

基于加权流关系的流程相似性算法

段瑞,方欢,詹悦

(安徽理工大学数学与大数据学院,安徽淮南 232001)

摘要: 为了提高从企业模型库中检索模型的效率,提出一种基于加权流关系的相似性算法,记作 WF. 首先,本文设计加权流关系集的生成算法,生成算法通过广度优先搜索方法遍历除输出库所外的所有节点及流关系,为每个流关系唯一分配权重,生成加权流关系集;其次,WF 算法基于加权流关系集计算出加权变迁集,并依据 WF 相似性定义计算相似性;最后,通过实验及算法的时间复杂度分析比较表明:算法是正确且唯一确定结果的,且 WF 算法具有良好的性能.

关键词: Petri 网; 加权流关系; 加权变迁; 流程相似性

中图分类号: TP301 **文献标识码:** A **文章编号:** 0372-2112 (2019)12-2596-06

电子学报 URL: <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2019.12.020

Process Similarity Algorithm Based on Weighted Flow Relationship

DUAN Rui, FANG Huan, ZHAN Yue

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

Abstract: In order to improve the efficiency of retrieving models from enterprise model library, a similarity algorithm based on weighted flow relationship is proposed, regarded as WF. Firstly, this paper designs a weighted flow relationship set generation algorithm. The generation algorithm traverses all nodes and flow relationships except the output place by breadth-first search method. The weight is assigned to each flow relationship to generate a weighted flow relationship set. Secondly, the WF algorithm calculates the weighted transition set based on the weighted flow relationship set, and calculates the similarity according to the WF similarity definition. Finally, through experimental and analyzing the time complexity of the algorithm results show that the proposed algorithm is correct and unique to determine the result and the algorithm has good performance.

Key words: Petri nets; weighted flow relationship; weighted transition; process similarity

1 引言

目前的流程相似性度量主要从三个角度进行^[1,2]:

①基于标签映射的相似性度量:这是最直观的流程相似性度量方法,该类算法关注单个流程模型的元素标签;②基于模型结构的相似性度量:此类度量算法从模型的拓扑结构出发,关注模型的元素集合及它们之间的行为关系;③基于行为语义的相似性度量:关注流程模型的执行语义,即行为特性.

近年来,科研工作者设计了多种流程相似性度量算法,基于可覆盖图编辑距离的行为相似性度量算法^[3]通过定义变迁标签图来表示过程模型的行为,设

计了变迁标签图节点匹配的最优方案,但该算法用到的可覆盖图的构造是指数级的空间复杂度.基于触发序列集合的算法(PTS++)^[4]通过定义完整触发序列表示模型行为,利用剪枝策略实现触发序列集合间的映射,该算法的基础是主序列算法(PTS)^[5],解决了PTS算法在处理循环结构上存在的缺陷,但完整触发序列集合由可覆盖树获得且受循环结构的影响,受到空间的局限.查海平等人通过考察流程变迁的两两紧邻关系,提出了TAR算法^[6],殷明等人提出一种改进的TAR算法,称作TAR++算法^[7],该算法创造性的为TAR增加了重要性系数.Weidlich等人提出了基于行为轮廓(BP)的相似性度量方法^[8],汪抒浩等人通过构造

收稿日期:2018-12-09;修回日期:2019-02-25;责任编辑:覃怀银

基金项目:国家自然科学基金(No. 61472003, No. 61402011, No. 61572035);安徽省自然科学基金(No. 1608085QF149);安徽省高校优秀青年人才基金(No. gxyqZD2018038);安徽省博士后基金(No. 2018B288)

任务最短跟随距离矩阵度量流程相似性^[9],并设计对应算法,称作 SSDT 算法.

为了解决上述相似性度量算法存在的问题,本文提出一种新的基于加权流关系的相似性算法(WF 算法),本文的主要贡献有:①扩展变迁的紧邻关系为流关系;②用广度优先搜索的方法为每个流关系合理加权,以表明不同流关系在相似性计算中的占比不同,极大地降低了算法的最坏时间复杂度;③不改动模型的前提下,WF 算法即满足所有的相似性性质;④WF 算法有个更高的上界,在度量相似性很高的两个流程模型时,更符合人们的直观认识.

