arXiv:2301.02185v1 [cs.DB] 3 Jan 2023

发现无阻塞结构的无声自由选择工作流网

Tsung-Hao Huang[0000−0002−3011−9999] 和 Wil M. P. van der Aalst[0000−0002−0955−6940]

过程与数据科学（PADS），亚琛工业大学，德国亚琛 {tsunghao.huang, wvdaalst}@pads.rwth-aachen.de

摘要。 流程发现旨在发现能够解释从信息系统中提取的事件日志行为的模型。尽管已有多种方法被提出，但只有少数方法能够保证健全性和自由选择等理想属性。利用流程树的表示偏置来提供这些保证的最先进方法受到块结构化形式的限制。这类构造限制了所发现模型的表达能力，也就是说，只能发现健全自由选择工作流网的一个子集。为支持更加灵活的结构表示，我们旨在发现能够提供相同保证、同时允许采用非块结构的流程模型。受现有自由选择网理论中运用综合规则的研究启发，我们提出一种自动化方法，通过预定义模式逐步向现有流程模型添加活动。遵循这些规则可确保生成的模型始终健全且自由选择。此外，所发现的模型不受块结构限制，因此更加灵活。该方法已使用 Python 实现，并通过多个真实事件日志进行了测试。实验表明，与现有方法相比，我们的方法确实能够发现质量具有竞争力且结构更加灵活的模型。

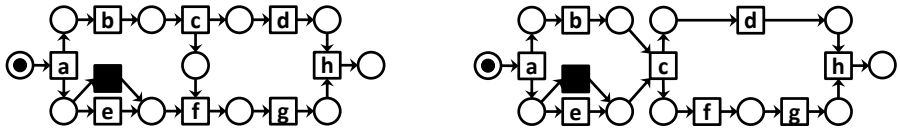
关键词： 流程发现 · 自由选择网 · 综合规则。

1 引言

流程发现旨在构建能够反映从信息系统中提取的给定事件日志行为的流程模型 [2]。由于这是一个非平凡问题，仍然存在许多挑战。在大多数情况下，并不存在唯一的“最佳模型”，因为四种模型质量指标之间存在权衡，即适应度、精确度、泛化度和简洁度 [2]。除了质量指标之外，人们还希望发现的模型具备某些属性。其中一项重要属性是成为无死锁工作流网，因为健全性能够确保不存在死锁、正确完成等情况 [1] ，并且这是许多关键自动化分析（例如一致性检查）的先决条件。另一项理想的结构属性是自由选择 [3]。在自由选择网中，选择与同步彼此分离。这使得发现的模型与许多流程建模语言之间能够轻松转换，例如业务流程建模标注（BPMN），

因为其中的等价构造（专用的分割连接器和合并连接器）能够自然地嵌入。此外，自由选择网已得到广泛研究，因此拥有大量理论支持 [10], ，这些理论提供了高效的分析技术。

虽然已有多种发现算法被提出，但只有少数算法能够提供此类保证。最先进的发现算法，例如归纳式挖掘器 (IM) [15] ，能够利用其表示偏置发现健全自由选择工作流网。然而，正因如此，发现出的模型被限制为块结构化。这限制了此类模型的表达能力，也就是说，只能发现健全自由选择工作流网的一个子集。例如，图 1a 展示了我们的方法[1]发现的一个健全自由选择工作流网（具有非块结构）。如图 1b 所示，IM 发现的模型无法表达相同的语言。



(a) 我们的方法发现的模型。相同的语言无法由归纳式挖掘器发现的模型表达，归纳式挖掘器在内部使用流程树。

(b) IM 使用以下内容发现的模型 (a) 中模型生成的日志。这两条在 c 之前的分支需要先同步然后才能执行 before d 。

图 1： 展示需要发现非块结构流程模型的示例。注意，(a) 中可能出现的轨迹 $\langle a, b, c, d, e, f, g, h\rangle$ 无法由 (b) 重放。

本文旨在发现具有非块结构的健全自由选择工作流网。受 [11,12], 中交互式流程发现方法的启发，我们开发了一种自动流程发现算法，通过综合规则 [10] 将活动逐步添加到现有网中。由于检查综合规则应用的可行性在计算上代价高昂，我们使用日志启发式来定位待添加活动在现有流程模型中最有可能的位置，而不是像 [11] 中那样评估综合规则的所有可能应用。此外，我们指出还需要一条额外规则，并扩展了 [12] 中引入的模式集。

遵循这些规则，可以保证我们的方法发现的流程模型是健全自由选择工作流网 [10,11]。此外，发现的模型不受块结构的限制。最后，重放适应度水平由用户设定的阈值予以保证。该方法已使用 Python 实现，并通过多个公开可用的真实事件日志进行了评估。评估结果表明，与最先进的发现算法相比，我们的方法能够发现质量具有竞争力的非块结构模型。

[1] 如定义 5 中所述，所提出的方法为开始和结束设置了专门的静默变迁。为便于比较，此处省略了它们。

本文其余部分安排如下：第 2 节回顾相关工作，第 3 节介绍必要的概念。第 4 节介绍该方法，第 5 节展示实验，第 6 节总结全文。

2 相关工作

本文不在此概述流程发现；更多详情请参阅[7,14] 。本节重点介绍能够保证健全性（和自由选择）性质的流程发现技术。 [6,8]等方法能够发现非块结构化模型，但无法保证这两种性质。Split Miner 虽然能够发现无死锁模型，但这些模型不一定健全 [8]。

归纳式挖掘器（IM）算法系列 [15] 利用流程树的表示偏置，保证发现的模型具有健全性和自由选择性质。按照设计，流程树表示一个健全的工作流网。它是一棵有根树，叶节点为活动，非叶节点为算子。层次化表示可以直接转换为佩特里网。然而，由于流程树只能表示那些可拆分为具有单一入口和出口的部分的流程模型 [15]，因此生成的模型仅限于块结构化模型。因此，流程树只能表示部分健全工作流网。对于基于流程树的方法，例如 Evolutionary Tree Miner（ETM） [9]以及近期开发的增量式流程发现方法 [16]，同样的论述也成立。

应用综合规则 [10],， [12,13,11] 中提出的交互式流程发现方法能够确保健全性和自由选择属性。 [12] 提出了一种半自动交互式工具 ProDiGy，用于推荐将活动添加到现有模型中的最佳方式。

我们的方法与 [12,13,11] 有几处不同。首先，我们采用自动化设置，因为活动的添加顺序是预先确定的，并且会根据模型质量选择对现有网的最佳修改。其次，我们使用日志启发式来确定添加新活动的最合适位置，而不是评估应用综合规则的所有可能性。此外，我们发现有必要制定一条新规则，因为通常需要通过结合约简规则和综合规则 [13] 来回迭代，才能发现所需模型。最后，我们扩展了模式集合并对其进行了形式化定义。

