DOI:10.13196/j.cims.2015.02.007

基于变迁紧邻关系重要性的流程相似性算法

殷　明[1],闻立杰[1+],王建民[1],肖　汉[1],丁子哲[2],高　翔[2]
(1.清华大学 软件学院,北京　100084;2.中国移动通信集团公司 管理信息系统部,北京　100033)

摘　要:为了提高模型的检索效率,提出一种基于变迁紧邻关系重要性的相似性算法 TAR++,其主要思想是在流程中两两任务之间的紧邻关系上增加一个重要性系数,并将流程的相似度用带重要性的变迁紧邻关系集合的相似度表示。经证明,该算法对应的 TAR++距离满足距离度量性质。在 SAP、东锅、北车三个公司实际业务流程数据上进行的相关实验表明,该算法具有比较好的时间效率以及良好的灵活性和可操作性。

关键词:展开网;行为相似性;变迁紧邻关系;相似性算法评估
中图分类号:TP309　　**文献标识码**:A

Process similarity algorithm based on importance of transition adjacent relations

YIN Ming[1], *WEN Li-jie*[1+], *WANG Jian-min*[1], *XIAO Han*[1], *DING Zi-zhe*[2], *GAO Xiang*[2]
(1. School of Software, Tsinghua University, Beijing 100084, China;
2. Department of Management Information System, China Mobile Communication Corporation, Beijing 100033, China)

Abstract: To improve the retrieval efficiency of models, a similarity algorithm named TAR++ was presented based on importance of Transition Adjacent Relations (TARs). The main idea of TAR++ was to describe the transitions relationship through adding an importance argument on TARs, and present the similarity of models with the similarity of TARs sets. The experiment proved that the distance of TAR++ algorithm was satisfied the properties of distance metrics. The relative tests on actual business process of three corporations of SAP, East pot and BeiChe showed that the proposed algorithm had better efficiency, flexibility and operability.

Key words: unfolding net; behavioral similarity; transition adjacency relation; similarity algorithm evaluation

0　引言

流程作为企业三要素(即组织、数据和流程)之一,其管理一直是企业管理的重要课题。尤其大型集团企业的业务流程数以万计,如何对这些流程进行有效地比较、索引和搜索,成为更具挑战的课题。流程的相似性问题更是重中之重,因为只有好的流程相似性算法才能给流程搜索的准确性和有效性提供保障[1]。Yan Zhiqiang 等阐述了一个快速的流程相似性算法对企业实际应用的价值[2]。在实际研究中,如中国移动这类大集团公司的业务流程非常繁杂众多,而且具有较大的冗余性,如果没有快速精准的相似性算法,将极大地影响企业的运行效率。为了提高效率,大型企业通过维护流程库进行流程管理,通过流程的搜索快速定位相似流程,从而为企业经常进行的建模、查询、重构等工作打好基础。

流程相似性问题是指给定一个流程模型,如何从已经构建的流程库中查询到最相似的流程,其中心思想是给定两个流程模型,快速计算其相似性值。BECKERA 等阐述了相似性问题的含义,并说明了

收稿日期:2014-12-01。Received 01 Dec. 2014.
基金项目:国家自然科学基金资助项目(61472207,61402301,61325008);教育部—中国移动科研基金资助项目(MCM20123011)。**Foundation items**: Project supported by the National Natural Science Foundation, China(No. 61472207,61402301,61325008), and the Ministry of Education & China Mobile Research Foundation, China(No. MCM20123011).

其研究价值[3]。虽然近些年流程相似性算法层出不穷,但是始终有一些未能解决的问题,主要表现在对一些常见模型的相似性度量值与预期不符、算法复杂度太高以至于无法运用到生产实际中,以及对非自主选择结构等特殊结构无法正确计算等。具体说来,相似性算法应该满足顺序结构漂移不变性(性质1)、互斥结构漂移不变性(性质 2)、跨度负相关性(性质 3)、非替代无关递减性(性质 4)、循环序列长度负相关性(性质 5)五个性质[4]。目前主流的相似性算法包括变迁紧邻关系(Transition Adjacency Relation,TAR)算法、CF(causal footprint)算法、PTS(principal transition sequence)算法、BP(behavior profile)算法、SSDT(short succession distance between tasks)算法等,下面对其进行简单介绍。

查海平等提出 TAR 算法[5],该算法考察流程变迁的两两紧邻关系:如果变迁 A 和变迁 B 在某一次执行中出现顺序相邻,则称作一个变迁紧邻关系。对于任意一个模型,获取其所有的两两紧邻关系,称之为 TAR 集合,最后将两个流程的相似性用其对应的两个 TAR 集合之间的 Jaccard 系数表征。然而,该算法只满足性质 4;另外,TAR 算法也不能正确处理不可见任务、非自主选择结构等特殊流程结构。

VAN DONGEN 等提出 CF 算法[6],通过定义前向链和后向链两种链接,描述了模型中变迁之间的关系,并通过这两种链接在词法、句法和语义三个层次的比较进行相似性度量,该算法也只满足性质 1 和性质 4;另外,该算法的性能经证实不理想。

WEIDLICH 等提出的 BP 算法[7]定义了一种弱关系,将 TAR 中的紧邻关系做了扩展,允许变迁不紧邻的情况,具体包括严格顺序关系、互斥关系和交错顺序关系等,该算法不能满足性质 3;另外,BP 算法要求给定 Petri 网必须是自由选择的,限制了其应用范围。

王建民教授等提出一种使用主变迁序列来计算相似性的方法,即 PTS 算法[8]。该算法通过计算流程模型的三种主变迁序列来表征模型行为,通过两两序列的最长公共子序列长度与两序列中最长序列长度的比值作为这两个序列的相似性度量,最后再将序列集合中的所有序列相似性加权求和作为模型的相似性,该算法不能满足性质 5,即对循环的处理不得当;另外,当并发分支很多时,该算法的效率也比较低。

汪抒浩等近期提出的 SSDT 算法[4]能够满足上述提出的五个性质,该算法通过构造任务最短跟随距离矩阵来度量模型的相似性。由于 SSDT 算法基于给定的两个模型计算其任务最短跟随距离矩阵,并且需要根据矩阵的秩进行相应的扩展以保持两个矩阵秩相等,从而导致每次计算两个模型相似性时都需要重新对 SSDT 矩阵进行同维化操作,大大影响了算法的性能,使其不能充当一类流程模型索引。

由于现存的流程相似性算法存在诸多缺陷,本文尝试从几个角度进行优化。首先是各个算法不能满足相似性算法五个性质的问题,通过增加重要性系数,避免将所有事件关系等同化,从而解决原有 TAR 算法在处理互斥、循环等典型结构时遇到的问题;为了提高相似性算法的效率,通过优化算法尽量降低算法的迭代次数,将复杂度控制在多项式时间内;针对灵活性不高的问题,为 TAR++算法做了线下的 TAR++集合构建,从而使线上的时间复杂度降到最低。