2 WF 的计算

WF 算法分为两个步骤:第一步从流关系 F 中获取可以依据权重分配规则计算其权重的流关系,第二步为获取的每个流关系合理加权生成 WF.

2.1 权重分配规则

在一个 workflow 网中,定义由输入库所引出的流关系的权重之和为 1,若输入库所只有一个出边,则赋予权重 1,若存在多条出边,则均分权重 1. 输出库所类似,由此得出规则 1.

规则 1 输入库所出边的权重之和为 1 且均分 1,输出库所入边的权重之和为 1 且均分 1. 当输入或输出库所出边或入边只有一条时,该边的权重为 1.

规则 2 当变迁被引发时,变迁的出边必然发生,因此,变迁节点的出边权重等于入边权重,即一个变迁节点的所有入边和出边权重相等.

规则 3 除输入和输出库所外,库所节点的出边权重之和等于入边权重之和,且均分入边权重之和.

规则 1 和 2 由定义和给出的假设条件即可得出,下面用图论的相关知识说明规则 3 中“库所节点的出边权重之和等于入边权重之和”:把 workflow 网看作是一个流网络,库所看作流网络的结点,其中输入库所对应源点,输出库所对应汇点,剪掉变迁,变迁的入边出边相连,把权重看作边的非负容量. 则依据流量守恒,除源点和汇点外,每个结点的流出流量等于它的流入流量.

下面举例说明权重分配规则 1~3 的基本权重分配思想,图 1 是给定的 workflow 网,它的所有流关系均被分配了一个确定的权重,其中,根据规则 1 确定边 (i, t_1) 和 (t_7, o) 的权重,根据规则 3 确定库所节点 s_1 的出边权重,根据规则 2 确定变迁节点 t_4 的出边权重,具体权重数值如图 1 中的边标注.

2.2 WF 生成算法

分析 WF 生成算法:算法维护一个先进先出队列 Q ,并为每个结点增加一个颜色属性,只为该属性赋予灰色,用以辨别该结点是否已经被访问过. 算法第 1 行初始化

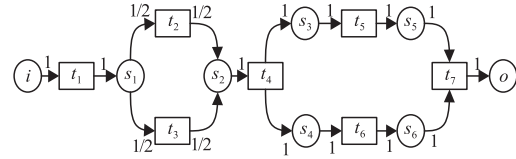


图1 工作流网的流关系分配权重示意图

WF, 算法第 2 行创建一个先进先出队列 Q , 算法第 3 行把输入库所压入到队列 Q 中. 算法第 4~18 行用广度优先搜索方法遍历流程模型中所有的节点(输出库所除外)和边, 其中第 5 行执行出队操作, 记为 u , 并从队列中删除 u , 第 6~18 行对每个从 u 可以直接到达的 v , 判断流关系 (u, v) 是否在 WF 中, 不在则依据权重分配规则计算权重, 并入到 WF 中. 算法第 19 行依据权重分配规则, 求出 WF 中的未知权重, 并整理得到最终的 WF.

算法 1 生成 workflow 网的加权流关系集

输入: workflow 网 WFN

输出: 加权流关系集

1. $WF = \varnothing$
2. 创建一个先进先出队列 Q
3. ENQUEUE(Q, i)
4. while $Q \neq \varnothing$
5. $u =$ DEQUEUE(Q)
6. for each v satisfy: $(u, v) \in F$
7. if $(u, v) \notin WF$
8. 依据权重分配规则计算边 (u, v) 的权重, 记作 W
9. $WF = WF \cup \{W(u, v)\}$
10. if v . color = \varnothing and $v \neq o$
11. v . color = gray
12. ENQUEUE(Q, v)
13. if $(u, v) \notin WF$ and 无法计算边 (u, v) 的权重
14. 设边 (u, v) 的权重为 $x(u, v)$
15. $WF = WF \cup \{x(u, v)\}$
16. if v . color = \varnothing and $v \neq o$
17. v . color = gray
18. ENQUEUE(Q, v)
19. 通过列方程组计算 WF 中的未知权重, 得到最终的 WF

