基金(61402011, 61572035)

通信作者: 方贤文(280060673@qq.com)

通信地址: 232001 安徽省淮南市泰丰大街 168 号安徽理工大学数学与大数据学院

Address: College of Mathematics and Big Data, Anhui University of Science and Technology, 168 Taifeng Street, Huainan 232001, Anhui, P. R. China

越来越关注。

模型修复与流程挖掘的3大类型是不同的。流程发现主要是从工作流日志中发现并提取流程模型,允许用户构建表示记录在日志中的流程行为的流程模型^[2]。例如,文献[3]探讨了一类基于业务流程执行事件的在线流的流程发现。还有一些研究侧重于评估所发现的流程模型,文献[4]提出了一种新的方法来评估人为产生的负面事件的流程模型,这种方法允许为发现的流程模型定义行为度量。对于一致性检查,其目的是评估记录在事件日志中的流程模型和事件数据是否一致^[5]。文献[6]利用用于交互式流程发现的底层技术提出了一个框架(及其应用)来快速计算一致性,以便增量地更新一致性结果。对于流程增强,文献[7]提供了流程增强的概述,描述了流程增强在各种工业中的应用以及在选择增强方法时考虑增强流程的目标和约束的重要性,相关的增强方法有分解模型修复^[8]、使用角色信息来增强业务流程模型^[9]等。而模型修复主要针对事件日志与原始模型不符合的部分进行修复,最大程度地与原始模型保持相似。文献[10,11]均研究了修复流程模型的问题,后者提出通过事件日志的触发序列,可以从删除行为、添加行为和更改行为3个方面对流程模型进行修复,修复后的流程模型均可以重放日志,并且与原始模型尽可能相似。因此,模型修复的方法既保留了原始模型的价值,又顺应了现实的流程执行,具有一定的研究价值。

本文余下部分的结构为:第2节介绍与本文相关的基本概念;第3节提出新的修复方法来解决本文所针对的问题;第4节用实例分析本文所提方法的可行性;第5节总结全文并对未来工作做出展望。

2 基本概念

定义1(Petri网)^[12] 满足下列条件的三元组 $N = (P, T, F)$ 称作一个 Petri 网:

- (1) $P \cup T \neq \emptyset$;
- (2) $P \cap T = \emptyset$;
- (3) $F \subseteq (P \times T) \cup (T \times P)$;
- (4) $\text{dom}(F) \cup \text{cod}(F) = P \cup T$ 。

其中, P 和 T 是2个不相交的集合,它们是 Petri 网 N 的基本元素集, P 中的元素称为 P -元或库所,也称为位置, T 中的元素称为 T -元或变迁, F 是 Petri 网 N 的流关系。 $\text{dom}(F) = \{x \in P \cup T \mid$

$\exists y \in P \cup T: (x, y) \in F\}$, $\text{cod}(F) = \{x \in P \cup T \mid \exists y \in P \cup T: (y, x) \in F\}$ 。 $\text{dom}(F)$ 表示 F 的定义域, Petri 网 N 的流关系中2个相邻节点 x 与 y 之间为因果关系, $\text{dom}(F)$ 为所有满足条件的 x 的集合;而 $\text{cod}(F)$ 表示 F 的值域,是2个相邻节点 x 与 y 之间满足 $\text{cod}(F)$ 的条件的所有 x 的集合。

定义2(迹,事件日志)^[13] 设 Φ 是一组活动名称的集合, Φ^* 是活动发生序列的一个集合,若 $\alpha \in \Phi^*$ 是一个活动序列,则称 α 是一条迹。若 $L \in B(\Phi^*)$ 是迹的有限非空的多重集,则称 L 是一个事件日志。

定义3(对齐)^[13] 设 $N = (P, T, F, \alpha, m_0, m_f)$ 是一个 Petri 网,其 m_0 为初始标识, m_f 为终止标识。设 $\alpha = a_1 a_2 \cdots a_\theta$ 是一条迹,其中 θ 表示活动序列中触发的活动事件的个数, α 到 Petri 网流程模型 N 之间的对齐 $\gamma = (a_1, t_1)(a_2, t_2) \cdots (a_z, t_z)$ (其中 (a_z, t_z) 表示对齐中第 z 组移动对)是满足下列条件的移动序列:

- (1) 对齐 γ 的第1行 (a_1, a_2, \cdots, a_z) 是事件日志中的一条迹;
- (2) 对齐 γ 的第2行 (t_1, t_2, \cdots, t_z) 是 Petri 网流程模型 N 中的一组变迁发生序列;
- (3) 对于所有的 $i = 1, 2, \cdots, z$, 如果 $t_i \neq \gg$, $\alpha(t_i) \neq \tau$, 且 $a_i \neq \gg$, 那么 $\alpha(t_i) = a_i$ 。

这里的移动对满足以下条件:

- (1) 如果 $a_i = \gg \wedge t_i \neq \gg$, 则称为模型移动;
- (2) 如果 $a_i \neq \gg \wedge t_i = \gg$, 则称为日志移动;
- (3) 如果 $a_i \neq \gg \wedge t_i \neq \gg$, 则称为同步移动。

定义4(状态集)^[14] 设 $N_s = (S, S_I, S_F, \Phi, T)$ 是一个 Petri 网流程模型 N 的一个状态集,其中:

- (1) S 是模型中的状态, $S_I \subseteq S$ 是初始状态, $S_F \subseteq S$ 是终止状态;
- (2) Φ 是一组活动集;
- (3) $T \subseteq S \times \Phi \times S$ 。