3 预备知识

我们将某个集合 A 上的所有序列的集合记为 A^*，将 A 的幂集记为 $\mathcal{P}(A)$，并将 A 上所有多重集的集合记为 $\mathcal{B}(A)$。对于某个多重集 $b \in \mathcal{B}(A)$， $b(a)$ 表示 $a \in A$ 在 b 中出现的次数。对于给定序列 $\sigma = \langle a_1, a_2, ..., a_n \rangle \in A^*$， $|\sigma| = n$ 是 σ 的长度，而 $dom(\sigma) = \{1, 2, ..., |\sigma|\}$ 是 σ 的定义域。 $\langle\rangle$ 是空序列。 $\sigma(i) = a_i$ 表示 σ 的第 i 个元素。给定序列 σ_1 和 σ_2， $\sigma_1 \cdot \sigma_2$ 表示二者的拼接。设 A 为一个集合， $X \subseteq A$ 为 A 的子集。对于 $\sigma \in A^*$ 和 $a \in A$,

我们递归地将 $\restriction_X \in A^* \to X^*$ 定义为投影函数，其中当 $\langle\rangle\restriction_X = \langle\rangle$ 时，$(\langle a\rangle \cdot \sigma)\restriction_X = \langle a\rangle \cdot \sigma\restriction_X$；当 $a \in X$ 时，$(\langle a\rangle \cdot \sigma)\restriction_X = \sigma\restriction_X$。例如，$a \notin X$ 时为 $\langle x, y, x\rangle\restriction_{\{x,z\}} = \langle x, x\rangle$。投影也可以应用于序列多重集，例如 $[\langle a, b, a\rangle^6, \langle a, b, c\rangle^6, \langle b, a, c\rangle^2]\restriction_{\{b,c\}} = [\langle b\rangle^6, \langle b, c\rangle^8]$。

定义 1（轨迹、日志）。轨迹 $\sigma \in \mathcal{U}_A^*$ 是活动序列，其中$\mathcal{U}_A$ 是活动全集。日志 $L \in \mathcal{B}(\mathcal{U}_A^*)$ 是轨迹多重集。

定义 2（日志属性）。设 $L \in \mathcal{B}(\mathcal{U}_A^*)$ 和 $a, b \in \mathcal{U}_A$。

– $\#(a, L) = \Sigma_{\sigma\in L}|\{i \in dom(\sigma)|\sigma(i) = a\}|$ 在 L中出现的次数。– $\#(a, b, L) = \Sigma_{\sigma\in L}|\{i \in dom(\sigma)\backslash\{|\sigma|\}|\sigma(i) = a \wedge \sigma(i+1) = b\}|$ 表示 L中从 a到 b 的直接后继关系数量。– $caus(a, b, L) = \begin{cases} \#(a,b,L) - \#(b,a,L) & \#(a,b,L) + \#(b,a,L) + 1 \text{ 如果} \\ a \neq b \ \#(a,b,L)\ \#(a,b,L) + 1 & \text{如果 } a = b \end{cases}$ 是因果关系（a, b）的强度。– $A_c^{pre}(a, L) = \{a_{pre} \in \mathcal{U}_A | caus(a_{pre}, a, L) \geq c\}$ 是 a的前置活动集合，由阈值 c确定。– $A_c^{fol}(a, L) = \{a_{fol} \in \mathcal{U}_A | caus(a, a_{fol}, L) \geq c\}$ 是 a 的后续活动集合，由阈值 c确定。– $A^s(L) = \{\sigma(1) \mid \sigma \in L \wedge \sigma \neq \langle\rangle\}$ 是 L中的起始活动集合。– $A^e(L) = \{\sigma(|\sigma|) \mid \sigma \in L \wedge \sigma \neq \langle\rangle\}$ 是 L中的结束活动集合。

定义 3（佩特里网、标记佩特里网）。佩特里网 $N = (P, T, F)$ 是一个元组，其中 P 是库所集合，T 是变迁集合，$P \cap T = \emptyset$，且$F \subseteq (P \times T) \cup (T \times P)$ 是弧集合。标记佩特里网 $N = (P, T, F, l)$ 是一个佩特里网 (P, T, F)，并带有一个标记函数 $l \in T \nrightarrow \mathcal{U}_A$，该函数将部分变迁映射到活动。若 t 不属于 l的定义域，则称 $t \in T$ 为不可见。

对于任意 $x \in P \cup T$, ${}^{\bullet}{}^{N}x = \{y | (y, x), \in F\}$ 表示输入节点集合，而$x^{N}{}^{\bullet} = \{y | (x, y), \in F\}$ 表示输出节点集合。如果上下文明确，则省略上标 N。该记法可推广到集合。对于任意 $X \subseteq P \cup T$, $\bullet X = \{y | \exists_{x\in X}(y, x), \in F\}$ 以及 $X\bullet = \{y | \exists_{x\in X}(x, y), \in F\}$。

定义 4（自由选择网）。设 $N = (P, T, F)$ 为一个佩特里网。N是自由选择网，当且仅当对于任意 $t_1, t_2 \in T$：$\bullet t_1 = \bullet t_2$ 或 $\bullet t_1 \cap \bullet t_2 = \emptyset$。

定义 *5*（工作流网（*WF*网） **[1,11]**）。设 $N = (P, T, F, l)$ 为一个标记佩特里网。$W = (P, T, F, l, p_s, p_e, \top, \bot)$ 是 *WF*网，当且仅当：*(1)* 它有一个专用源库所 $p_s \in P$：$\bullet p_s = \emptyset$，以及一个专用汇库所 $p_e \in P$：$p_e\bullet = \emptyset$；*(2)* $\top \in T$：$\bullet\top = \{p_s\} \wedge p_s\bullet = \{\top\}$ 且 $\bot \in T$：$\bot\bullet = \{p_e\} \wedge \bullet p_e = \{\bot\}$；*(3)* 每个节点 x 都位于从 p_s 到 p_e 的某条路径上，即 $\forall_{x\in P\cup T}(p_s, x) \in F^* \wedge (x, p_e) \in F^*$，其中 F^* 是 F 的自反传递闭包。

定义*6*（短路*WF*网 **[1]**）。设 $W = (P, T, F, l, p_s, p_e, \top, \bot)$ 是一个*WF*网。W的短路*WF*网记为 $SC(W)$，其通过 $SC(W)=(P, T\cup\{t'\}, F\cup\{(\bot, t'), (t', \top)\}, l, p_s, p_e, \top, \bot)$ 构造，其中 $t' \notin T$。

定义7（路径、初等路径）。佩特里网 $N=(P,T,F)$ 的一条路径是节点 $\rho=\langle x_1,x_2,...,x_n\rangle$ 的非空序列，使得 $(x_i,x_{i+1})\in F$对于$1\le i<n$成立。ρ 是初等路径，当且仅当 $x_i\neq x_j$ 对于 $1\le i<j\le n$ 成立。

定义8（关联矩阵 [10]）。设 $N=(P,T,F)$ 是一个佩特里网。其关联矩阵 $\mathbf{N}:(P\times T)\to\{-1,0,1\}$ 定义如下：

$$\mathbf{N}(p,t)=\begin{cases}0 & if\ ((p,t)\notin F\wedge(t,p)\notin F)\vee((p,t)\in F\wedge(t,p)\in F)\\ -1 & if\ (p,t)\in F\wedge(t,p)\notin F\\ 1 & if\ (p,t)\notin F\wedge(t,p)\in F\end{cases}$$

对于佩特里网 $N=(P,T,F)$ 及其对应的关联矩阵 $\mathbf{N}$，我们用$\mathbf{N}(p)$ 表示对应的 $p\in P$ 的行向量，用 $\mathbf{N}(t)$ 表示对应的 $t\in T$的列向量。