图 2 为 WF 生成算法的概括步骤图, 图 2(a) 到图 2(b) 的过程: 计算 (s_4, t_6) 的权重, 由于 s_4 的入边 (t_7, s_4) 未知, 设 (s_4, t_6) 的权重为 x ; 图 2(b) 到图 2(c) 的过程: 访问到输出库所时, 不会把输入库所压入到队列中; 图 2(c) 到图 2(d) 的过程: 得出 (t_7, s_4) 的权重为 $x/2$, 则有一元一次方程 $x/2 + 1/2 = x$, 求解得 $x = 1$. 图 2(d) 到图 2(e) 的过程: 由上一步已求出所有未知数, 得到正确且唯一的加权流关系集 WF.

图 2(a) 的生成过程: 初始化 $WF = \varnothing$; 创建一个先进先出队列 $Q = \varnothing$; 把输入库所 i 压入到队列中, 此时 $Q = i$; 第一次循环, 取 Q 的队头元素记为 u , 此时 $u = i$;

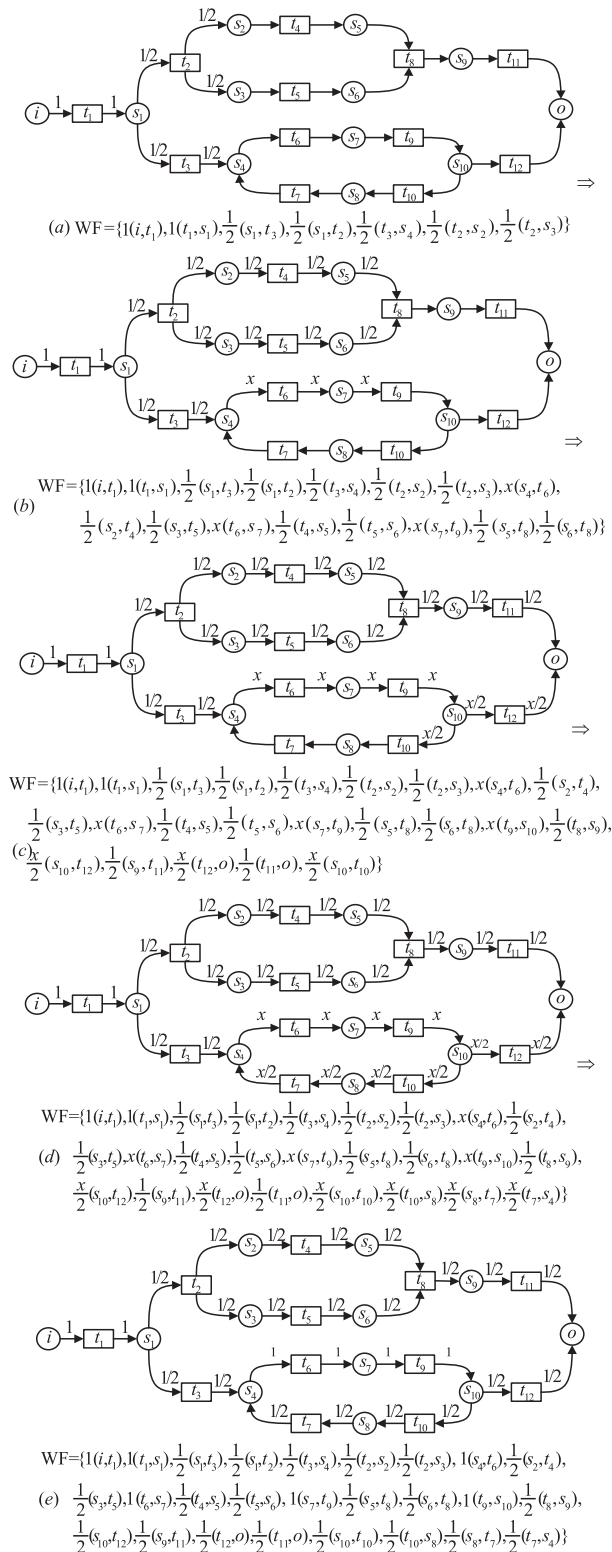


图2 WF生成算法生成WF的过程

对每个满足 $(u, v) \in F$ 的 v , 此时 $v = i$; 判断 $(i, t_1) \notin WF$ 成立, 为 (i, t_1) 分配权重 1, 把 $1(i, t_1)$ 并入到 WF 中, 此时 $WF = \{1(i, t_1)\}$; 判断 $t_1 \cdot color = \varnothing$ 成立且 $t_1 \neq o$, 为 t_1 颜色赋予灰色并把 t_1 压入到 Q 中; 第二次循环开始时 Q

$= t_1$, 则循环结束时为边 (t_1, s_1) 赋予权重 1 并为 s_1 赋予灰色, 此时 $Q = s_1$, 如此循环得到图 2(a). 图 2(b) 的生成过程与图 2(a) 的不同点在于: 图 2(b) 在分配权重时会遇到一个环形结构, 无法依据权重分配规则为环形结构中的边分配权重, 即执行 WF 生成算法第 13 ~ 18 行, 遇到无法分配权重的边 (s_4, t_6) , 设它的权重为 x , 继续执行循环则得到图 2(b). 图 2(c) 和图 2(d) 的生成过程与图 2(a)、图 2(b) 同理, 图 2(e) 为通过列方程组计算出所有未知权重之后得到的权重分配图.