例如,若 $(s_1, act, s_2) \in T$, 则说明状态 s_1 通过变迁 act 可以转变为状态 s_2 。

3 基于并发事件的模型修复

3.1 最优对齐以及问题的提出

对于给定的迹和 Petri 网流程模型 N , 可能存在多个不同的对齐, 为了方便比较不同对齐与流程模型之间偏差的大小, 本文分别赋予同步移动、日志移动和模型移动的成本函数值为0,1,1, 特别

地, (\gg, τ) 的成本函数值为 0。因此, 一个迹对齐 γ 的成本 $c(\gamma)$ 为 γ 中所有移动的成本函数值的和。 $c(\gamma)$ 越小则偏差越小, 因此本文将考虑最优对齐。

定义 5(最优对齐)^[15] 给定一条迹 α 和 Petri 网流程模型 N , 对于 α 和 N 之间所有的对齐 χ , $\exists \gamma \in \chi$, 对 $\forall \gamma' \in \chi$, 都有 $c(\gamma) \leq c(\gamma')$, 则称 γ 是 α 和 N 之间的最优对齐。

例如, 考虑本文第 4 节给出的迹 α_1 , 可以看出 α_1 与事故保险理赔流程模型 N 之间存在 3 种不同的对齐, 即 $\gamma_{(\alpha_1)}$ 、 $\gamma'_{(\alpha_1)}$ 和 $\gamma''_{(\alpha_1)}$, 如式(1)所示, 由于 $2 = c(\gamma_{(\alpha_1)}) < c(\gamma'_{(\alpha_1)}) = 3 < c(\gamma''_{(\alpha_1)}) = 4$, 因此称 $\gamma_{(\alpha_1)}$ 是 α_1 和 N 之间的最优对齐。

本文第 4 节给出了 3 个事件日志, 共有 12 条迹, 其中 $\alpha_1, \alpha_6, \alpha_7, \alpha_{11}, \alpha_{12}$ 这 5 条迹与事故保险理赔流程模型存在偏差, 但是观察发现这 5 条迹在现实生活中皆是合理的。交通事故涉及到的纠纷比较复杂, 若当事人直接选择走司法程序则会出现 α_1 的情形, 由司法程序进行裁决, 并对相关损失作出估价; 通常需要走司法程序的时候都会开具相关证明, 在办理理赔手续的过程需要用到这些证明, 即 α_6 和 α_7 ; 当事故较小时也可以不进行现场勘查, 若在允许理赔情况下, 当事人对理赔金额存在异议, 则保险公司便不执行汇款, 等待异议解除后确定了最终金额再进行汇款, 即 α_{11} 和 α_{12} 。然而这几条合理有效的迹并不能完全在流程模型 N 中重

放, 说明理赔系统目前不够完善, 应该进一步修复模型, 使得模型能更好地反映出各个事件日志。

3.2 重构子流程修复模型

目前, 已有的经典修复方法允许简单的插入和跳过活动, 在 $\gamma_{(\alpha_1)}$ 中有 2 个模型移动, 即 (\gg, h) , (\gg, j) 。活动事件 h 和 j 不能在迹中观察到但在模型中却存在, 这种情况下允许插入隐变迁跳过相应行为活动来进行模型修复, 如图 1 所示。

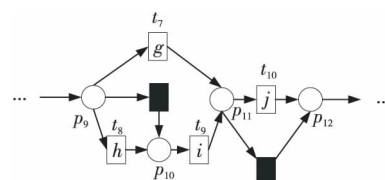


Figure 1 Model repair diagram of the insertion of implicit transitions

图 1 插入隐变迁的模型修复图

图 1 中变量与本文第 4 节一致, 其中 $p_9 \sim p_{12}$ 表示库所的序号, $t_7 \sim t_{10}$ 表示变迁的序号, g 表示轻微事故, 损失较小, h 表示重大事故, 损失较大, i 表示进入司法程序, j 表示事故现场勘查检验。对于迹 α_{11} 和 α_{12} 与流程模型 N 之间的最优对齐 $\gamma_{(\alpha_{11})}$ 和 $\gamma_{(\alpha_{12})}$ 如式(2)所示, 都包含了 2 个连续的日志移动 (s, \gg) 和 (u, \gg) , 只是顺序不同, 对于这种情况, 可以根据文献[13]中基于子流程的修复算法进行模型修复, 如图 2 所示。