定义*9*（线性相关节点 **[10]**）。设 $N=(P,T,F)$ 是一个佩特里网。$\mathbb{Q}$ 是有理数集。若存在行向量 $\vec{v}:P\to\mathbb{Q}$，使得 $\vec{v}(p)=0$ 且 $\vec{v}\cdot\mathbf{N}=\mathbf{N}(p)$，则库所 p 线性相关。若存在列向量 $\vec{v}:T\to\mathbb{Q}$，使得 $\vec{v}(t)=0$ 且 $\vec{v}\cdot\mathbf{N}=\mathbf{N}(t)$，则变迁 t 线性相关。

定义10（综合规则 [10,11]）。设 W 和 W' 是两个自由选择工作流网，且 $SC(W)=(P,T,F,l,p_s,p_e,\top,\bot)$和 $SC(W')=(P',T',F',l',p_s,p_e,\top,\bot)$是相应的短路 *WF* 网：

– 线性相关库所规则 ψ_P：W'由 W使用 ψ_P导出，即，当 *(1)* $T'=T$，$P'=P\cup\{p\}$且 $p\notin P$在$\sim$ $\sim SC(W')$ 中线性相关时，$(W,W')\in\psi_P$；$F'=F\cup F$其中 $F\subseteq((\{p\}\times T)\cup(T\times\{p\}))$；*(2)* $SC(W')$ 中的每个虹吸都包含 p_s。– 线性相关变迁规则 ψ_T：W'由 W使用 ψ_T导出，即，当 $P'=P$，$T'=T\cup\{t\}$且 $t\notin T$在$\sim$ $\sim SC(W')$ 中线性相关，并且 $F'=F\cup F$时，$(W,W')\in\psi_T$，其中 $F\subseteq((P\times\{t\})\cup(\{t\}\times P))$，且 $\forall_{t\in T\cap T'}l(t)=l'(t)$。– 抽象规则 ψ_A：当 *(1)* 存在一个变迁集$R\subseteq T$和一个库所集 $S\subseteq P$，使得 $(R\times S\subseteq F)\wedge(R\times S\neq\emptyset)$；*(2)*$SC(W')$ 是通过添加一个额外库所 $p\notin P$和一个变迁$t\notin T$构造的，且满足 $P'=P\cup\{p\},T'=T\cup\{t\},F'=(F\backslash(R\times S))\cup((R\times\{p\})\cup(\{p\}\times\{t\})\cup(\{t\}\times S))$，并且 $\forall_{t\in T\cap T'}l(t)=l'(t)$ 时，$(W,W')\in\psi_A$。

应用三条合成规则（ψ_P、ψ_T、ψ_A）从健全自由选择工作流网 W 推导出 W'，可确保 W' 也具有健全性 [13,11]。WF网要实现健全性，必须满足三个性质：(1) 安全性：库所不能同时持有多个令牌；(2) 可完成性：始终能够到达仅汇库所被标记的标识；(3) 不存在死变迁。接下来，我们介绍初始网 [11]，并展示一些应用综合规则的示例。

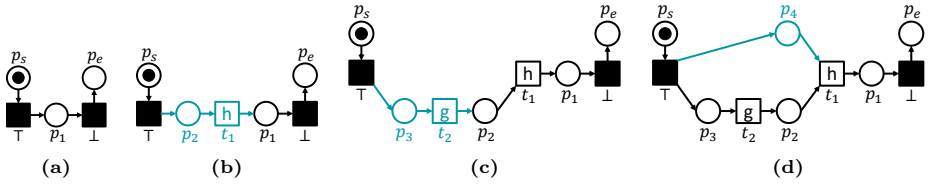


图3： 综合规则应用示例，起始于 (a) 初始网。(b) 使用 ψ_A、p_2 和 t_1，将它们与 $R = \{\top\}$ 和 $S = \{p_1\}$ 一起添加到初始网中。(c) 使用 ψ_A、p_3 和 t_2，将它们与 $R = \{\top\}$ 和 $S = \{p_2\}$ 一起添加到前一个网中。(d) 使用 ψ_p 添加 p_4，因为 p_4 是 p_3 和 p_2 的线性组合。

定义 11（初始网 [13]）。设 $W = (P, T, F, l, p_s, p_e, \top, \bot)$ 是一个自由选择*WF*网。若 $P = \{p_s, p_1, p_e\}$, $T = \{\top, \bot\}$, $F = \{(p_s, \top)、(\top, p_1)、(p_1, \bot)、(\bot, p_e)\}$, 则 W 是一个初始网。

初始网如图2a所示。显然，它是一个健全自由选择工作流网。从初始网出发，可以根据综合规则逐步添加更多节点。图2展示了从初始网出发应用综合规则的示例。

4 方法

介绍了必要概念后，我们现在可以介绍该方法了。我们先借助图3展示该方法的基本思路，然后再详细讨论各个步骤。该方法在内部会逐步向现有网添加一个新活动。图中展示了一次迭代。在每次迭代中，我们都有一个来自上一次迭代的现有模型，以及一个投影到迄今已添加活动和待添加活动上的日志。

我们首先根据日志启发式，确定最有可能添加新活动的位置。此步骤的结果是现有模型中的一个节点子集。随后将利用这些节点对搜索空间进行剪枝。然后，将预定义模式应用于现有网，以获得一组候选网。最后，根据适应度和精确度，从候选网中选择最佳网（下一个现有网）。需要注意的是，第一次迭代中的现有网由初始网（定义11）初始化。作为一个贯穿示例，考虑用于通过我们的方法发现图1a中佩特里网的相应日志：$L_s = [\langle a,b,c,d,f,g,h\rangle^{22}, \langle a,b,c,f,d,g,h\rangle^{14}, \langle a,e,b,c,d,f,g,h\rangle^{13}, \langle a,e,b,c,f,d,g,h\rangle^{13}, \langle a,e,b,c,f,g,d,h\rangle^{10}, \langle a,b,c,f,g,d,h\rangle^{10}, \langle a,b,e,c,d,f,g,h\rangle^{6}, \langle a,b,e,c,f,g,d,h\rangle^{3}, \langle a,b,e,c,f,d,g,h\rangle^{3}, \langle a,b,c,d,e,f,g,h\rangle^{2}, \langle a,b,c,e,d,f,g,h\rangle^{2}, \langle a,b,c,e,f,g,d,h\rangle^{1}, \langle a,b,c,e,f,d,g,h\rangle^{1}]$。图3中的实例展示了该贯穿示例的第3次迭代 L_s。在下面的小节中，我们将介绍每个步骤的详细内容。

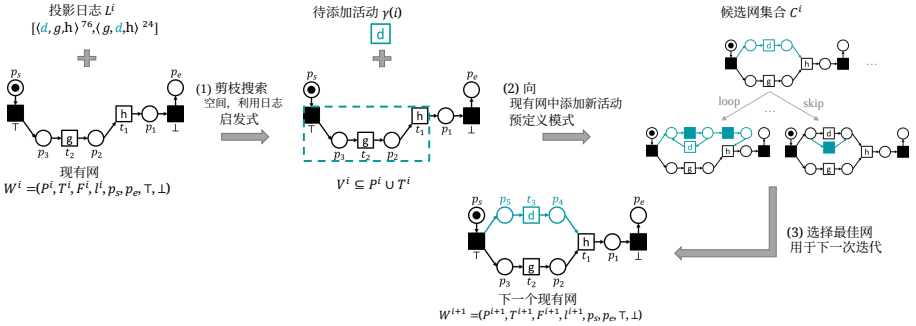


图3： 我们的方法进行单次迭代的示例。

4.1 添加活动的排序策略

在开始任何迭代之前，我们需要根据给定日志 L 确定添加活动的顺序。这一点很重要，因为发现的模型的质量通常取决于添加活动的顺序 [11]。此外，与搜索空间剪枝结合使用时，它会显著影响每次迭代的计算时间。在本文中，我们介绍两种排序策略。第一种相对直接。 L 中的活动仅按其频率排序。