3 WF 相似性

3.1 相似性的计算方法

基于加权流关系的相似性算法的关键部分是加权流关系集的计算, 即上节给出的加权流关系集的生成算法, 本节基于加权流关系提出加权变迁的概念, 并给出 WF 相似性的定义.

定义 1 (加权变迁) 一个 workflow 网中任意一个变迁的权重等于该变迁的所有入边权重与所有出边权重之和, 计算变迁权重的方法为: 从加权流关系集中查找所有包含该变迁的流关系权重, 并求和.

所有变迁及其权重组成的集合称为加权变迁集 WT, 由变迁集 T 、流关系 F 及加权流关系集 WF, 可以很简单的求出 WT. 在计算相似性之前, 先给出加权变迁集的交集、并集、及模运算, 给定两个加权变迁集 WT_1 和 WT_2 :

$$WT_1 \cap WT_2 = \{Wx \mid W = \min(W_1, W_2) \wedge (W_1x \in WT_1 \wedge W_2x \in WT_2)\}$$

$$WT_1 \cup WT_2 = \{Wx \mid W = \max(W_1, W_2) \wedge (W_1x \in WT_1 \vee W_2x \in WT_2)\}$$

$$|WT| = \sum_{W \in WT} W_j$$

定义 2 (WF 相似性) 给定两个 workflow 网 WFN_1 和 WFN_2 , 它们的加权流关系集分别为 WF_1 和 WF_2 , 加权变迁集分别为 WT_1 和 WT_2 , 则相似性 $\text{sim}(WFN_1, WFN_2)$

$$= \frac{|WT_1 \cap WT_2|}{|WT_1 \cup WT_2|}$$

图 3 为给定两个 workflow 网 WFN_1 和 WFN_2 , 通过 WF 生成算法为流关系分配权重, 依据加权流关系集求得加权变迁集分别为: $WT_1 = \{2t_1, 1t_2, 1t_3, 2t_4\}$ 、 $WT_2 = \{2t_1, 2t_2, 2t_3, 2t_4\}$. 则 WT_1 和 WT_2 的相似度为 $\text{sim}(WFN_1, WFN_2) = \frac{|WT_1 \cap WT_2|}{|WT_1 \cup WT_2|} = \frac{2+1+1+2}{2+2+2+2} = 0.75$.