$$\begin{aligned}
 \gamma_{(\alpha_1)} &:= \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|} \hline a & b & e & d & c & f & \gg & i & \gg & k & m & l & n & q & r \\ \hline a & b & e & d & c & f & h & i & j & k & m & l & n & q & r \\ \hline t_1 & t_2 & t_5 & t_4 & t_3 & t_6 & t_8 & t_9 & t_{10} & t_{11} & t_{13} & t_{12} & t_{14} & t_{17} & t_{18} \\ \hline \end{array}, \\
 \gamma'_{(\alpha_1)} &:= \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|} \hline a & b & e & d & c & f & i & \gg & \gg & k & m & l & n & q & r \\ \hline a & b & e & d & c & f & \gg & g & j & k & m & l & n & q & r \\ \hline t_1 & t_2 & t_5 & t_4 & t_3 & t_6 & & t_7 & t_{10} & t_{11} & t_{13} & t_{12} & t_{14} & t_{17} & t_{18} \\ \hline \end{array}, \\
 \gamma''_{(\alpha_1)} &:= \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|} \hline a & b & e & d & c & f & i & \gg & \gg & \gg & k & m & l & n & q & r \\ \hline a & b & e & d & c & f & \gg & h & i & j & k & m & l & n & q & r \\ \hline t_1 & t_2 & t_5 & t_4 & t_3 & t_6 & & t_8 & t_9 & t_{10} & t_{11} & t_{13} & t_{12} & t_{14} & t_{17} & t_{18} \\ \hline \end{array} \quad (1) \\
 \gamma_{(\alpha_{11})} &:= \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|} \hline a & b & c & d & e & f & g & & k & l & m & n & o & s & u & p & r \\ \hline a & b & c & d & e & f & g & j & k & l & m & n & o & \gg & \gg & p & r \\ \hline t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 & t_{10} & t_{11} & t_{12} & t_{13} & t_{14} & t_{15} & & & t_{16} & t_{18} \\ \hline \end{array}, \\
 \gamma_{(\alpha_{12})} &:= \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|} \hline a & b & d & c & e & f & h & i & j & k & m & l & n & o & u & s & p & r \\ \hline a & b & d & c & e & f & h & i & j & k & m & l & n & o & \gg & \gg & p & r \\ \hline t_1 & t_2 & t_4 & t_3 & t_5 & t_6 & t_8 & t_9 & t_{10} & t_{11} & t_{13} & t_{12} & t_{14} & t_{15} & & & t_{16} & t_{18} \\ \hline \end{array}, \quad (2)
 \end{aligned}$$

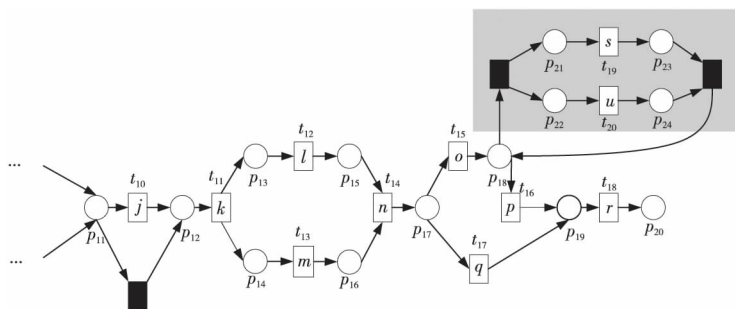


Figure 2 Model repair partial display diagram based on subprocess

图 2 基于子流程的模型修复部分显示图

图 2 中变量与本文第 4 节一致,其中 $p_{11} \sim p_{24}$ 表示库所的序号, $t_{10} \sim t_{20}$ 表示变迁的序号。根据上述模型修复方法,修复日志移动需要插入相应的活动,那么就有可能发生具有相同行为的同一个活动在模型中不同位置重复出现,分别多次构成自循环的情况,增加了很多不必要的循环行为。例如,式(3)所示的最优对齐 $\gamma_{(a_6)}$ 和 $\gamma_{(a_7)}$ 中都含有日志移动 (v, \gg) , 显然,若在模型中插入 2 次相同的活动 v 会构成两个自循环结构,如图 3 所示,模型允许出现活动 v 多次循环的行为: $(\dots, j, v, v, k, l, m, n, \dots), (\dots, j, v, v, v, k, l, m, n, \dots), (\dots, j, v, k, v, v, l, m, n, \dots), (\dots, j, k, v, v, v, l, m, n, \dots), \dots$, 而日志中并未出现任何循环行为。

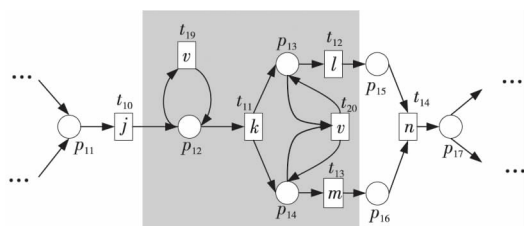


Figure 3 Schematic diagram of unnecessary cyclic behavior

图 3 不必要的循环行为示意图

因此,本文针对类似于 $\gamma_{(a_6)}$ 和 $\gamma_{(a_7)}$ 包含的这类对齐提出一种新的修复方法,即先将相关活动的所有流关系断开,然后重新构造一个并发结构的子流程,最后将重构的子流程连接到流程模型中以达到修复效果。

这种修复方法适合修复并发事件,因此本文研究的并发事件符合下列 2 个条件:

(1) $w(w \geq 2)$ 个对齐中皆含有相同活动事件的日志移动,且原流程模型中不包含此活动事件。

(2) 若将此日志移动的活动事件与相邻同步移动的活动事件视为一组并发事件 ξ_2 , 且 $\xi_{2\uparrow} \in \beta_{i-1}$, $\xi_2 \in \beta_i$, $\xi_{2\downarrow} \in \beta_{i+1}$, 其中 $\beta_{i-1}, \beta_i, \beta_{i+1}$ 分别属于块结构工作流网^[16]中的一个结构,则有 $\beta_{i-1}, \beta_i, \beta_{i+1}$ 为顺序关系,且在 $w(w \geq 2)$ 个对齐中均成立。其中, $\xi_{2\uparrow}$ 和 $\xi_{2\downarrow}$ 分别表示对应迹中 ξ_2 的前一个活动事件和后一个活动事件。