定义 12（活动添加顺序，基于频率的排序）。令 $L \in \mathcal{B}(\mathcal{U}_A^*)$ 且 $A = \bigcup_{\sigma \in L}\{a \in \sigma\}$。$\gamma \in A^*$ 是L 的活动添加顺序，当且仅当 $\{a \in \gamma\} = A$ 且 $|\gamma| = |A|$。基于频率的排序是顺序$freq(L) = \gamma$，满足 γ 是活动添加顺序，且 $\forall_{1\le i<j\le|\gamma|} \#(\gamma(i)\ L) \ge \#(\gamma(j)\ L)$。

第二种排序策略类似于广度优先搜索（BFS）算法。它的优点在于，除了频率之外，还考虑了日志中活动之间的接近程度。为了说明这一排序策略，我们先定义一个子函数。

定义 *13*（直接前驱活动排序）。令 $L \in \mathcal{B}(\mathcal{U}_A^*)$ 且$a \in \mathcal{U}_A$。 $A=\{b \in \mathcal{U}_A | \#(b,a,L)>0\}$ 是 L 中至少出现一次直接前驱为 a 的活动集合，且 $\sigma \in A^*$。直接前驱活动排序是 $sortPreceded(a,L)=\sigma$，满足 $\{b \in \sigma\} = A$ 且 $|\sigma| = |A|$，并且 $\forall_{1\le i<j\le|\sigma|} \#(\sigma(i)\ a,L) \ge \#(\sigma(j)\ a,L)$。

函数 $sortPreceded$ 接收一个活动 a 和一个日志 L，并返回一个序列，其中包含a的直接前序活动 b，这些活动按 $\#(b,a,L)$ 的频率排序。最后，我们可以定义基于广度优先搜索的排序策略。

定义 14（基于广度优先搜索的排序）。令 $L \in \mathcal{B}(\mathcal{U}_A^*)$和 $A=\bigcup_{\sigma \in L}\{a \in \sigma\}$。基于广度优先搜索的排序定义为 $order_{BFS}(L)=\gamma$，其中γ是针对 L和 $\gamma=\gamma_1 \cdot \gamma_2 \cdot \ldots \cdot \gamma_{||\gamma||}$的活动添加顺序，对于每个 $1 \le j \le |\gamma|$,

$$\gamma_j = \begin{cases} order_{freq}(L\restriction_{A^e(L)}) & if\ j=1 \\ sortPreceded(\gamma(j-1),L)\restriction_{A\setminus\{\gamma(1),\gamma(2),\ldots,\gamma(j-1)\}} & otherwise \end{cases}$$

该函数首先根据结束活动 $A^e(L)$ 在日志中的频率对其进行排序。然后，它遍历序列 γ，并按照直接后继关系的频率对 $\gamma(j-1)$ 的前置活动进行排序。定义 14 中第二种情况下的投影函数会过滤掉已在 γ中的活动。

与基于频率的排序相比，基于广度优先搜索的排序会考虑活动的接近程度。这使我们能够添加彼此接近的活动。结合搜索空间剪枝的效果，预计基于广度优先搜索的排序所需的计算时间会更短。将函数$order_{BFS}$ 应用于我们的运行示例 L_s，得到的活动添加顺序为$\gamma = order_{BFS}(L_s)$
$= \langle h\rangle \cdot \langle g,d\rangle \cdot \langle f\rangle \cdot \langle c,e\rangle \cdot \langle\rangle \cdot \langle b\rangle \cdot \langle a\rangle \cdot \langle\rangle = \langle h,g,d,f,c,e,b,a\rangle$。$\gamma$ 随后用于确定活动的添加顺序。给定活动添加顺序 γ，我们将每次迭代 i 的工件定义如下。

定义15（投影日志）。设 $L \in \mathcal{B}(\mathcal{U}_A^*)$ 和 γ 是针对 L的活动添加顺序。第 i 次迭代中 L 的投影日志为 $L^i = L{\restriction}_{\{\gamma(1),\gamma(2),\ldots\gamma(i)\}}$。

例如，如图3所示，运行示例 L_s 在第3次迭代中的投影日志为 $L_s^3 = L_s{\restriction}_{\{h,g,d\}} = [\langle d,g,h\rangle^{76}, \langle g,d,h\rangle^{24}]$。待添加的活动表示为 $\gamma(i)$，即第3次迭代的 $\gamma(3) = d$。此外，我们将迭代 i对应的现有健全自由选择工作流网记为 W^i。注意，对于该运行示例，W_1、W_2和 W^3分别如图2a、2b和2c所示。

4.2 搜索空间剪枝

由于检查应用线性依赖规则 ψ_T, ψ_P是否可行的计算开销很大 [11]，，因此计算综合规则所有可能的应用并不切实际。此外，其中一些应用并不值得关注。例如，如图3所示，很明显，在投影日志中，待添加的活动 d 从未在h 之后发生。利用这一信息，我们可以直接排除那些允许活动 d 在 h之后执行的构造（综合规则的应用）。因此，在每次迭代 i 中，我们首先定位添加 $\gamma(i)$的最可能位置。这有助于将综合规则的应用限制在现有网 W^i中的一部分节点上，该节点集合记为 $V^i \subseteq P^i \cup T^i$。为此，我们首先在投影日志 L^i中确定 $\gamma(i)$的前序活动集合和后续活动集合，分别为 $A_c^{pre}(\gamma(i)\ L^i)$ 和 $A_c^{fol}(\gamma(i)\ L^i)$。回想一下，$c$ 是因果关系的阈值，可由用户作为输入指定。我们使用 $c = 0.9$ 作为默认值。然后，在 W^i中找出相应的带标签变迁。最后，V^i 是从前序变迁到后续变迁的初等路径上所有节点的集合。如果 $A_c^{pre}(\gamma(i)\ L)/A_c^{fol}(\gamma(i)\ L)$为空集，则使用 $\top/\perp$ 变迁。例如，在图3中，我们确定 $A_c^{pre}(d, L_s^3)=\emptyset$ 以及 $A_c^{fol}(d, L_s^3)=\{h\}$。因此，我们找出 $\{\top\}$中的每个节点与 $\{t_1\}$中的每个节点之间初等路径上的所有节点。由此，集合 $V^3 = \{\top, p_3, t_2, p_2, t_1\}$ 用于剪枝搜索空间，即限制综合规则的应用范围。

限制综合规则 对于抽象规则 ψ_A，这意味着用作应用ψ_A 前置条件的变迁集合 R 和库所集合 S 必须是 V的子集，即 $S \subseteq V \wedge R \subseteq V$。对于线性依赖规则$\psi_P$/$\psi_T$，新库所/变迁（$p'$/$t'$）不能与 V之外的任何节点通过弧相连。这样可以缩短计算时间，因为某些规则应用可以被排除，无需再检查其可行性。