3.2 WF 算法

算法 2 计算 workflow 网间的相似性

输入: 两个 workflow 网 WFN_1 和 WFN_2

输出: 相似性结果

1. 用算法 1 分别处理 WFN_1 和 WFN_2 , 得 WF_1 和 WF_2
2. $WT_1 = \varnothing$

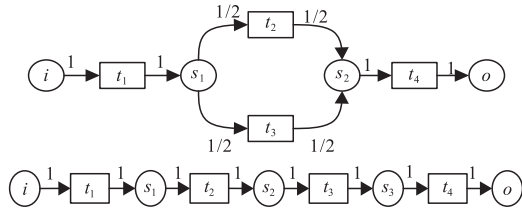


图3 带有权重的两个工作流网

3. for each $t \in T_1$
4. 计算 t 的所有边的权重之和, 记作 W
5. $WT_1 = WT_1 \cup \{Wt\}$
6. $WT_2 = \varphi$
7. for each $t \in T_2$
8. 计算 t 的所有边的权重之和, 记作 W
9. $WT_2 = WT_2 \cup \{Wt\}$
10. $WT_1 \cap WT_2 = \{Wx | W = \min(W_1, W_2) \wedge (W_1x \in WT_1 \wedge W_2x \in WT_2)\}$
11. $WT_1 \cup WT_2 = \{Wx | W = \max(W_1, W_2) \wedge (W_1x \in WT_1 \vee W_2x \in WT_2)\}$
12. $\text{sim}(WFN_1, FWN_2) = \frac{|WT_1 \cap WT_2|}{|WT_1 \cup WT_2|}$

WF 算法的输入是两个工作流网, 输出是两个工作流网的相似性, 算法第 1、2 行调用 WF 生成算法生成加权流关系集, 3 ~ 10 行计算加权变迁集, 11 ~ 13 行计算 WF 相似性。

表 1 抽取的 10 个模型之间的相似性

序号	1	2	3	4	5	6	7	8	9	10
1	1.000	0.387	0.401	0.711	0.287	0.396	0.144	0.601	0.813	0.597
2	0.387	1.000	0.330	0.299	0.156	0.487	0.106	0.400	0.356	0.271
3	0.401	0.330	1.000	0.411	0.356	0.000	0.318	0.656	0.460	0.667
4	0.711	0.299	0.411	1.000	0.223	0.337	0.000	0.667	0.711	0.501
5	0.287	0.156	0.356	0.223	1.000	0.273	0.166	0.195	0.262	0.567
6	0.396	0.487	0.000	0.337	0.273	1.000	0.143	0.314	0.433	0.511
7	0.144	0.106	0.318	0.000	0.166	0.143	1.000	0.000	0.135	0.145
8	0.601	0.400	0.656	0.667	0.195	0.314	0.000	1.000	0.602	0.375
9	0.813	0.356	0.460	0.711	0.262	0.433	0.135	0.602	1.000	0.567
10	0.597	0.271	0.667	0.501	0.567	0.511	0.145	0.375	0.567	1.000

4.2 WF 算法相似性性质验证

本小节通过实验比较 WF 算法相似性性质的满足情况, 本文总结已有算法提出的 7 种相似性性质为 4 种, 分别是: 结构漂移不变性 (性质 1)、跨度负相关性 (性质 2)、无关递减性 (性质 3)、循环序列长度负相关性 (性质 4)。

性质 1 结构漂移不变性, 包括顺序、排他、并行、循环结构漂移不变性。结构漂移不变性是指对流程模型中的顺序结构进行改造, 改造成的结构分支无论添加在顺序结构的哪部分上, 得到的新模型与原模型的相似性均相等。

如图 4, 在原模型的 3 个不同位置上, 添加上顺序

4 实验与分析

4.1 WF 算法验证

本文的实验数据分为两类: 第一类是从 SAP 流程库中抽取的 210 个典型的业务流程模型, 并用按照本文对工作流网的要求及本文方法适用的工作流网, 第二类是人工编纂的 50 个流程模型以达到实验需求, 人工编纂的模型主要用来验证 WF 算法的相似性性质满足情况, 也用于验证 WF 算法的性能。限于篇幅, 本节只给出 10 个模型之间的相似性实验结果, 如表 1 所示。