为了筛选符合条件的并发事件,需要考虑流程模型的顺序结构,因此用块结构工作流网^[16]来展示更为直观,如图 5 所示。下面将给出详细的筛选算法。

算法 1 筛选符合条件的并发事件

输入:所有最优对齐 γ , 流程模型 N 。

输出:所有符合条件的并发事件 ξ 。

步骤 1 观察所有的最优对齐 γ , 保留含有日志移动的对齐, 设保留后的对齐为 $\gamma_1, \gamma_2, \dots, \gamma_p^*$, 令 $\eta_1 = \{\gamma_1, \gamma_2, \dots, \gamma_p^*\}$ 。

步骤 2 将 η_1 中的对齐进行分组, 包含相同日志移动的对齐分为一组, 然后将单独成组的对齐删除, 设分组完成后形成 λ 组, 即 $G_1, G_2, \dots, G_\lambda (\lambda \geq 1)$ 。

$$\gamma_{(a_6)} := \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|} \hline a & b & d & c & e & f & h & i & j & k & v & l & m & n & o & p & r \\ \hline a & b & d & c & e & f & h & i & j & k & \gg & l & m & n & o & p & r \\ \hline t_1 & t_2 & t_4 & t_3 & t_5 & t_6 & t_8 & t_9 & t_{10} & t_{11} & & t_{12} & t_{13} & t_{14} & t_{15} & t_{16} & t_{18} \\ \hline \end{array},$$

$$\gamma_{(a_7)} := \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|} \hline a & b & d & e & c & f & h & i & j & v & k & l & m & n & o & p & r \\ \hline a & b & d & e & c & f & h & i & j & \gg & k & l & m & n & o & p & r \\ \hline t_1 & t_2 & t_4 & t_5 & t_3 & t_6 & t_8 & t_9 & t_{10} & & t_{11} & t_{12} & t_{13} & t_{14} & t_{15} & t_{16} & t_{18} \\ \hline \end{array}$$

(3)

步骤 3 采用流程树语言对流程模型进行描述,列出构成顺序结构的各个分支 $\beta_1, \dots, \beta_k^*$, 然后执行步骤 4 对 $G_1, G_2, \dots, G_\lambda (\lambda \geq 1)$ 分别进行组内筛选。

步骤 4 观察组内各个对齐,保留日志移动(包含连续日志移动)前后均为同步移动的对齐,其余情况的对齐删除。若 $|G_i| < 2$, 则删除第 i 组, 否则执行步骤 5。

步骤 5 在 G_i 中的各个对齐内,设日志移动的活动事件为 α , 若均满足 $\alpha_{\uparrow} \in \beta_i, \alpha_{\downarrow} \in \beta_{i+1}, 1 \leq i \leq k-1$, 则删除 G_i , 然后对剩余各组执行步骤 6。

步骤 6 在各组中分别对目标日志移动中的活动事件 δ 进行分析,若将 δ 与相邻活动事件 a 视为一组并发事件 $\xi_j = [\delta, a], a \in \beta_i$ 并且在组内各个对齐中均满足 $\xi_{j\uparrow} \in \beta_{i-1}, \xi_j \in \beta_i, \xi_{j\downarrow} \in \beta_{i+1}, 2 \leq i \leq k-1$, 则保留符合条件的并发事件 ξ_j , 否则删除此组对齐。

步骤 7 得到各组中保留的符合条件的并发事件 ξ_1, ξ_2, \dots , 输出所有符合条件的并发事件 $\xi = \{\xi_1, \xi_2, \dots\}$, 算法结束。

找到符合条件的并发事件后,便可以对模型进行修复,重构一个并发结构的子流程,然后将其连接到模型中,使得事件日志与流程模型偏差为零,保证了流程模型能够更好地反映出实际事件日志的情况。具体步骤如算法 2 所示。

算法 2 重构子流程模型修复

输入: 一组并发事件 $\xi_j = \langle a_1, a_2, \dots, a_* \rangle$, 流程模型 N , 含有 ξ_j 的一组对齐 $\gamma(\alpha)$ 。

输出: 含有重构子流程的修复模型 N' 。

步骤 1 设 $\gamma(\alpha)$ 中存在 $(a_1, t_1), (a_2, \gg), \dots, (a_*, \gg)$, 其中 $t_1 \neq \gg$, 令 $\cdot t_1 = \{p'\}, t_1^* = \{p''\}$;

步骤 2 将模型中 t_1 的输入边及输出边全部删除,然后在 t_1 的对应位置插入变迁 t_2, \dots, t_{n^*} , 满足 $\cdot t_i^* = |t_i^*| = 1, \cdot t_i^* \cap t_i^* = \cdot t_i^* \cap \cdot t_l^* = t_i^* \cap t_l^* = \cdot t_i^* \cap t_l^* = \emptyset, i^*, l^* = 1, 2, \dots, n^*$, 且 $i^* \neq l^*$ 。

步骤 3 在 t_2, \dots, t_{n^*} 的对应位置添加隐变迁 $\tau_2, \dots, \tau_{n^*}$, 满足 $\cdot \tau_i^* = \cdot t_i^*, \tau_i^* = t_i^* (i = 2, \dots, n^*)$;

步骤 4 添加一个隐变迁 τ , 使得 $\tau \in \cdot s'$, 其中 $s' = \{s_i^* \in \cdot t_i^* | i^* = 1, 2, \dots, n^*\}$;