4.3 模式

本节介绍用于将活动 $\gamma(i)$ 添加到现有自由选择工作流网 W^i 的模式。首先，我们说明为什么需要一条额外规则。

需要一条额外规则 已证明，任何健全自由选择工作流网都可以通过三条合成规则 ψ_A, ψ_P, ψ_T [10,11] 构造。然而，在用于发现流程模型时，由于现有结构的限制，往往无法推导出理想的模型。图 4 展示了一个例子。虽然可以在图 4a 中添加一个标记为a 的变迁，但由于没有规则允许这种转换，因此无法在图 4b 中推导出相同的网。一种可能的变通方法是按照 [13] 中的建议，结合约简规则和合成规则反复进行约简与合成。然而，随着现有网变得更加复杂，这种方法将难以追踪，因而不可行。

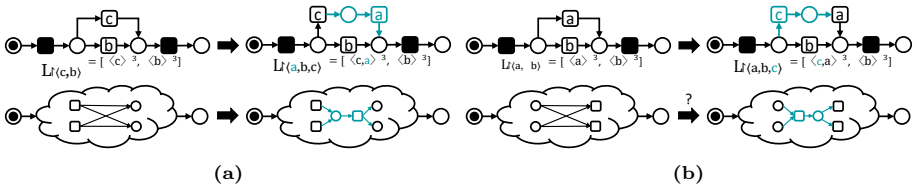


图4： 展示对偶抽象规则 ψ_D动机的示例。尽管 (a) 和 (b) 右侧所期望的网在语义上相同，现有的综合规则却只允许进行 (a) 中的变换。对于 (b) 中的变换，没有相应的规则。

我们观察到，在许多情况下（包括图4b中的示例），由于没有规则允许在一组库所 S 和一组完全连接的变迁 R 之间引入一个新的变迁 t 和一个新的库所 p，即 $S \times R \subseteq F$，因此无法构造所需的模型。因此，我们定义了对偶抽象规则，以允许这种构造。

定义16（对偶抽象规则 ψ_D）。设 $W = (P, T, F, l, p_s, p_e, \top, \perp)$和 $W' = (P', T', F', l', p_s, p_e, \top, \perp)$ 是两个自由选择工作流网。$(W, W') \in \psi_D$，如果（*1*）存在一个库所集合 $S \subseteq P$ 和一个变迁集合 $R \subseteq T$，使得 $S \times R \subseteq F \wedge S \times R \neq \emptyset$。（*2*） W' 通过添加一个额外的

变迁 $t \notin T$和一个库所 $p \notin P$构造而成，使得 $P' = P \cup \{p\}, T' = T \cup \{t\}, F' = (F \setminus (S \times R)) \cup ((S \times \{t\}) \cup (\{t\} \times \{p\}) \cup (\{p\} \times R))$，并且 $\forall_{t \in T \cap T'} l(t) = l'(t)$。由于我们只关注健全自由选择工作流网，因此需要确保对偶抽象规则 ψ_D 保持健全性。

命题1（ψ_D保持健全性）。设 $W = (P, T, F, l, p_s, p_e, \top, \bot)$ 和 $W' = (P', T', F', l', p_s, p_e, \top, \bot)$ 是自由选择工作流网，且 W' 是使用 ψ_D 从 W 推导而来，即 $(W, W') \in \psi_D$。那么，如果 W 是健全的，则 W' 也是健全的。

证明。设 $t' \in T' \setminus T$ 和 $p' \in P' \setminus P$ 分别是 W' 中新增的变迁和库所。设 $R = p'\bullet$ 和 $S = \bullet t'$。新网 W' 仅在两种情况下是自由选择的：要么 $S = \overset{W}{\bullet} R$，要么 $R = S \overset{W}{\bullet}$。在这两种情况下，$(W, [p_s])$ 中任何不需要发射 $t_R \in R$ 的可达标识，在 $(W', [p_s])$ 中仍然可达。此外，$(W, [p_s])$ 中需要发射 $t_R \in R$ 的可达标识，可以在 $(W', [p_s])$ 中到达，因为只需在相应发射序列中、t_R 之前的某处添加 t' 即可。因此，显然，如果 W 也健全，那么 W' 满足健全性的三个条件。□

接下来，我们扩展线性相关库所规则 ψ_P。由于我们的目标是向现有的带标签的自由选择工作流网 W^i 中添加一个标记为 $\gamma(i)$ 的变迁，仅通过 ψ_P 添加库所 p' 并不足够。因此，在我们的方法中，应用 ψ_P 后总是紧接着直接应用抽象规则 ψ_A，以纳入一个变迁。ψ_A 应用于新增库所 p' 与其前集 $\bullet p'$ 之间。根据定义，$\bullet p'$ 中的每个变迁都与 $\{p'\}$ 中的每个库所相连，因此满足 ψ_A 的前置条件，这使得上述操作成为可能。图5展示了一个例子：在通过 ψ_P 添加 p_4 后，p_5 和 t_3 由 ψ_A 直接添加。更准确地说，我们定义扩展规则 ψ'_P，用于描述该模式。

定义17（扩展线性相关库所规则 ψ'_P）。**设 $W=(P,T,F,l,p_s,p_e,\top,\bot)$ 和 $W''=(P'',T'',F'',l'',p_s,p_e,\top,\bot)$ 为自由选择工作流网。若 (1) $\exists_{W'=(P',T',F',l',p_s,p_e,\top,\bot)}(W,W') \in \psi_P \wedge (W',W'') \in \psi_A$，且 (2) $\exists!_{p^* \in P''}(\{p^*\}=P' \setminus P) \wedge (((T'' \setminus T') \times \{p^*\}) \subset F'') \wedge ((T' \times \{p^*\}) \not\subset F'')$，则 $(W,W'') \in \psi'_P$。**

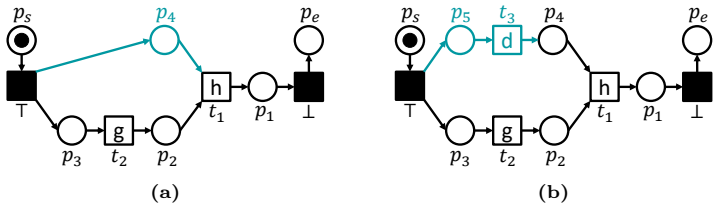


图5： (a) ψ_P 添加了一个库所 p_4。(b) 由于 $\bullet p_4$ 中的每个变迁都与 $\{p_4\}$中的每个库所之间都有一条弧，因此可以直接应用 ψ_A 来添加 p_5 和 t_3。

然后，我们定义通过规则 ψ_A、ψ'_P、ψ_T、ψ_D 各应用一次所构造的所有网的集合。

定义18（基础候选集）。设 $W=(P,T,F,l,p_s,p_e,\top,\bot)$、$W'=(P',T',F',l',p_s,p_e,\top,\bot)$ 为自由选择工作流网。设 $X=(P'\cup T')\setminus(P\cup T)$、$V\subseteq P\cup T$、$V'=(P\cup T)\setminus V$，并设 $a\in\mathcal{U}_A$ 为一个活动标签。基础候选集为 $base(W,V,a)=\{W'|((W,W')\in(\psi_A\cup\psi_T\cup\psi'_P\cup\psi_D))\wedge(\nexists_{x\in X}((\{x\}\times V')\cup(V'\times\{x\}))\subseteq F')\wedge(l'=l\cup((T'\setminus T)\times\{a\}))\}$。