表 1 为从 SAP 流程库和人工编纂中抽取的 10 个模型之间的实验结果, 实验结果为两两模型间的相似性, 该实验结果验证了 WF 算法的正确性和可行性, WF 算法包含了 WF 生成算法, 因此 WF 生成算法的正确性和可行性同样得到验证。由实验结果及 WF 相似性定义可知: ①WF 相似性取值范围为 $[0, 1]$; ②两个模型完全相同时, WF 相似性取到 1 值; ③两个模型完全不同时, 即没有相同的变迁, WF 相似性取到 0 值。由表 1 不难发现, WF 相似性满足自反性、对称性、非负性等, 由 WF 相似性及相关定义即可证明上述三种性质, 本文不作赘述。

结构, 得到 3 个新的模型, 3 个模型满足: $\text{sim}(N_1, N) = \text{sim}(N_2, N) = \text{sim}(N_3, N) = 0.667$ 。如图 5, 在 N 上分别添加排他结构的分支, 构造出两个新的模型 N_4 和 N_5 , 满足: $\text{sim}(N_4, N) = \text{sim}(N_5, N) = 0.600$ 。扩展原模型至 10 个顺序变迁, 以验证并发和循环结构漂移不变性, 新的含有并发结构的模型与原模型的相似性为 0.909, 新的含有循环结构的模型与原模型的相似性为 0.833。为了防止偶然性, 用扩展至 10 个变迁的模型回溯验证顺序、排他结构漂移不变性: WF 算法确实满足顺序、排他结构漂移性, 相似性分别分: 0.909、0.905。具体模型及计算过程本文不再赘述。

性质 2 跨度负相关性, 是指对模型中顺序结构添

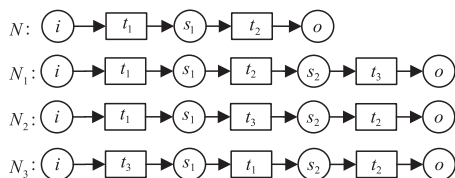


图4 顺序结构漂移不变性

加排他结构分支,该分支在原模型结构上跨度越大,得到的新模型与原模型的相似性越小.

性质3 无关递减性,是指对模型中顺序结构添加排他结构分支,添加的分支越多,得到的新模型与原模型的相似性越小.

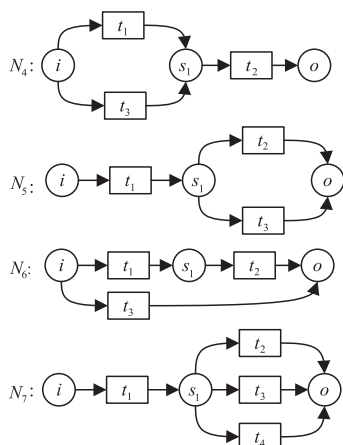


图5 排他结构漂移不变性和跨度负相关性和无关递减性

如图5所示,模型 N_6 和 N_7 是改造原模型 N 得到的模型,其中 $\text{sim}(N_6, N) = 0.400 < 0.600 = \text{sim}(N_4, N)$, $\text{sim}(N_7, N) = 0.500 < 0.600 = \text{sim}(N_5, N)$. 由两个式子判断,WF相似性分别满足跨度负相关性和无关递减性,扩展原模型至10个变迁进一步验证:WF相似性确实满足跨度负相关性和无关递减性.

性质4 循环序列长度负相关性,指对流程模型中的顺序结构进行添加循环结构改造,循环结构的跨度越大,得到的新模型与原模型的相似性越小.