本文主要介绍一种基于 TAR 的流程模型行为相似性改进算法 TAR++,创造性地为 TAR 增加了重要性系数,解决了原始 TAR 算法不能解决的众多问题。本文的主要贡献如下:

(1)通过在 TAR 上增加重要性系数,构造出一种带重要性的 TAR 集合,并进一步提出一种新的行为相似性算法——TAR++。

(2)证明了 TAR++算法满足度量空间的自反性、非负性、同一性和三角不等式四个特性,并证明了通过 TAR++算法计算相似性一定可以得到收敛的解。

(3)通过流程模型应该满足的五个基本性质给出了一个简单的相似性算法评估指标,并且比较了 TAR++算法和其他主流算法对该评估指标的满足情况。

(4)通过企业的实际业务模型数据证明了上述结论,并说明 TAR++算法具有良好的性能,可以投入到实际应用中。

1　预备知识

首先介绍 TAR++算法需要的一些预备知识,主要包括 Petri 网基本概念、Petri 网展开技术、行为语义表示技术等,这些研究工作构成 TAR++算法的基础。

1.1 Petri 网

定义 1 Petri 网。Petri 网是一个三元组(P,T,F)。其中:P 是库所的有限集合,T 是变迁的有限集合,满足 $P\cap T=\varnothing$;$F\subseteq(P\times T)\cup(T\times P)$是边集合,即 F 是一个映射到$\{0,1\}$的函数,若 $F(p,t)=1$,则 p 到 t 有一条有向弧。一个标签 Petri 网系统是一个六元组$(P,T,F,N,L,M0)$,其中:N 为事件名称集合,L 为变迁集合到名称集合的一个映射,$M0$ 为其初始标识。Petri 网集合 $P\cup T$ 中的任意元素称为 Petri 网的节点;Petri 网 F 集合中的一个元素即为 Petri 网的边或称一条有向弧,对于某个节点,其相邻的边称作该节点的边,节点的边包括输入边和输出边,分别简称为入边和出边。

更详细的 Petri 网相关概念请参考文献[9]。

定义 2 工作流网。一个 Petri 网 $PN=(P,T,F)$被称作工作流网,当且仅当:①PN 只有一个起始库所 i,即 $\cdot i=\varnothing$;②PN 只有一个终止库所 o,即 $o\cdot=\varnothing$;③如果增加一个变迁 t^* 连接库所 o 和 i,即 $t^*\cdot=\{i\}$且 $\cdot t^*=\{o\}$,则得到的 Petri 网是强联通的。

本文讨论的所有 Petri 网模型都是基于安全的工作流网。

1.2 变迁紧邻关系

定义 3 TAR。如果 FS 是工作流网(P,T,F)的所有可能发生序列,并且有 $a,b\in T$,则$\langle a,b\rangle$称为一个 TAR,当且仅当存在一个轨迹 $\sigma=t_1t_2t_3\cdots t_n$ 使得$\sigma\in FS,t_i=a$ 且 $t_{i+1}=b(i\in\{1,2,\cdots,n-1\})$。对于任意工作流网,所有 TAR 构成的集合称为该工作流网的 TAR 集合。

定义 4 隐式依赖。对于一个工作流网(P,T,F),网中的两个变迁 T_1,T_2 存在隐式依赖关系,即 $T_1\gg T_2$,当且仅当 T_1 和 T_2 满足以下三个性质:①连通性,即 T_1 的输出库所和 T_2 的输入库所交集不为空,且交集中的任何一个库所都不是冗余的;②分割性,即不存在任何一个可达标识 $s\in[PN,[i]\rangle$使得$(PN,s)[T_1\rangle$且$(PN,s-\cdot T_1+T_1\cdot)[T_2\rangle$;③可达性,即存在可达标识 $s([PN,[i]\rangle$使得$(PN,s)[T_1\rangle$,同时存在可达标识 $s'\in(PN,s-\cdot T_1+T_1\cdot\rangle$使得$(PN,s')[T_2\rangle$。

1.3 展开网和完全有向前缀

定义 5 展开网。对于给定的 Petri 网(P,T,F),可以构造其展开网。展开网是一个六元组$(P',T',F',L',M0',Z)$,其中 Z 是展开网中 P'和 T'分别到原网 P 和 T 的映射函数。该网的特征包括:①网中每个库所的输入变迁个数不大于 1;②网中不存在环;③网中不存在自冲突的元素;④对于网中的任意一个节点 $x\in P'\cup T'$,$y(y\in P'\cup T'$,满足 $y\rightarrow x)$的数量是有限的。

定义 6 配置与本地配置。在一个出现网$(P',T',F',L',M0',Z)$中,T'的子集被称作一个配置,当且仅当:①该子集是因果关系上的闭包;②该子集是没有冲突的。

在一个出现网中,如果存在这样一个配置,它是包含变迁 t 及其之前所有变迁的最小配置,则称其为变迁 t 的本地配置,记为$=\!>t$。

定义 7 完全有向前缀。称一个网系统的分支进程 B 是完全的,当且仅当对于所有可达状态 M,都存在一个配置 C,满足:①$Mark(C)=M$(即 M 在 B 中);②对于所有可以被 M 使能的变迁 t,都存在一个配置 $C\cup\{e\}$,满足 e 不属于 C 且 e 被 t 标记。

对于具有重名任务或者包含隐式依赖的模型,需要首先进行预处理。

下面举例对以上概念进行说明。图 1 所示为要展开的原始 Petri 网模型,通过检查从初始状态每一步可能发生的变迁以及各状态下的配置信息说明如何得到展开图(如表 1)。

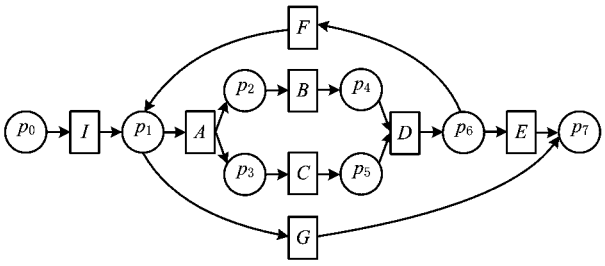


图1 将要展开的原始Petri网模型

表 1 图 1 所示 Petri 网的展开过程

步骤	状态及可能发生的变迁	配置信息
1	$(I,\{p_0\})$	
2	$(A,\{p_1\})$ $(G,\{p_1\})$	$[A]=\{I,A\}$ $[G]=\{I,G\}$
3	$(B,\{p_2\})$ $(C,\{p_3\})$	$[B]=\{I,A,B\}$ $[C]=\{I,A,C\}$
4	$(D,\{p_4,p_5\})$	$[D]=\{I,A,B,C,D\}$
5	$(E,\{p_6\})$ $(F,\{p_6\})$	$[E]=\{I,A,B,C,D,E\}$ $[F]=\{I,A,B,C,D,F\}$