基础候选集 C^i_{base}=基$(W^i,\ V^i,\ \gamma(i))$ 由通过对规则 ψ_A，ψ'_P，ψ_T，ψ_D 各应用一次所构造的网组成，这些规则用于向 W^i 添加一个标签为 $\gamma(i)$ 的变迁，同时考虑关于 V^i 的约束。

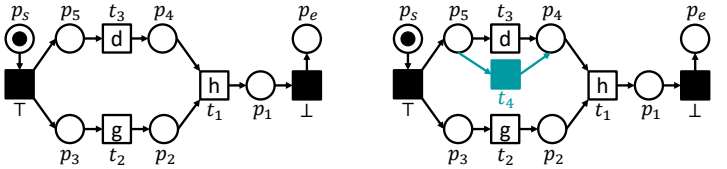
接下来，我们介绍三种模式，它们可以使变迁可跳过、处于严格循环中，或处于可选（τ）循环中。严格循环中的变迁意味着必须执行该变迁，否则它就是一个可选循环。

定义19（模式构建函数）。设 $W=(P,T,F,l,p_s,p_e,\top,\bot)$和 $W'=(P',T',F',l',p_s,p_e,\top,\bot)$ 是两个自由选择工作流网。设 $a\in\mathcal{U}_A$ 是一个活动标签，$t_a\in T$：$l(t_a)=a$ 是 W 中对应的变迁。我们将三个模式构建函数[2] 定义如下：

- $skip(W,a)=W'$ *such that*
  - $(W,W')\in\psi_T$
  - $F'=F\cup(\{t'\}\times t_a\bullet)\cup(\bullet t_a\times\{t'\})$ *(where* $t'\in T'\setminus T$*)*
  - $l'=l$ *(*t' *is a silent transition)*
- 循环$_s(LW,a)$ 由两种情况定义：
  1. *if* $\nexists_{t^*\in((t_a\bullet)\bullet)}(|\bullet t^*|>1)\wedge(\bullet t^*\setminus t_a\bullet\neq\emptyset)$*, then* $loop_s(W,a)=W'$ *such that*
     - $(W,W')\in\psi_T$
     - $F'=F\cup(t_a\bullet\times\{t'\})\cup(\{t'\}\times\bullet t_a)$ *(where* $t'\in T'\setminus T$*)*
     - $l'=l$ *(*t' *is a silent transition)*
  2. 否则，返回循环$_s(W',a)$，使得
     - $(W,W')\in\psi_A$
     - $((\{t_a\}\times(P'\setminus P))\in F')\wedge((\{t_a\}\times P)\notin F')$
     - $l'=l$
- 循环$_\tau(W,a)$ 由两种情况定义：
  1. *if* $\nexists_{t^*\in((t_a\bullet)\bullet)}(|\bullet t^*|>1)\wedge(\bullet t^*\setminus t_a\bullet\neq\emptyset)$*, then* $loop_\tau(W,a)=W'$ *such that*
     - $(W,W')\in\psi_T$
     - $F'=F\cup(t_a\bullet\times\{t'\})\cup(\{t'\}\times\bullet t_a)$ *(where* $t'\in T'\setminus T$*)*
     - $l'=(l\setminus\{(t_a,a)\})\cup\{(t',a)\}$ *(the labels of* t_a *and* t' *are swapped)*
  2. *otherwise, return* $loop_\tau(W,a)$ *such that*
     - $(W,W')\in\psi_A$
     - $((\{t_a\}\times(P'\setminus P))\in F')\wedge((\{t_a\}\times P)\notin F')$
     - $l'=l$

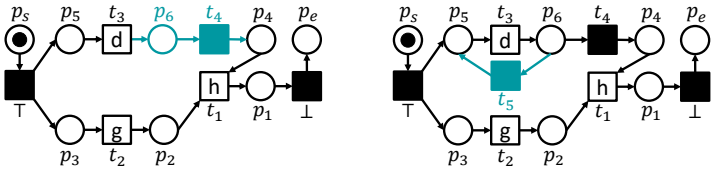
循环函数的第二种情况是为了保持自由选择性质。为了使用运行示例说明这些思想，考虑图 6a 所示的网作为输入网，W 和 t_3（标记为 d）是我们将应用这些函数以推导模式的变迁。图 6b 展示了该函数

[2] 定义 19 中使用的输入/输出节点记法（•）指的是输入网 W。为便于阅读，我们省略上标。



(a) 网 (W) t_3 （由 d 标记）是目标变迁。

(b) $skip(W, d)$ 添加了一个静默变迁—变迁 t_4，使 t_3 可跳过。



(c) 一个中间网 W'（位于（a）和（d）之间）构造以保持自由选择性质。

(d) the 所得的 net of $loop_s(W, d)$，其使 t_3 在循环中。

图 6： 展示如何应用这些函数来推导模式的示例。

skip(W, d)仅添加了一个静默变迁 t_4，其连接方式与 t_3 到W相同。图 6c 和 6d 展示了 loop$_s(W, d)$ 的应用，并说明了循环函数需要分为两种情况的原因。如图 6c 所示，应用了 loop$_s$ 的第二种情况，因为存在一个变迁 $t_1 \in (t_3 \bullet\bullet)$，其前集包含多个库所（$|\bullet\, t_1| > 1$和 $\bullet t_1 \backslash t_3 \bullet \neq \emptyset$）。因此，首先通过添加 p_6和 t_4构造 W'（图 6c）。随后，该函数返回 loop$_s(W', d)$。现在应应用第一种情况。在这种情况下，将添加 t_5，其连接与 t_3相反。如前所述，循环函数的第二种情况有助于保持自由选择性质。设想这样一个网：在图 6a 的网中添加 t'，并使用连接（p_4、t'）和（t'、p_5）。这样的网会使 t_3 处于循环中，但不再是自由选择网。loop$_s$ 和 loop$_\tau$ 的构造几乎相同，区别在于 t_3 和静默变迁t_5 的标签互换了。

最后，为了得到候选网集合 C^i，我们将三个模式构建函数应用于每个网 $W \in C^i_{base}$。请注意，图 6 中的所有网都是 C^3 的元素。

4.4 选择与回退

选择 在最后一步中，我们从候选集 C^i 中选择下一个现有网 W^{i+1}，该候选集由投影日志 L^i 评估得出。选择以逐步方式进行。我们首先尝试过滤掉未达到用户定义的重放适应度阈值 θ 的候选项，然后根据 F1 分数从其余候选项中选择最佳网。F1 分数计算为适应度和精确度的调和平均数。我们使用基于对齐的适应度 [4] 和精确度 [5]。

回退 如果 C^i 中没有任何网达到适应度阈值 θ，我们将采用回退策略。具体做法是返回第 2 步，在该步骤中，将 $\gamma(i)$ 添加到 $W^i = (P^i、T^i、F^i、l^i、p_s、p_e、\top、\perp)$，但不受 V^i 的约束。这也可以看作将 $V^i = P^i \cup T^i$ 设置为该值。在这种情况下，新的库所 p' 及其弧 $\{(\top、p')、(p'、\perp)\}$ 始终可以由 ψ_P 添加，因为 p 与 p_s 和 p_e 线性相关。然后，可以应用模式构建函数，以确保在每次迭代中都达到适应度阈值 θ。