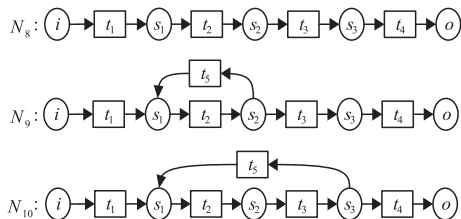


图6 循环序列长度负相关性

如图6,模型 N_9 和 N_{10} 是在原模型 N_8 上添加循环结构改造成的新模型,添加的循环序列长度(跨度)不同, $\text{sim}(N_9, N_8) = 0.667$, $\text{sim}(N_{10}, N_8) = 0.571$,因此WF相似性满足 $\text{sim}(N_{10}, N_8) < \text{sim}(N_9, N_8)$,即符合循环序列长度负相关性. 扩展原模型至10个变迁,取不同跨度的

循环结构共8个,所有新得到的8个模型均满足循环序列长度负相关性.

表2 各算法对相似性性质的满足情况

	性质1	性质2	性质3	性质4
PTS	√	√	√	×
TAR	×	×	√	×
TAR++	√	√	√	√
BP	√	×	√	√
SSDT	√	√	√	√
WF	√	√	√	√

经过实验及文献研究得出各算法对相似性性质的满足情况如表2所示,其中√表示满足,×表示不满足. 由表2可以看出:TAR++、SSDT、WF算法能够满足所有的4种相似性性质. 然而,TAR++和SSDT算法在时间复杂度上弱于WF算法,且TAR++算法需改动模型,引言中具体描述了算法之间的优劣,并将在后续章节中详细分析.

4.3 时间复杂度分析及性能比较

更全面的算法优劣比较应该包括算法运行时间的对比,分析WF算法的时间复杂度,给定一个工作流网WFN,变迁集为 T ,库所集为 P ,流关系集为 F ,则WF生成算法的时间复杂度为 $O(|T| + |P| + |F|)$. WF生成算法遍历WFN中除输出库所外的所有节点及流关系,操作所需时间为 $O(|T| + |P| + |F|)$. 条件判断、计算权重、并入操作、赋予颜色、出队、入队等操作所需时间均为 $O(1)$,总时间为 $O(1)$. 用广度优先搜索方法遍历模型,需要设的未知权重为模型中循环的个数,循环个数极少时,求解未知权重所需时间为 $O(1)$. 用直接寻址表存储WF和WT,则WF算法3~10行、11~13行时间复杂度为 $\max(O(T_1), O(T_2))$. 综上,WF算法的时间复杂度为 $O(|T| + |P| + |F|)$,其中 $|T| = \max(|T_1|, |T_2|)$, $|P| = \max(|P_1|, |P_2|)$, $|F| = \max(|F_1|, |F_2|)$.

WF算法在满足相似性性质方面优于PTS、TAR、BP算法,且相比于SSDT算法有一个良好的时间复杂度,算法性能强于SSDT算法. 与TAR++算法比较:本文采用广度优先搜索方法生成加权流关系集,除非遇到循环结构,否则不会遇见入边权重未知的库所,极大地降低了算法的最坏时间复杂度,且相比于TAR++算法,WF算法有一个更高的上界.

表3 各算法运行时间

	PTS	TAR	TAR++	BP	SSDT	WT
数据1	5.8	5.3	5.1	7.6	10.1	4.3
数据2	6.1	5.3	5.2	7.7	10.3	4.4
数据3	6.3	5.4	5.2	7.9	11	4.6

本节通过实验比较各算法的运行时间,为了取消

偶然性,从 210 个模型中随机选取三组模型作为实验数据,分别记作实验数据 1、2 和 3,分别含有 98、113 和 120 个模型. 实验结果如表 3 所示,表中数字的单位为秒,SSDT 算法的运行时间最长,WF 算法(本文算法)运行时间最短. 结合相似性性质实验得出结论:WF 算法优于其他算法.

5 结束语

本文提出了一种基于加权流关系的相似性度量方法,并设计了加权流关系集的生成算法及相似性计算算法,通过实验验证了算法的正确性和可行性,算法不但满足所有的相似性性质,还降低了时间复杂度,提供了一个良好的相似性上界. 在今后的研究工作中,将继续深入研究各种相似性度量方法,扩展相似性性质,优化 WF 算法或提出新的算法,使相似性度量结果更加贴近领域专家的评估结果.