续表 1

6	$(A,\{p_1\})$ $(G,\{p_1\})$	$Mark(F)=\{p_1\}$ $Mark(I)=\{p_1\}[I]=\{I\}$ $[F]=\{I,A,B,C,D,F\}$
…	…	…

有关消除重名任务、确定隐式依赖及 TAR 集合的构造等内容请参考文献[10]。

2　TAR++算法

本文通过在 TAR 上增加重要性系数来研制新的算法——TAR++。首先引入 TAR 关系的重要性概念,然后介绍流程模型中节点边系数的确定方法,说明如何通过这些边系数计算两个模型之间的 TAR++相似性,并证明该相似性对应的距离满足度量空间四要素。

2.1　TAR 关系的重要性

TAR 算法只考虑了两个工作流网的 TAR 集合元素交集与并集数量之间的比例关系,对于 TAR 集合内部的每个元素来说其重要性相等。造成的结果是,对于互斥分支和循环结构等情况,TAR 集合不能很好地反映其内部元素的重要程度,导致无法区分这些结构上的差异。例如,在两个分支的互斥结构中,TAR 算法认为所有 TAR 关系的重要性是相同的,而实际上若主干上 TAR 的重要性为 1,考虑到互斥分支上变迁发生的概率都是 1/2,有理由说互斥分支上 TAR 关系的重要性分别是 1/2。

因此,对 TAR 集合内的每个元素引入重要性的概念。首先定义边系数的相关概念和规则。

定义 8　节点的边系数。对于任意工作流网 PN,每个节点相邻的边上可以标记一个系数,该系数称为节点的边系数。系数由后面的算法 1 确定。

例 1　节点边系数示意。如图 2 所示,各条边上都标记了一个系数,该系数被称为节点的边系数。具体说来,变迁 P_1 的边系数包括:边 $T0-P_1$ 的系数为 1,边 P_1-T_1 的系数为 1/2,边 P_1-T_2 的系数为 1/2。

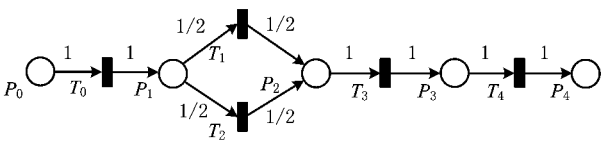


图2　节点的边系数示意

定义 9　节点边系数的确定性。对于任一合理的工作流网 PN,如果所有节点的边系数都已标注完成,则称为该工作流网的边系数被唯一确定。给定一个未被确定的工作流网,对于某个节点,如果该节点的所有边系数可以根据目前的系数标注唯一确定,则称该节点的边系数可以被确定,否则该节点的边系数无法确定。

节点边系数的确定需要根据三个规则进行。如果通过这三个规则可以唯一确定节点的边系数,则说明该节点边系数可以被确定,否则不能。这三个规则分别是变迁节点边系数相等、库所节点出边系数相等和库所出入边系数总和守恒。

首先,在一个工作流网中,并发结构均是由变迁引出多条边连接各个库所的结构,所有变迁中的事件都将必然发生,只是发生顺序不同。因此,并发结构分支上的变迁发生可能性和主干上的变迁相同,由此得到规则 1。

规则 1　变迁节点的边系数相等。给定一个合理的工作流网,对于其中任意的变迁节点,该节点所有的边系数(包括出边和入边)都相等。

其次,在互斥结构中,因为互斥的多个分支只有一个可以发生,所以对于各分支来说,假设发生的可能性相等,但是可能性只有主干分支的若干分之一,由此得到规则 2。

规则 2　库所节点出边系数相等。给定一个合理工作流网,对于其中任意的库所节点,设该节点共有 n 条出边,边系数分别为 $\theta_1,\theta_2,\cdots,\theta_n$,则这些出边的系数都相等,即 $\theta_1=\theta_2=\theta_3=\cdots=\theta_n$。

除此之外,如果将工作流网看做一个流网络(flow network),则可以引入图论中的相关结论。在一个流模型 $G=(V,E)$ 中,V 是该流模型的顶点,s 是其源点,t 是其汇点,E 是其边,其中每条边 $(u,v)\in E$ 均有一非负容量 $c(u,v)\geqslant 0$,可以在该流模型上定义流函数 f,该函数满足流守恒性,即对所有 $u\in V-\{s,t\}$,要求 $\sum_{v\in V} f(u,v)=0$。

由于工作流网中同样有边系数的概念,将以上结论映射到工作流网中,则对于任意库所,非起始库所和终止库的出边系数之和与入边系数之和应该相等,由此得到规则 3。

规则 3　库所出入边系数总和守恒。给定一个合理工作流网,对于其中任意非起始和终止库所的库所节点,其 m 条入边和 n 条出边的系数分别为 $\delta_1,\delta_2,\cdots,\delta_m,\theta_1,\theta_2,\cdots,\theta_n$,满足 $\sum_{i=1}^{n}\theta_i=\sum_{j=1}^{m}\delta_j$。

2.2 节点边系数的确定

根据确定边系数的三个规则，可以判断节点的边系数是否被唯一确定。如果可以被确定，则通过节点边系数的确定算法来确定该系数。对于整个工作流网，按照一定顺序遍历网络中的所有节点，如果可以确定其边系数，则标记其系数，否则计算下个节点，直至所有节点均被标注。

确定工作流网边系数的算法具体如下：

算法 1 工作流网 PN 边系数的确定。

Workflow_Coefficients_Calculation(WF-net PN)

输入：工作流网 PN。

输出：带系数的工作流网 PN。

BEGIN

1. 桥接起始库所和终止库所：若原工作流网有多个起始库所，则用变迁作为桥接节点使其依次相连；终止库所进行类似处理

2. 增加四个节点四条边，即增加两个库所 Ps 和 Pe，并增加两个变迁 Ts 和 Te 以及四条有向弧，分别连接：

-人工增加的起始库所 Ps 连接至人工增加的起始变迁 Ts

-人工增加的起始变迁 Ts 连接至原工作流的起始库所 *source*

-原工作流的终止库所 *sink* 连接至人工增加的终止变迁 Te

-人工增加的终止变迁 Te 连接至人工增加的终止库所 Pe

3. 消除重名任务

4. 初始化步骤：初始化队列 Q 的元素为新增的初始库所 Ps，初始化工作流网中所有的边系数为 0，初始化与人工增加的两个库所 Ps，Pe 相邻的边系数为 1，初始化所有节点为未访问过

5. 对队列中的每个节点 x 进行循环访问，且当队列为空时跳出循环：如果 x 还未被访问过，则根据上文的三个规则计算该节点的入边系数，并深度优先搜索该节点 DFS_Search(PN,x)

6. 循环结束后，所有节点被搜索完毕，得到原网络的 n 元一次方程组，解该方程组，将结果标记到网络的所有边上

END

其中，深度优先搜索节点 x 的步骤如下：

DFS_Search(WF-net PN,node x)