5 评估

本节介绍用于评估我们方法的实验。本文提出的方法使用 Python 和 PM4Py[3] 实现，可在此处访问[4]。如前所述，该算法以日志和三个参数作为输入，其中包括两个阈值 θ, c以及排序策略的类型。我们使用此实现开展了三项实验，以回答以下问题：(1) 预定义模式的效果如何？(2) 排序策略对模型质量和执行时间有何影响？(3) 非块结构能否提升模型质量？

5.1 实验设置

数据集：我们使用了四个公开可用的真实事件日志，分别为 BPI2017[5],helpdesk[6], hospitalBilling[7], 和 traffic[8] 。BPI2017 使用事件前缀拆分为两个子日志：BPI2017A 和 BPI2017O。为聚焦于主流行为，我们对日志进行过滤，仅保留至少包含 95% 案例的部分。

实验1（模式的有效性）：第一个实验旨在评估预定义模式的有效性。由于我们的方法基于[11], ，因此可以通过将我们的方法生成的中间模型的质量与 ProDiGy [12], 生成的模型进行比较来评估；ProDiGy 采用了类似的设置。为开展实验，我们在每个步骤中都遵循 ProDiGy 的首选建议，以生成中间模型，并将其质量与我们的方法生成的模型进行比较。我们使用每次迭代的投影日志，评估模型添加额外活动后的质量。为确保比较公平，我们强制我们的方法按照与 ProDiGy 相同的顺序添加活动。

实验2（排序策略与搜索空间剪枝的影响）：向日志中添加活动的顺序对我们的方法至关重要，因为模型质量在很大程度上取决于顺序 [11]。此外，顺序会影响搜索空间剪枝，从而影响执行时间。因此，我们希望研究排序策略对模型质量和执行时间的影响。为设置实验，我们将该方法应用于

[3] https://pm4py.fit.fraunhofer.de/ [4]
https://github.com/tsunghao-huang/synthesisRulesMiner [5]
https://doi.org/10.4121/uuid:3926db30-f712-4394-aebc-75976070e91f [6]
https://doi.org/10.4121/uuid:0c60edf1-6f83-4e75-9367-4c63b3e9d5bb [7]
https://doi.org/10 741 8 4121 76 46 83 9304798 1 94 94 . /uuid: c b -c - -a c - be
dfeb https://doi.org/10.4121/uuid:270fd440-1057-4fb9-89a9-b699b47990f5

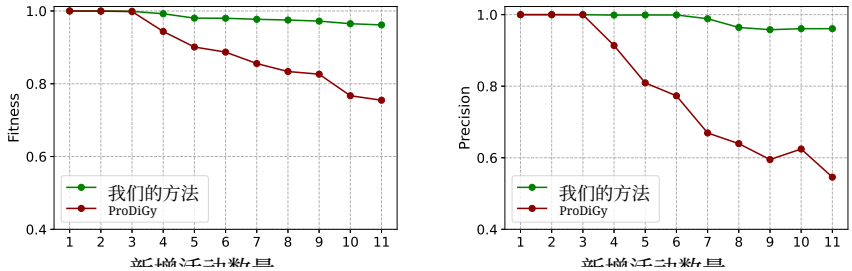
五个事件日志，使用两种不同的排序策略，同时将另外两个参数保持为相同的值。我们从适应度、精确度和 F1 分数三个方面评估模型质量。此外，我们还跟踪缩减节点的比例，该比例通过 $\frac{|V^i|}{|P^i \cup T^i|}$ 计算得出。这使我们能够了解搜索空间剪枝的效果。

实验 3（非块结构的影响）： 在本实验中，我们将我们的方法与最先进的方法进行比较：InductiveMiner - Infrequent (IMf)[15]。由于 IMf 发现的模型也保证是健全自由选择工作流网，因此将我们的方法与 IMf 进行比较，可以让我们了解我们的方法所发现的非块结构能否使模型受益。对于每个事件日志，我们使用五个不同的过滤阈值 ($[0.1, 0.2, 0.3, 0.4, 0.5]$) 运行 IMf，并选择最佳模型（按 F1 分数评定），将其质量与实验 2 中我们的方法发现的模型进行比较。

在所有实验中，我们使用基于对齐的方法计算适应度 [4]和精确度 [5]。我们还将 F1 分数计算为适应度与精确度的调和平均值。

5.2 结果

模式的有效性 图 7 展示了比较结果。



(a) 适应度 (b) 精确度

图 7： 模式有效性的适应度和精确度比较结果

适应度和精确度是五个事件日志的平均值。从图中可以看出，对于最初添加的三个活动，两种方法都能很好地捕捉行为。向模型中添加更多活动时，与 ProDiGy 相比，我们的方法在适应度和精确度方面始终具有更高的数值。有人可能认为，这是预料之中的，因为我们扩展了 ProDiGy 所使用的模式集合。不过请注意，ProDiGy 会评估每种综合规则的所有可能应用，而我们只根据日志启发式方法关注部分节点。我们的方法需要在最优解与时间之间权衡。尽管如此，结果表明，扩展后的模式使我们能够发现质量更高的模型，相较于现有方法 ProDiGy 更胜一筹，同时还能限制搜索空间。

排序策略与搜索空间剪枝的影响 表 1 展示了实验 2 和实验 3 的结果。我们发现，对于五个日志中的四个，基于广度优先搜索的排序策略优于基于频率的策略（就 F1 分数和时间而言）。我们进一步探究基于广度优先搜索的排序为何能缩短执行时间。如图 8 所示，结果表明，在发现过程的后期，基于广度优先搜索的排序策略在缩减搜索空间方面更有效（$\frac{|V^i|}{|P^i \cup T^i|}$ 更低）。

随着模型规模的增大，检查应用线性相关库所或变迁规则的前置条件会变得更加耗时。在大多数情况下，在后期更有效地缩小搜索空间更有利于减少执行时间。基于广度优先搜索的排序通过考虑流程中各活动的接近程度来实现这一点。在这种情况下，彼此更接近的活动会优先添加，因此基于广度优先搜索的排序更有可能专注于较小的子集