步骤 5 添加一个隐变迁 τ' , 使得 $\tau' \in S'$, 其中, $S' = \{S_i^* \in t_i^* | i^* = 1, 2, \dots, n^*\}$, 从而重构了一个 Petri 网模型并发结构的子流程 U 。

步骤 6 将 U 连接到模型中,满足 $p' \in \cdot U$,

$p'' \in U^*$, 即 p' 为 τ 的输入库所, p'' 为 τ' 的输出库所;

步骤 7 得到一个含有重构子流程的修复模型 N' , 算法结束。

4 实例分析

本节以事故保险理赔流程模型为例,使用本文提出的新方法对原始模型进行修复,使得修复后的模型能够完全重放事件日志,并且尽可能减少不必要的行为发生。最后与目前经典的插入与跳过活动的修复方法进行比较,以验证本文所提方法的可行性。

近年来,随着人们生活水平显著提高,交通事故的保险理赔也越来越受到重视。正常情况下,事故发生之后,第一时间先报案,然后打电话通知保险公司,继而保险公司会进行以下正常的理赔程序: a 报案受理; b 查询保单信息; c 填写出险报案表; d 报案登记; e 人员调度; f 理赔初审; g 轻微事故;损失较小; h 重大事故;损失较大; i 进入司法程序; j 事故现场勘查检验; k 定损估价; v 开具证明; l 填写索赔相关材料; m 提交相关证明; n 理赔复核; o 理赔; s 暂时不支付; u 等待最终确认金额; p 汇款; q 拒赔; r 结案归档。由于事故发生类型错综复杂,事故保险理赔系统难免会出现不符合现实案例的情况,因此进一步完善保险理赔系统是有必要的。本文从保险公司理赔系统中随机调取了部分事件日志,经删除噪声处理后得到如式(3)所示的事件日志:

$$L_1 = \{\alpha_1 = \langle a, b, e, d, c, f, i, k, m, l, n, q, r \rangle^{23}, \alpha_2 = \langle a, b, d, e, c, f, h, i, j, k, l, m, n, q, r \rangle^{316}, \alpha_3 = \langle a, b, d, c, e, f, g, j, k, l, m, n, o, p, r \rangle^{2952}, \alpha_4 = \langle a, b, e, c, d, f, g, j, k, m, l, n, o, p, r \rangle^{2710}\},$$

$$L_2 = \{\alpha_5 = \langle a, b, c, d, e, f, h, i, j, k, l, m, n, o, p, r \rangle^{2108}, \alpha_6 = \langle a, b, c, e, d, f, h, i, j, k, v, l, m, n, o, p, r \rangle^{18}, \alpha_7 = \langle a, b, d, e, c, f, h, i, j, v, k, m, l, n, o, p, r \rangle^{31}\},$$

$$L_3 = \{\alpha_8 = \langle a, b, e, d, c, f, g, j, k, l, m, n, q, r \rangle^{1791}, \alpha_9 = \langle a, b, e, d, c, f, h, i, j, k, m, l, n, q, r \rangle^{536}, \alpha_{10} = \langle a, b, c, e, d, f, h, i, j, k, m, l, n, q, r \rangle^{824}, \alpha_{11} = \langle a, b, c, d, e, f, g, k, l, m, n, o, s, u, p, r \rangle^{29}, \alpha_{12} = \langle a, b, d, c, e, f, h, i, j, k, m, l, n, o, u, s, p, r \rangle^{16}\} \quad (4)$$

其中, L_1, L_2, L_3 表示 3 个不同的事件日志, α_φ ($\varphi = 1, 2, \dots, 12$) 表示 12 条不同的迹,每条迹中后面的数字上标表示这条迹在事件日志中出现的

次数。根据这些事件日志,本文构建了基于 Petri 网的事故保险理赔流程模型 N ,如图 4 所示,该流程模型符合一般的理赔程序。根据 Petri 网定义可知,图 4 显然满足条件(1)~条件(3),对于条件(4),已知对于 $x \in P \cup T$,记 $\cdot x = \{y \mid y \in P \cup T \wedge (y, x) \in F\}$, $x \cdot = \{y \mid y \in P \cup T \wedge (x, y) \in F\}$,称 $\cdot x$ 为 x 的前集, $x \cdot$ 为 x 的后集,因此,图 4 中所有变迁与库所中, $dom(F)$ 表示相邻 2 节点 x 和 y 满足 $(x, y) \in F$ 的 $\cdot y$ 的集合, $cod(F)$ 表示相邻 2 节点 x 和 y 满足 $(y, x) \in F$ 的 $y \cdot$ 的集合,即 $dom(F) = \{p_1, p_2, \dots, p_{18}, p_{19}, t_1, t_2, \dots, t_{17}, t_{18}\}$, $cod(F) = \{p_2, p_3, \dots, p_{19}, p_{20}, t_1, t_2, \dots, t_{17}, t_{18}\}$,满足条件(4),因此,图 4 是合理的 Petri 网模型。

通过比对给定的事件日志与流程模型可以发现,事件日志中大多数的迹都可以在流程模型中重放,但 $\alpha_1, \alpha_6, \alpha_7, \alpha_{11}, \alpha_{12}$ 并不能与流程模型完全拟合,对于 $\alpha_1, \alpha_{11}, \alpha_{12}$,前文已经进行了详细分析,本节主要针对 α_6, α_7 这类行为,对事件日志 L_2 使用本文提出的新方法对原始模型进行修复。