输入：工作流网 PN，当前要搜索的节点 x。

BEGIN

1. 如果节点 x 可以计算其边系数，则计算其边系数并且标记，否则新增一个未知数 xi，并且设该节点的一个出边的系数为 xi

2. 将该节点标记为已访问，并从队列中删除该节点

3. 令 outputNode p 为该节点紧邻的一个出节点

4. 对于每个出节点 p，循环进行以下操作：

5. 如果该出节点还未被访问过，则将该节点增加到队列中，并递归地深度优先搜索该节点 DFS_Search(PN,p)，

END

下面用一个实际流程模型的例子说明以上计算过程。

图 3 所示为 HighSpeed 公司的内部订单结算流程及其应用上述算法的计算过程。从上而下进行的操作解释如下：

步骤 1 执行算法的前三步，完成模型的初始化，当前搜索到节点 addedSource P_1。

步骤 2 针对队列中每个未被访问过的节点进行深度优先搜索，当前搜索到节点 P_1。

步骤 3 根据规则 2 确定 P_1 的出边系数为 1/2，并且沿着某条支线一直搜索下去，当前搜索到 P_3。

步骤 4 由于节点 P_3 属于无法确定边系数的节点，需要引入位置变量 x 作为该节点一条出边的系数，并以此为基础继续深度搜索，直到节点 P_5。

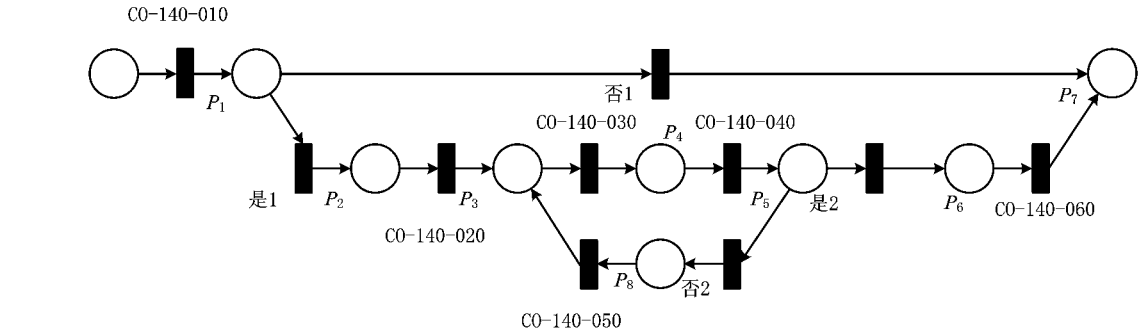
步骤 5 根据规则 2 确定 P_5 的一条出边的系数，并且继续搜索到节点 P_3。此时发现 P_3 已经被搜索过，因此就节点 P_3 利用规则 3 的节点出入边系数总和相等列方程。

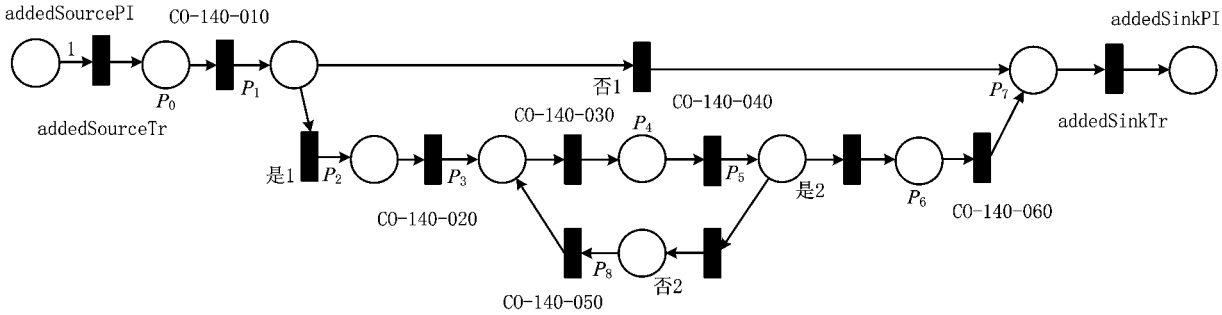
步骤 6 根据列出的方程解出此时的未知数 x，将 x 的值代入原图。

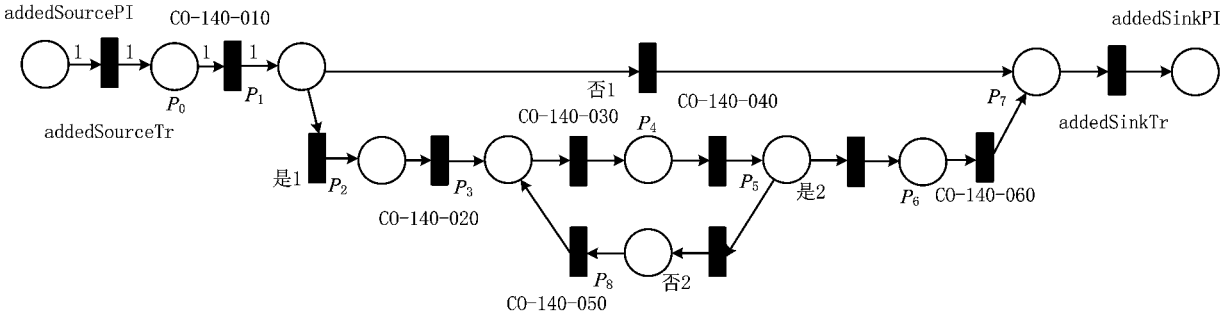
步骤 7 搜索回溯到节点 P_5，继续深度搜索下一个节点并标注系数。继续以上类似过程，最后确定所有系数。

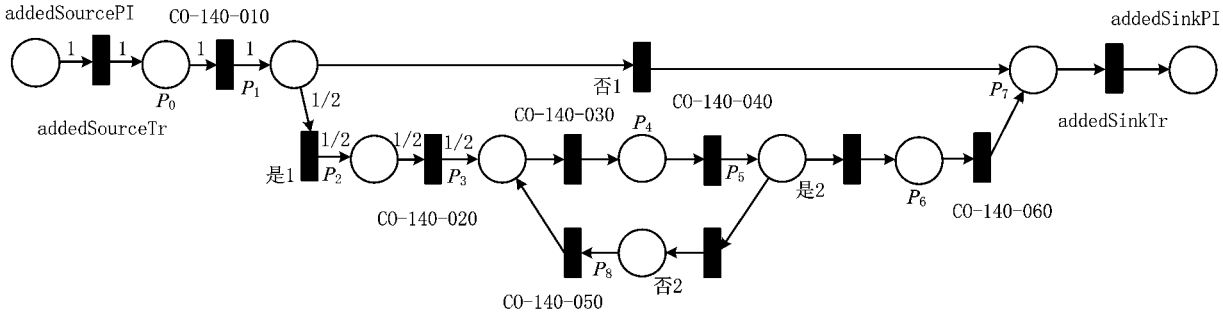
定理 1 通过算法 1 可以唯一确定整个工作流网的所有边系数。或者说，通过该算法求出的工作流网的边系数是唯一的，且该方法必将收敛。