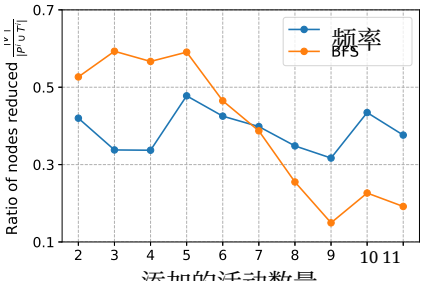


图 8： 两种排序策略在每次迭代中待考虑节点比例的比较。

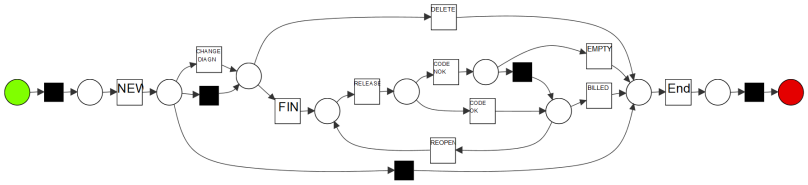
在剪枝搜索空间时，现有网中的节点数量与基于频率的方法相比。

表1： 排序策略的效果及与 IMf 的比较结果

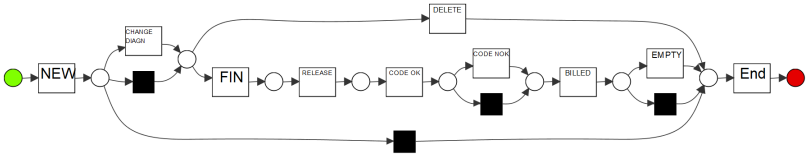
Log	矿工	排序策略	IMf 过滤器	适应度	精度	F1	时间（秒）
BPI2017A	我们的方法	频率	-	0.970	0.947	0.958	734
	ours	BFS	-	0.989	0.935	0.961	342
	IMf	-	0.2	0.999	0.936	0.967	10
BPI2017O	我们的方法	频率	-	0.994	0.962	0.978	560
	ours	BFS	-	0.989	1.000	0.994	240
	IMf	-	0.2	0.997	0.907	0.950	7
helpdesk	我们的方法	频率	-	0.972	0.984	0.977	54
	ours	BFS	-	0.981	0.976	0.978	44
	IMf	-	0.2	0.967	0.950	0.958	1
医院账单	我们的方法	频率	-	0.961	0.810	0.879	567
	ours	BFS	-	0.989	0.935	0.961	407
	IMf	-	0.2	0.982	0.906	0.943	45
traffic	我们的方法	频率	-	0.960	0.930	0.945	321
	ours	BFS	-	0.964	0.720	0.825	427
	IMf	-	0.4	0.904	0.720	0.801	28

非块结构的影响 表1显示，与 IMf 相比，我们的方法发现的模型在五个日志中的四个日志上取得了更高的 F1分数。请注意，我们的方法发现的模型的适应度值

均高于设定的阈值 0。95。总体而言，IMf 倾向于发现适应度值较高的模型，而我们的方法发现的模型精确度较高。在 IMf 中，可以使用过滤阈值来平衡适应度和精确度。我们的做法也一样，用户可以设定较低的适应度阈值，以纳入更多适应度较低但精确度较高的候选网。图9展示了两种方法针对 hospitalBilling 日志发现的模型。尽管图9a的整体结构与图9b中的对应结构相似，但我们的方法在流程后期发现了非块结构。这种结构无法通过 IMf 建模。结果表明，我们的方法能够发现具有非块结构的健全自由选择工作流网，并达到与最先进算法相当的模型质量。



(a) 使用我们的方法发现的模型。由于结构更加灵活，在$CODE\ NOK$ 之后可以执行$EMPTY$、$BILLED$或$REOPEN$ ，而在$CODE\ OK$之后只能执行$BILLED$ 或$REOPEN$ 。IMf 无法发现这种结构。



(b) 使用 IMf 发现的模型。请注意，活动$REOPEN$ 被 IMf 的过滤器删除了。

图 9： 我们的方法和 IMf 为 hospitalBilling 日志发现的模型。

6 结论与未来工作

本文提出了一种发现算法，旨在发现具有非块结构的可靠自由选择工作流网。该算法利用合成规则，通过添加具有预定义模式的活动来逐步发现模型，并保证所发现的模型可靠且具有自由选择性。此外，通过用户定义的阈值，可以保证一定程度的重放适应度。

我们已实现该方法，并使用多个真实事件日志对其进行了评估。结果表明，与同样依赖综合规则的现有方法 [12]，相比，我们的方法发现的流程模型具有更高的模型质量（重放适应度和精确度均更高）。此外，与当前最先进的方法 Inductive Miner - Infrequent 相比，我们的方法也能达到具有竞争力的模型质量。未来，我们计划探索更先进的排序策略，并研究这些策略对模型质量和计算时间的影响。另一个方向是进一步加快该方法的速度，因为执行时间过长显然是其一项局限。我们可以通过进一步利用基于日志的启发式方法来实现这一点。

致谢。我们感谢亚历山大·冯·洪堡基金会对我们的研究提供支持。

参考文献

1. van der Aalst, W.M.P.：佩特里网在工作流管理中的应用。J. Circuits Syst. Comput. **8**(1), 21–66 (1998) 2. van der Aalst, W.M.P.：流程挖掘——行动中的数据科学（第二版）。Springer (2016) 3. van der Aalst, W.M.P.：使用自由选择网进行流程挖掘和业务流程管理。载于：FedCSIS 2021。第 25 卷，第 9–15 页 (2021) 4. van der Aalst, W.M.P., Adriansyah, A., van Dongen, B.F.：在流程模型上重放历史，以进行符合性检查和性能分析。WIREs Data Mining Knowl. Discov. **2**(2), 182–192 (2012) 5. Adriansyah, A., Munoz-Gama, J., Carmona, J., van Dongen, B.F., van der Aalst, W.M.P.：衡量建模行为的精确度。Inf. Syst. E Bus. Manag. **13**(1), 37–67 (2015) 6. Augusto, A., Conforti, R., Dumas, M., Rosa, M.L., Bruno, G.：从事件日志中自动发现结构化流程模型：发现与结构化方法。Data Knowl. Eng. **117**, 373–392 (2018) 7. Augusto, A., Conforti, R., Dumas, M., Rosa, M.L., Maggi, F.M., Marrella, A., Mecella, M., Soo, A.：从事件日志中自动发现流程模型：综述与基准测试。IEEE Trans. Knowl. Data Eng. **31**(4), 686–705 (2019) 8. Augusto, A., Conforti, R., Dumas, M., Rosa, M.L., Polyvyanyy, A.：Split Miner：从事件日志中自动发现准确且简单的业务流程模型。Knowl. Inf. Syst. **59**(2), 251–284 (2019) 9. Buijs, J.C.A.M., van Dongen, B.F., van der Aalst, W.M.P.：用于发现流程树的遗传算法。载于：CEC 2012。第 1–8 页。IEEE (2012) 10. Desel, J., Esparza, J.：自由选择佩特里网。第 40 号，剑桥大学出版社 (1995) 11. Dixit, P.M.：交互式流程挖掘。博士论文，埃因霍温理工大学 (2019) 12. Dixit, P.M., Buijs, J.C.A.M., van der Aalst, W.M.P.：ProDiGy：人在环路中的流程发现。载于：RCIS 2018。第 1–12 页。IEEE (2018) 13. Dixit, P.M., Verbeek, H.M.W., Buijs, J.C.A.M., van der Aalst, W.M.P.：交互式数据驱动流程模型构建。载于：ER 2018。第 11157 卷，第 251–265 页。Springer (2018) 14. van Dongen, B.F., de Medeiros, A.K.A., Wen, L.：流程挖掘：佩特里网发现算法概述与展望。Trans. Petri Nets Other Model. Concurr.**2**, 225–242 (2009) 15. Leemans, S.J.J., Fahland, D., van der Aalst, W.M.P.：可扩展的流程发现与符合性检查。Softw. Syst. Model. **17**(2), 599–631 (2018) 16. Schuster, D., van Zelst, S.J., van der Aalst, W.M.P.：分层流程模型的增量式发现。载于：RCIS 2020。第 385 卷，第 417–433 页。Springer (2020)