首先用流程树语言描述流程模型的结构,并用矩形区域将流程树节点表示出来,如图 5 所示,可以清楚地看到流程模型中的各个结构,即顺序结构、选择结构和并发结构。

$\rightarrow (a, b, \wedge (c, d, e), f, \times (g, \rightarrow (h, i)), j, k, \wedge (l, m), n, \times (\rightarrow (o, p), q), r)$

然后根据算法 1 筛选符合条件的并发事件,具体的筛选过程如表 1 所示。通过表 1 可以看出 $[k, v]$ 是一组符合条件的并发事件,找到并发事件后根据算法 2 对模型进行修复。首先断开活动 k 的所有流关系,然后与活动 v 重构为一个并发结构

Table 1 Main processes and results of filtering concurrent events

表 1 筛选并发事件的主要过程及结果

执行步骤	筛选过程及结果
1	$\eta_1 = \{\gamma_{(a_6)}, \gamma_{(a_7)}, \gamma_{(a_{11})}, \gamma_{(a_{12})}\}$
2	$G_1 = \{\gamma_{(a_6)}, \gamma_{(a_7)}\}, G_2 = \{\gamma_{(a_{11})}, \gamma_{(a_{12})}\}$
3	$a \in \beta_1, b \in \beta_2, c, d, e \in \beta_3, f \in \beta_4, g, \rightarrow (h, i) \in \beta_5, h \in \beta_{5_1}, i \in \beta_{5_2}, j \in \beta_6, k \in \beta_7, l, m \in \beta_8, n \in \beta_9, \rightarrow (o, p), q \in \beta_{10}, o \in \beta_{10_1}, p \in \beta_{10_2}, r \in \beta_{11}$
4	$G_1 = \{\gamma_{(a_6)}, \gamma_{(a_7)}\}, G_2 = \{\gamma_{(a_{11})}, \gamma_{(a_{12})}\}$
5	$G_1: \gamma_{(a_6)}$ 中, $v \uparrow \in \beta_7, v \downarrow \in \beta_8; \gamma_{(a_7)}$ 中, $v \uparrow \in \beta_6, v \downarrow \in \beta_7$,故保留 G_1 ; G_2 在 $\gamma_{(a_{11})}$ 和 $\gamma_{(a_{12})}$ 中均满足 $\langle s, u \rangle \uparrow \in \beta_{10_1}, \langle s, u \rangle \downarrow \in \beta_{10_2}$,故删除 G_2
6	$\langle k, v \rangle \uparrow \in \beta_6, \langle k, v \rangle \in \beta_7, \langle k, v \rangle \downarrow \in \beta_8$,故得到 $\xi_1 = \langle k, v \rangle$
7	$\xi = \{\xi_1\} = \{\langle k, v \rangle\}$

的子流程 U : 在 $\gamma_{(a_6)}$ 中存在移动对 $[k, k]$, $[v, \gg]$, 在模型中有 $\cdot t_{11} = \{p_{12}\}$, $t_{11} \cdot = \{p_{13}, p_{14}\}$, 首先将 t_{11} 的输入边和输出边全部删除,然后在 t_{11} 的对应位置插入变迁 t_{19} , 满足 $|\cdot t_{19}| = |t_{19} \cdot| = 1, \cdot t_{11} \cap t_{11} \cdot = \cdot t_{19} \cap t_{19} \cdot = \cdot t_{11} \cap \cdot t_{19} = t_{11} \cap t_{19} \cdot = \cdot t_{11} \cap t_{19} \cdot = \emptyset$, 接着在 t_{19} 的对应位置添加隐变迁 τ_1 , 满足 $\cdot \tau_1 = \cdot t_{19}, \tau_1 \cdot = t_{19} \cdot$, 之后再添加隐变迁 τ 和 τ_2 , 分别满足 $\tau \in \cdot s', \tau_2 \in S'$, 其中 $s' = \{p_{21}, p_{22}\}, S' = \{p_{23}, p_{24}\}$ 。此时并发结构的子流程 U (如图 6 所示)构造完成。

最后将 U 连接到模型中,满足 $\cdot U = \{p_{12}\}$, $U \cdot = \{p_{13}, p_{14}\}$, 即得到修复后的模型 N' , 如图 7 所示。

显然,相较于原始模型,含有重构子流程的修

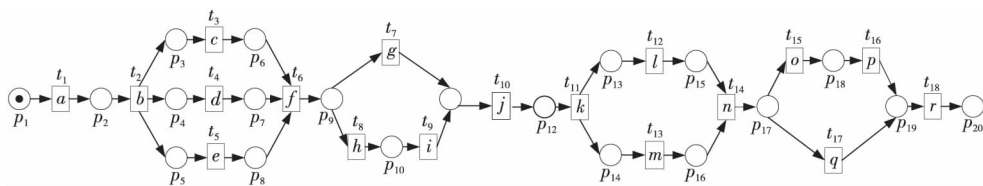


Figure 4 Process model N of accident insurance claim based on Petri net

图 4 基于 Petri 网的事故保险理赔流程模型 N

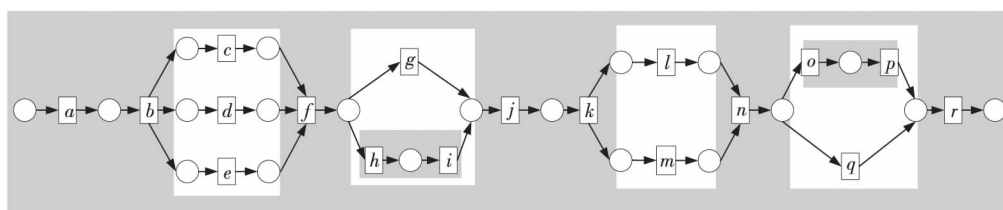
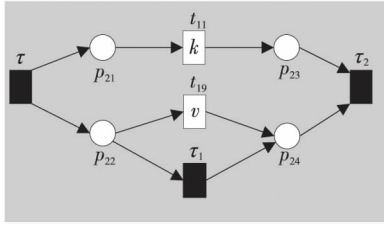