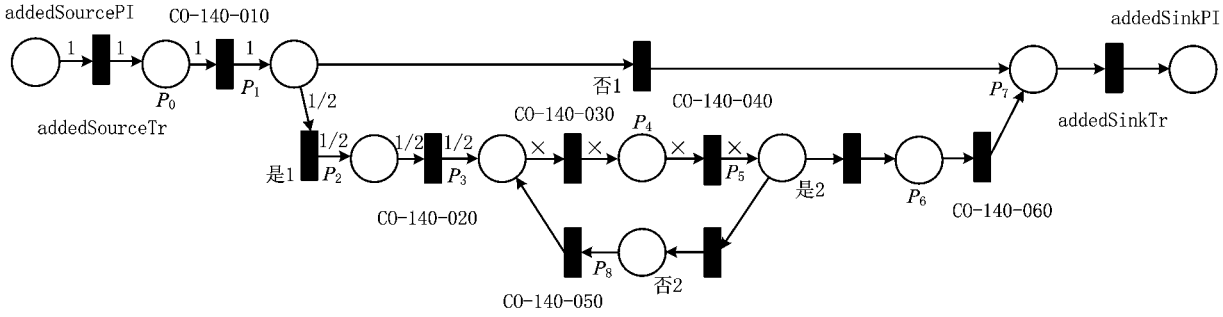
证明 由于确定工作流网边系数的过程是从前到后遍历整个网的过程，遍历过程中需要设一些未知变量，通过列方程求解这些未知变量即可获得解。根据标记边系数的三个规则，无论变迁还是库所，都是对边系数进行线性变换（即乘以一个常数），因此最终得到的方程必然是一次方程组。其次，新增未知变量的情况只有一种情况，即如图 4 所示，当遍历过程进行到某个库所时，该库所有 m 个未标记的输入边和 n 个未标记的输出边以及若干已知系数的边（由之前的遍历过程得到），需要增加 m 个未知变量，以将该节点相邻的边系数都表示出来（先增加一个变量 x 标记所有未标记的出边，再增加 $m-1$ 个变量分别标记 $m-1$ 个入边，最后一个入边根据该节点的出入边系数守恒得到）。因为这里的入边没有标记过，所以必然由 n 个未标记的输出边继续向后遍历得到。当遍历由这些出边再次进行到当前节点时，可以根据系数相等的规则列方程，即每个未标记的入边都可以得到一个方程，对于该节点，设 m 个未知量得到 m 个方程。同理，对于其他节点，每当设一个未知变量，就将得到一个方程，并且这些方程是互相独立列出的。由此得到一个 n 元一次方程











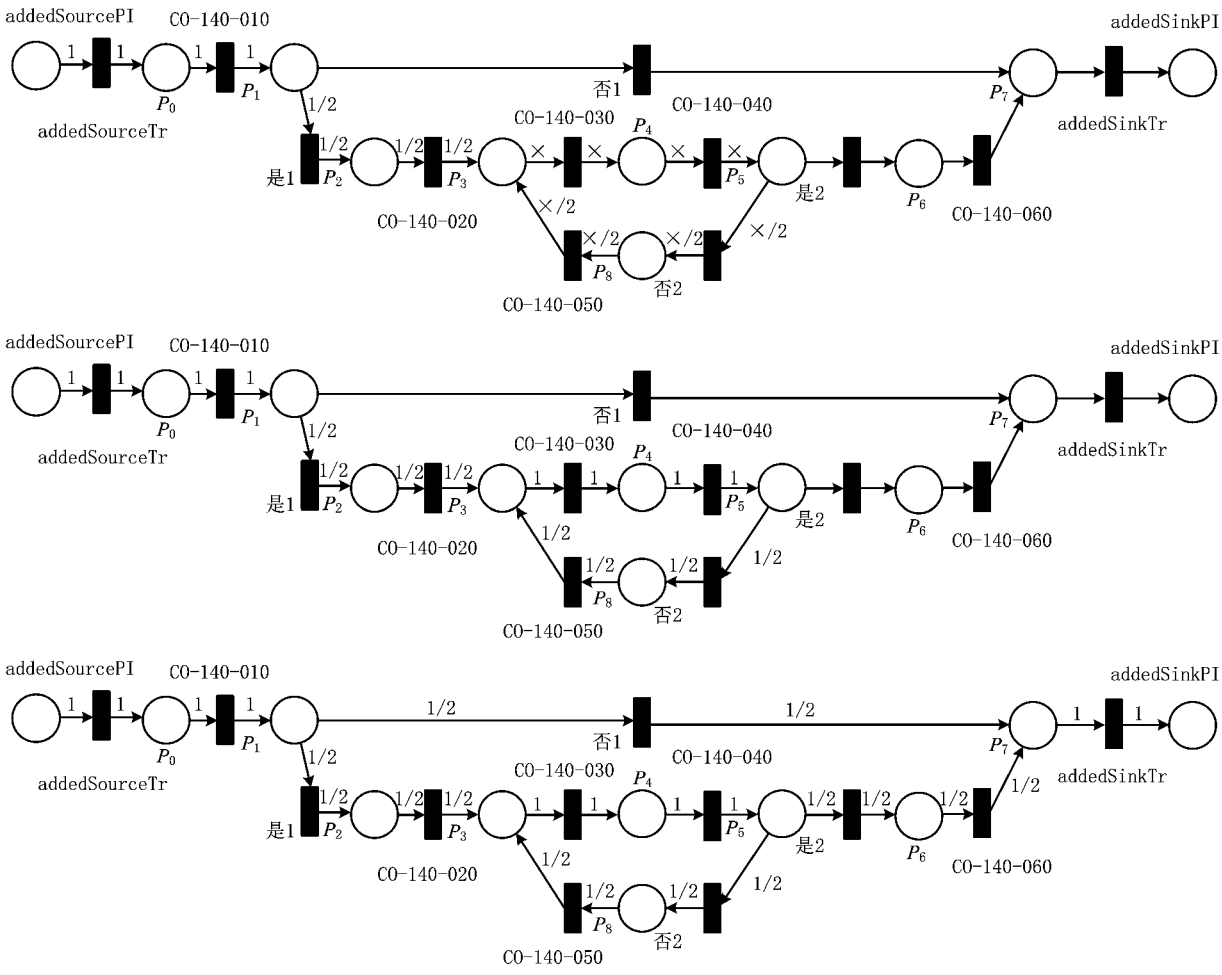


图3 在HighSpeed公司内部订单结算流程使用TAR++算法的实例

组，方程组的系数构成一个 n 阶矩阵，且矩阵的各行是独立的。因此，得到一组唯一的解向量，从而确定工作流网的边系数。算法 1 的实质是对整个模型进行了深度优先搜索，因此搜索完成之时程序结束，程序必将收敛。