参考文献

- [1] Becker M, Laue R. A comparative survey of business process similarity measures [J]. *Computers in Industry*, 2012, 63(2): 148 - 167.
- [2] Wang J, Jin T, Wong R K, et al. Querying business process model repositories [J]. *World Wide Web-internet & Web Information Systems*, 2014, 17(3): 427 - 454.
- [3] 王子璇, 闻立杰, 汪抒浩, 等. 基于变迁标签图编辑距离的过程模型相似性度量 [J]. *计算机集成制造系统*, 2016, 22(2): 343 - 352.
WANG Zi-xuan, WEN Li-jie, WANG Shu-hao, et al. Similarity measurement for process models based on transition-labeled graph edit distance [J]. *Computer Integrated Manufacturing Systems*, 2016, 22(2): 343 - 352. (in Chinese)
- [4] 董子禾, 闻立杰, 黄浩未, 等. 基于触发序列集合的过程模型行为相似性算法 [J]. *软件学报*, 2015, 26(3): 449 - 459.
DONG Zi-he, WEN Li-jie, HUANG Hao-wei, et al. Behavioral similarity algorithm for process models based on firing sequence collection [J]. *Journal of Software*, 2015, 26(3): 449 - 459. (in Chinese)
- [5] WANG Jian-min, HE Teng-fei, WEN Li-jie. A behavioral similarity measure between labeled petri nets based on principal transition sequences [A]. *On the Move to Meaningful Internet Systems: OTM 2010* [C]. Berlin, Germany: Springer-Verlag, 2010. 394 - 401.
- [6] Zha H, Wang J, Wen L, et al. A workflow net similarity measure based on transition adjacency relations [J]. *Computers in Industry*, 2010, 61(5): 463 - 471.

- [7] 殷明, 闻立杰, 王建民, 等. 基于变迁紧邻关系重要性的流程相似性算法 [J]. *计算机集成制造系统*, 2015, 21(2): 344 - 358.
YIN Ming, WEN Li-jie, WANG Jian-min, et al. Process similarity algorithm based on importance of transition adjacent relations [J]. *Computer Integrated Manufacturing Systems*, 2015, 26(3): 449 - 459. (in Chinese)
- [8] Weidlich M, Emendling J, Weske M. Efficient consistency measurement based on behavioral profiles of process models [J]. *IEEE Transactions on Software Engineering*, 2011, 37(3): 410 - 429.
- [9] 汪抒浩, 闻立杰, 魏代森, 等. 基于任务最短跟随距离矩阵的流程模型行为相似性算法 [J]. *计算机集成制造系统*, 2013, 19(8): 1822 - 1831.
WANG Shu-hao, WEN Li-jie, WEI Dai-seng, et al. SSDT matrix-based behavioral similarity algorithm for process models [J]. *Computer Integrated Manufacturing Systems*, 2013, 19(8): 1822 - 1831. (in Chinese)
- [10] Montani S, Leonardi G, Quaglini S, et al. A knowledge-intensive approach to process similarity calculation [J]. *Expert Systems with Applications*, 2015, 42(9): 4207 - 4215.

作者简介



段 瑞 男, 1993 年 3 月出生于安徽省涡阳县. 现为安徽理工大学数学与大数据学院硕士研究生. 主要研究方向为业务流程过程挖掘, Petri 网理论与应用.
E-mail: 85768312@qq.com



方 欢 (通信作者) 女, 1982 年 3 月出生于安徽省池州市. 现为安徽理工大学数学与大数据学院副教授, 博士. 主要研究方向为业务流程过程挖掘, Petri 网理论与应用.
E-mail: fanghuan0307@163.com



詹 悦 女, 1994 年 3 月出生于安徽省安庆市太湖县. 现为安徽理工大学数学与大数据学院硕士研究生. 主要研究方向为 Petri 网理论与应用.
E-mail: 2272350170@qq.com