Figure 5 Block structure diagram of Petri net model N

图 5 Petri 网模型 N 的块结构示意图

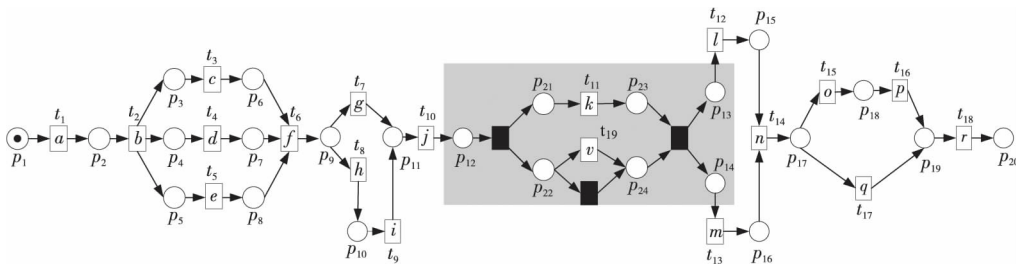
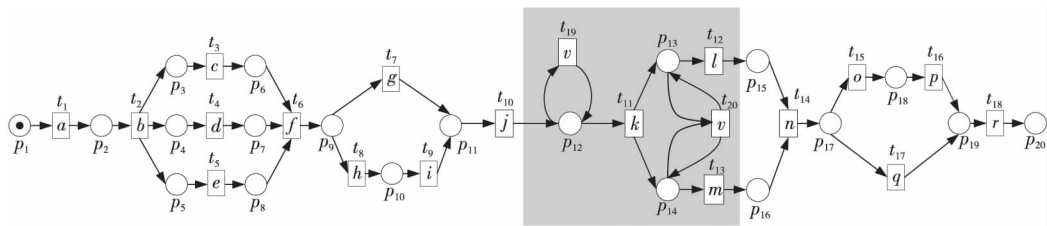
Figure 6 Subprocess U of the refactoring图 6 重构的子流程 U

复模型 N' 可以完全重放事件日志 L_2 。不难发现,在实际生活中,当发生重大事故采取司法程序处理时,都会收到开具的相关证明,以便在理赔手续中提供合法凭证。若使用现有的修复方法,例如,使用目前经典的插入与跳过活动的修复方法所得到的修复模型 N_1 (如图 8 所示),可以看出修复模型 N_1 中包含自循环结构,允许无限多的循环行为,因此图 8 过度概括了日志中出现的行为,导致模型表达不精确。相反,使用本文所提方法修复的模型 N' ,结构相对比较简单,不包含循环结构,不会多次出现循环活动 v 的行为,如图 7 灰色部分所示,只涵盖 3 种执行顺序: $\langle k, v \rangle$, $\langle v, k \rangle$, $\langle k \rangle$, 与 L_2 中的情况完全吻合。

最后,为了对比 N_1 与 N' 这 2 个修复模型的精确性,本文根据文献[14]给出适合度 $fitness$ 和精度 $precision$ 的相关公式,如式(5)所示:

$$fitness(L, N) = 1 - \frac{\sum_{a \in L} c(\gamma_a)}{move_L(L) + |L| \times move_N(N)},$$

$$precision(L, N) = \frac{1}{|\epsilon|} \sum_{e \in \epsilon} \frac{|en_L(e)|}{|en_N(e)|} \quad (5)$$

Figure 7 Repair model N' containing the refactored subprocess图 7 含有重构子流程的修复模型 N' Figure 8 Repair model N_1 with existing methods图 8 现有方法的修复模型 N_1

其中, L 是一个事件日志, $move_L(L) = \sum_{a \in L} \sum_{x \in a} c(x, \gg)$, $move_N(N) = \min_{\sigma_N \in \beta(N)} \sum_{y \in \sigma_N} c(\gg, y)$; σ_N 是流程模型 N 中的一个完整的执行序列, 即从初始状态执行到终止状态的一个完整执行序列; $\beta(N)$ 是流程模型 N 中所有的完整执行序列的集合。 $en_N(e) = \{a \in \Phi_N \mid \exists s_* \in S(state_N(e), act, s_*) \in T\}$, 且 $act(e) \in en_N(e)$; $en_L(e) = \{act(e') \mid e' \in \epsilon \wedge context_L(e') = context_L(e)\}$, $state_N(e)$ 表示活动 e 发生的前一个状态, $context_L(e)$ 表示 e 发生之前在流程实例中的活动前缀; ϵ 是事件日志中每个事件的集合, 例如, 在事件日志 L_2 中, ϵ 包含 $2108 \times 16 + 18 \times 17 + 31 \times 17 = 34561$ 个事件。