2.3 TAR＋＋算法

对于任意的工作流网 PN，如果其所有边系数都已经被唯一确定，则可以构造出该网的带重要性的 TAR 集合。

定义 10 TAR 的重要性。对于给定的工作流网 PN，在其变迁紧邻集合 TAR 集合的基础上，每个 TAR 关系可以标注一个重要性系数，称为 TAR 的重要性。如果一个 TAR 集合中的所有 TAR 关系都被标注了重要性，则称该集合为带重要性的 TAR 集合，或 TAR＋＋集合。

算法 2 TAR 的重要性的确定。给定带边系数的工作流网 PN，对于任意一个 TAR 关系$\langle a,b\rangle$，必然存在一个中间库所 s 和 TAR 关系的两个变迁 a 和 b 同时相连接，其中 $a\rightarrow s$ 和 $s\rightarrow b$ 构成两条有向弧，其系数分别为 α 和 β，则 TAR$\langle a,b\rangle$的重要性取两者的最小值，即

$$\mathrm{TAR}\langle a,b\rangle = \min\{\alpha,\beta\}。$$

可以通过完全有向前缀(Complete Finite Prefix，CFP)构造一个工作流网的 TAR 集合，因此仅需遍历该 TAR 集合，根据已经标记好的边系数，即可唯一确定该工作流网中每个 TAR 的重要性，从而构造出带有重要性的 TAR 集合。在用 CFP 构造 TAR 集合时要注意变迁的隐藏依赖，即只包含显式依赖。

在 TAR＋＋算法中，TAR＋＋集合可以被看作是一个多集，多集的重复度即为元素的重要性系数。由此交集和并集的计算不能简单地用元素数目来表示，而是根据多集的交并运算，即设 A 和 B 是两个多重集合，A 与 B 的并集也是多集，并且每个元素的重复度等于该元素在 A 和 B 中重复度的最大值；A 与 B 的交集也是一个多集，并且每个元素的重复度等于该元素在 A 和 B 的重复度的最小值。

因此引入带重要性的 TAR 集合的交与并运算。

定义 11　带重要性的 TAR 集合的交与并运算。设给定两个带重要性的 TAR 集合 $TARS_1$ 和 $TARS_2$,其元素个数分别为 m 和 n,其中每个 TAR 关系的重要性系数分别为 $\alpha_1,\alpha_2,\cdots,\alpha_m$ 和 $\beta_1,\beta_2,\cdots,\beta_n$。则:

$TARS_1 \cap TARS_2 = \{\delta_i TAR_i \mid \delta_i = \min(\alpha_i,\beta_i)\}$;

$TARS_1 \cup TARS_2 = \{\delta_i TAR_i \mid \delta_i = \max(\alpha_i,\beta_i)\}$。

给定两个带重要性的 TAR 集合,就可以唯一确定地计算其交集与并集。因此得这两个工作流网的相似性定义:

定义 12　TAR++相似性。给定两个工作流网 N_1 和 N_2,初始状态为 M_1 和 M_2,其带重要性的 TAR 集合分别为多重集合 $TARS_1$ 和 $TARS_2$,则这两个工作流网的 TAR++相似性为

$$similarity((N_1,M_1),(N_2,M_2)) = \frac{|TARS_1 \cap TARS_2|}{|TARS_1 \cup TARS_2|}。$$

由于 $TARS_1$ 和 $TARS_2$ 是多集,交并运算应该使用多集的交并运算,求集合元素数目的运算也应该使用多集中元素重复度的加和来代替。

例 2　TAR++相似性计算示意。计算图 5 中模型 N_1 和模型 N_2 的相似性。这两个模型的 TAR++集合分别为

$$TARS_{N_1} = \{1T_s\text{-}T0, 1T_0\text{-}T_1, 1T_1\text{-}T_2, 1T_2\text{-}T_e\},$$

$$TARS_{N_2} = \{1T_s\text{-}T0, 1/2T_0\text{-}T_1, 1/2T_0\text{-}T_3, 1/2T_1\text{-}T_2, 1/2T3\text{-}T_2, 1T_2\text{-}T_e\}。$$

根据 TAR++相似性公式,模型 N_1 和 N_2 的相似性为

$$similarity(N_1,N_2) = \frac{1+\frac{1}{2}+\frac{1}{2}+1}{1+1+1+1+\frac{1}{2}+\frac{1}{2}} = 0.6。$$

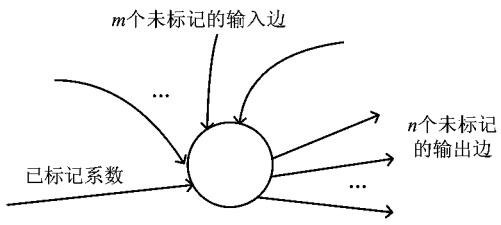


图4　当前节点的边标记情况示意

由此也可以定义两个工作流网的 TAR++距离:

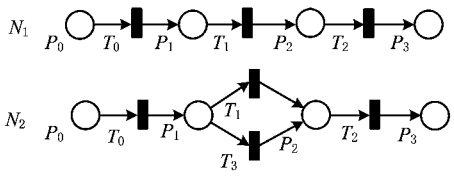


图5　TAR++相似性计算示意模型

定义 13　TAR++距离。给定两个工作流网 N_1 和 N_2,初始状态为 M_1 和 M_2,其带重要性的 TAR 集合分别为多集 $TARS_1$ 和 $TARS_2$,则这两个工作流网的 TAR++距离为

$$Distance((N_1,M_1),(N_2,M_2)) = 1 - \frac{|TARS_1 \cap TARS_2|}{|TARS_1 \cup TARS_2|}。$$

定理 2　TAR++距离满足度量空间的自反性、非负性、同一性、三角不等式四大特性。

证明　对于任意给定的模型 M_1 和 M_2,设其距离的度量空间矩阵为 $\boldsymbol{D}$(亦简称距离度量)。根据 TAR++算法,该度量空间由 M_1 和 M_2 的 TAR++集合的运算公式

$$Distance(M_1,M_2) = 1 - \frac{|TARS_1 \cap TARS_2|}{|TARS_1 \cup TARS_2|}$$

给定。这里 $|TARS_1 \cap TARS_2|$ 和 $|TARS_1 \cup TARS_2|$ 均为多集的交与并运算。

考虑这样的映射 $H(MS_1,MS_2)$,该映射将原来的多集 MS_1 和 MS_2 分别映射为两个单集 $TARS_1$ 和 $TARS_2$,即

$$MS_1,MS_2 \xrightarrow{H} TARS_1,TARS_2。$$

设多重集合 MS_1 中 m 个元素的系数分别为 $\lambda_1,\lambda_2,\cdots,\lambda_m$,$MS_2$ 中 n 个元素的系数分别为 $\eta_1,\eta_2,\cdots,\eta_n$,即

$$MS_1 = \{\lambda_i TAR_{1i}, i = 1,2,\cdots,m\};$$

$$MS_2 = \{\eta_j TAR_{2j}, j = 1,2,\cdots,n\}。$$

该映射的具体操作如下:

(1)计算一个最小的因子 E,该因子乘以每个 λ_i 和 η_j 后的所有系数均为正整数。因为根据 TAR++算法计算得到的系数 λ_i 和 η_j 必为有理数,且 m 和 n 为有界量(即 TAR++集合的元素为有限个),所以 E 必存在且唯一。将求得的 E 作为因子乘到每个元素上,得到:

$$MS'_1 = \{E\lambda_i TAR_{1i}, i = 1,2,\cdots,m\} = \{A_i TAR_{1i}, i = 1,2,\cdots,m\};$$

$$MS'_2 = \{E\eta_j TAR_{2j}, j = 1,2,\cdots,n\} = \{B_j TAR_{2j}, j = 1,2,\cdots,n\}。$$

(1)将每一个 TAR_{1i} 元素拆成 A_i 个新元素，即 TAR_{11}^{A1}，TAR_{12}^{A2}，TAR_{13}^{A3}，…，TAR_{1m}^{Am}，同理对每一个 TAR_{2j} 进行类似操作，拆为 B_j 个新元素，即 TAR_{21}^{B1}，TAR_{22}^{B2}，TAR_{23}^{B3}，…，TAR_{2n}^{Bn}，从而构成两个单集合：

$$TARS_1 = \{TAR_{1i}^{Ai}, i = 1,2,\cdots,m\},$$

$$TARS_2 = \{TAR_{2j}^{Bj}, j = 1,2,\cdots,n\}。$$

对于这两个单集合，可以运用一般集合上的结论进行推导。

根据斯坦豪斯变换(Steinhaus transform)，给定一个度量(X,d)和一个固定点 $a\in X$，可以定义一个新的距离度量 $\boldsymbol{D}'$，

$$D'(x,y) = \frac{2D(x,y)}{D(x,a)+D(y-a)+D(x,y)}。$$

如果 $\boldsymbol{D}$ 是一个距离度量，则新构造的 $\boldsymbol{D}'$ 也是一个距离度量。现在设 $\boldsymbol{D}$ 为集合的对称差，则新构造的 $\boldsymbol{D}'$ 就是 Jaccard 距离。这说明映射得到的单集 TAR_1 和 TAR_2 是一种距离度量。由于映射过程只是为相同的元素添加了新的标记，并没有改变集合本身的结构，可知原来的多集下的 TAR++度量也是一种距离度量。证毕。

2.4 算法复杂度分析

本节提出的确定边系数的算法本质上是进行了一次深度优先搜索，其结果是一个深度优先森林。由于在遍历过程中仅做了变量标记，这一步算法的遍历操作的时间复杂度为 $O(V+E)$。该时间复杂度是多项式的，即对于任意的工作流网，理论上都可以在多项式时间内确定其边系数。之后，通过解方程确定系数的具体值。假设算法运行到当前节点时出现了还没有标记的入边，因为算法从前向后依次进行，这些入边显然是从该节点的出边出发，经过一系列循环最终回到该节点，所以称为该节点的回边。如果该节点为变迁，则根据变迁出入边系数相等原则，只要继续向前标记算法即可以继续前进。然而，当该节点为库所时需要引入新的未知变量 x。因此，如果一个模型中存在 N 个有回边的库所，则应引入 N 个未知变量，得到一个 N 元一次方程组，通过解该方程组完成对系数的标记。根据克莱姆法则(Cramer's rule)计算 N 元一次方程组的复杂度为 $O(N!)$，则最终算法复杂度应该为 $O(V+E+N!)$。因此当具有回边的库所很多时，该算法复杂度就不可容忍了。

如果要提高算法效率，则必须减少引入的未知数的数目。针对这个问题有两种加速方式：①在每次根据关键节点列方程时，都尽最大可能解出当前的未知数值，并将原图的未知数替代为解出来的新值。这样虽然遍历模型的次数增加了，但是在最后解方程时系数矩阵的阶数降低了。②由于除初始节点相邻的边系数已知且为 1 外，终止节点的相邻边系数也已知且为 1。这样，考虑将初始节点和终止节点同时加入初始队列中，依次迭代地搜索，相当于从模型的两边向中间逼近。遍历到某节点时，如果可以确定其边系数，则继续向下搜索；如果不能确定其边系数，则放弃该节点，尝试队列中的下一个节点，直至整个队列中的所有节点都不能确定其边系数时再引入未知变量。这两种方式大大减少了未知数的数量，降低了算法的复杂度。

3 实验与分析

3.1 数据来源

本文的实验模型来自 SAP 公司(简称 SAP)、东方锅炉股份有限公司(简称东锅)和中国北方机车车辆工业集团公司(简称北车)三家企业的实际业务模型。这三家公司分别涉及企业管理、重工业生产等领域，其业务流程既具有大中型企业模型复杂的共同特征，又带有其自身明显的业务特色，极具研究价值。从 SAP 流程库中抽取 80 个典型业务流程，以及为了实验需要人工编纂的一些流程作为数据集一(其中人工编纂是通过人工对 SAP 中的一些模型建模得到)，从东锅的流程库中抽取 36 个典型业务流程作为数据集二，从北车流程库中抽取 63 个典型业务流程作为数据集三，共计 179 个流程作为实验的基础数据集，如表 2 所示。

3.2 距离性质验证

对 3 个数据集分别进行距离性质验证。考虑到模型数量较多，这里只列出数据集一的 10 个模型的计算数据，如表 3 所示。

下面通过这 3 个数据集的实验数据验证 TAR++度量满足自反性、非负性、三角不等式 3 个特性。

自反性和非负性显而易见，不过多阐述。对于三角不等式的验证，除去自反性重复的部分以及自身和自身的相似性值，该数据集中共有 9+8+7+…+1=45 个相似性值。用 1 减去这些值，得到所有 45 个距离值，抽取其中 3 个进行三角不等式检验，共有 A_{10}^3=720 组。经验证，所有 3 个数据集共 720×3=2 160 组均满足三角不等式性质。

表 2　数据集简介

数据集	来源	数量	构成
数据集一	SAP 公司+人工编纂	80	10 个带循环、10 个带重名任务、10 个带不可见任务、10 个带嵌套循环、10 个带非自主选择结构、10 个短循环、10 个带并发、10 个带互斥、10 个顺序结构
数据集二	东锅公司实际流程	36	根据实际业务含义分为 5 个小组的流程,组内流程语义相似,小组内部模型的数量分别为 9,12,9,2,4
数据集三	北车公司实际流程	63	根据实际业务含义分为 4 个小组的流程,组内流程语义相似,小组构成为 12 个内部业务+32 个采购相关+7 个文档相关+12 个项目相关