由于修复后的模型 N_1 与 N' 均能与事件日志 L_2 完全对齐, 所以 $\sum_{a \in L_2} c(\gamma_a) = 0$, 因此 2 个模型的适合度均为 1, 即:

$$fitness(L_2, N_1) = 1 - 0 = 1$$

$$fitness(L_2, N') = 1 - 0 = 1$$

对于 2 个模型的精度, 根据精度计算公式可得:

$$precision(L_2, N_1) = \frac{28966.96}{34561} \approx 0.84$$

$$precision(L_2, N') = \frac{33398.41}{34561} \approx 0.97$$

显然, $0.84 < 0.97$, 在适合度都为 1 的情况下, 本文方法修复后的模型精度更高一些。因此, 通过上述对比可知, 本文提出的方法具有合理性和可行性。

5 结束语

本文在已有研究的基础上,基于事件日志与流程模型之间的最优对齐,提出了一种重构子流程的修复方法。首先根据最优对齐,利用相关算法筛选符合条件的并发事件,并将其重构为一个并发结构的子流程,之后将子流程连接到原模型中完成修复。最后本文通过事故保险理赔系统的一个实例验证了该方法的合理性。

由于本文提出的种方法对于含有复杂结构的流程模型并不能很好地适用,因此,在未来的研究中,我们希望能进一步拓展,寻找一种在复杂结构的情况下也能普遍适用的修复方法,并尽可能提高相应的适合度以及精度。

参考文献:

- [1] Yu Jian-bo, Dong Chen-yang, Li Chuan-feng, et al. Process mining based on statistical α -algorithm[J]. Journal of Beijing University of Aeronautics and Astronautics, 2018, 44(5): 895-906. (in Chinese)
- [2] Omer S. Performance analysis of business processes using process mining[J]. International Journal of Computer Applications, 2018, 180(37): 27-30.
- [3] van Zelst S J, van Dongen B F, Aalst W M P V D. Event stream-based process discovery using abstract representations [J]. Knowledge & Information Systems, 2018, 54(2): 407-435.
- [4] Weerdt J D, Backer M D, Vanthienen J, et al. A robust F-measure for evaluating discovered process models[C]//Proc of IEEE Symposium on Computational Intelligence & Data Mining, 2011: 148-155.
- [5] van Zelst S J, Bolt A, van Dongen B F. Computing alignments of event data and process models[M]//Transactions on Petri Nets and Other Models of Concurrency XIII. Berlin: Springer, 2018: 1-26.
- [6] Dixit P M, Buijs J C A M, Verbeek H M W, et al. Fast incremental conformance analysis for interactive process discovery [C]//Proc of International Conference on Business Information Systems, 2018: 163-175.
- [7] Yasmin F A, Bukhsh F A, Silva P D A. Process enhancement in process mining: A literature review[C]//Proc of the 8th International Symposium on Data Driven Process Discovery and Analysis, 2018: 64-72.
- [8] Shugurov I, Mitsyuk A. Iskra: A tool for process model repair [J]. Proceedings of ISP RAS, 2015, 27(3): 237-254.
- [9] Burattin A, Sperduti A, Veluscek M. Business models enhancement through discovery of roles [C]//Proc of 2013 IEEE Symposium on Computational Intelligence and Data Mining (CIDM), 2013: 103-110.

- [10] Fahland D, van der Aalst W M P. Repairing process models to reflect reality[C]//Proc of the 10th International Conference on Business Management, 2012: 229-245.
- [11] Wang Lu, Du Yu-yue, Qi Hong-da. Process model repair based on firing sequences[J]. Journal of Computer Research and Development, 2018, 55(3): 585-601. (in Chinese)
- [12] Wu Zhe-hui. Petri nets theory[M]. Beijing: Mechanical Industry Press, 2006. (in Chinese)
- [13] Fahland D, van der Aalst W M P. Model repair—Aligning process models to reality[J]. Information Systems, 2015, 47: 220-243.
- [14] van der Aalst W M P, Adriansyah A, van Dongen B. Replaying history on process models for conformance checking and performance analysis[J]. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2012, 2(2): 182-192.
- [15] Polyvyanyy A, van der Aalst W M P V D, Hofstede A H M T, et al. Impact-driven process model repair[J]. Acm Transactions on Software Engineering & Methodology, 2016, 25(4): 1-60.
- [16] Leemans S J J, Fahland D, van der Aalst W M P. Discovering block-structured process models from event logs—A constructive approach[C]//Proc of International Conference on Applications and Theory of Petri Nets and Concurrency, 2013: 311-329.

附中文参考文献:

- [1] 余建波,董晨阳,李传锋,等. 基于统计 α 算法的过程挖掘[J]. 北京航空航天大学学报, 2018, 44(5): 895-906.
- [11] 王路,杜玉越,祁宏达. 基于引发序列的流程模型修正[J]. 计算机研究与发展, 2018, 55(3): 585-601.
- [12] 吴哲辉. Petri 网理论[M]. 北京:机械工业出版社, 2006.

作者简介:



net.

杨慧慧(1995-),女,河南长垣人,硕士生,研究方向为 Petri 网。E-mail: 1401927003@qq.com

YANG Hui-hui, born in 1995, MS candidate, her research interest includes Petri



方贤文(1975-),男,河南信阳人,博士,教授,研究方向为 Petri 网和可信软件。E-mail: 280060673@qq.com

FANG Xian-wen, born in 1975, PhD, professor, his research interests include Petri net, and trusted software.