表 3　SAP 模型集合一

模型序号	1	2	3	4	5	6	7	8	9	10
1	1.000	0.286	0.289	0.607	0.241	0.290	0.040	0.472	0.704	0.481
2	0.286	1.000	0.182	0.200	0.054	0.387	0.036	0.217	0.256	0.162
3	0.289	0.182	1.000	0.300	0.265	0.270	0.031	0.250	0.359	0.567
4	0.607	0.200	0.300	1.000	0.114	0.229	0.000	0.556	0.633	0.387
5	0.241	0.054	0.265	0.114	1.000	0.172	0.053	0.068	0.176	0.478
6	0.290	0.387	0.270	0.229	0.172	1.000	0.043	0.214	0.333	0.407
7	0.040	0.036	0.031	0.000	0.053	0.043	1.000	0.000	0.034	0.043
8	0.472	0.217	0.250	0.556	0.068	0.214	0.000	1.000	0.500	0.275
9	0.704	0.256	0.359	0.633	0.176	0.333	0.034	0.500	1.000	0.467
10	0.481	0.162	0.567	0.387	0.478	0.407	0.043	0.275	0.467	1.000

3.3　相似性五个性质的比较

上文已经指出,相似性算法应该至少满足 5 个性质,本节通过实验比较各算法对 5 个性质的满足情况。

性质 1　顺序结构漂移不变性。如图 6 所示,对于顺序结构,无论在哪两个变迁之间增加变迁,新模型与原模型的相似性均相同,即 $Sim(N_1,N_2)=Sim(N_1,N_3)=Sim(N_1,N_4)$。

实验模型如图 6 所示,将 N_1 中的变迁数增加至 10 个,然后分别将新增加的变迁插入 11 个不同的间隙中(包括首尾),得到 $N_2\sim N_{12}$ 共 11 个模型,分别比较 N_1 和 $N_2\sim N_{12}$ 的相似性值,计算结果如表 4 所示。

表 4　不同算法性质 1 的比较结果

	N_2	N_3	N_4	N_5	N_6	N_7	N_8	N_9	N_{10}	N_{11}	N_{12}
TAR	0.900	0.727	0.727	0.727	0.727	0.727	0.727	0.727	0.727	0.727	0.900
CF	0.836	0.836	0.836	0.836	0.836	0.836	0.836	0.836	0.836	0.836	0.836
BP	0.864	0.864	0.864	0.864	0.864	0.864	0.864	0.864	0.864	0.864	0.864
PTS	0.909	0.909	0.909	0.909	0.909	0.909	0.909	0.909	0.909	0.909	0.909
SSDT	0.917	0.917	0.917	0.917	0.917	0.917	0.917	0.917	0.917	0.917	0.917
TAR++	0.786	0.786	0.786	0.786	0.786	0.786	0.786	0.786	0.786	0.786	0.786

由表 4 可以看出,随着新增加变迁的位置变化,各模型和原模型之间的 TAR++相似性数值并没有发生变化,维持在 0.786 这个恒定值,由此验证了 TAR++相似性满足性质 1。此外,CF,BP,PTS,SSDT 等算法也能较好地满足该性质,只有 TAR 算法随着变迁位置的变化其相似性发生了变化。研究发现,只有在流程模型的起始变迁前或终止变迁后增加变迁才会导致 TAR 相似性变化,这主要是由于 TAR 算法没有通过人工增加起始变迁和终止变迁造成的,该问题在 TAR++算法中得到了改进。

性质 2　互斥结构漂移不变性。如图 7 所示,对于互斥结构,无论在哪个变迁上增加互斥分支,新模型与原模型的相似性均相同,即 $Sim(N_1,N_2)=Sim(N_1,N_3)=Sim(N_1,N_4)$。

实验模型如图 7 所示,将 N_1 中的变迁数增加至 10 个,然后分别将新增加的互斥分支加到这 10

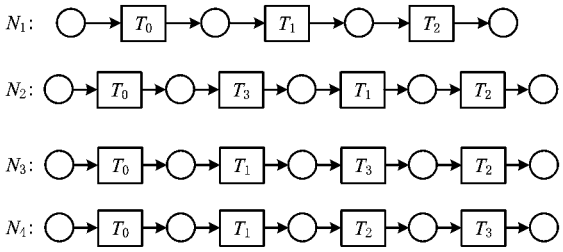


图6 顺序结构漂移不变性

个变迁上，得到 $N_2 \sim N_{11}$ 共 10 个模型，分别比较 N_1 和 $N_2 \sim N_11$ 的相似性值，计算结果如表 5 所示。

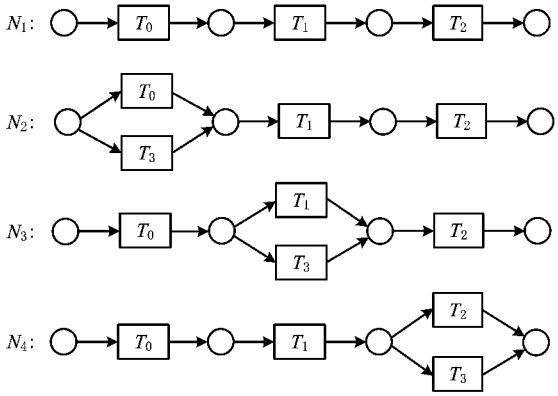


图7 互斥结构漂移不变性

表 5 不同算法性质 2 的比较结果

	N_2	N_3	N_4	N_5	N_6	N_7	N_8	N_9	N_{10}	N_{11}
TAR	0.900	0.818	0.818	0.818	0.818	0.818	0.818	0.818	0.818	0.900
CF	0.825	0.827	0.829	0.830	0.831	0.831	0.830	0.829	0.827	0.825
BP	0.801	0.801	0.801	0.801	0.801	0.801	0.801	0.801	0.801	0.801
PTS	0.967	0.967	0.967	0.967	0.967	0.967	0.967	0.967	0.967	0.967
SSDT	0.909	0.909	0.909	0.909	0.909	0.909	0.909	0.909	0.909	0.909
TAR++	0.846	0.846	0.846	0.846	0.846	0.846	0.846	0.846	0.846	0.846

由表 5 可以看出，随着新增加互斥分支的位置变化，各模型和原模型之间的 TAR++相似性数值并没有发生变化，维持在 0.846 这一恒定值，由此验证了 TAR++相似性满足互斥结构漂移不变性。而观察其他几个算法，PTS 和 SSDT 的相似性数值也没有变化，均能很好地满足该性质；TAR 算法依然在开始节点和终止节点处出现了相似性变化问题，该问题和性质 1 同样是由于没有人工增加起始变迁和终止变迁所致；CF 算法相似性随着互斥分支的位置变化呈现先上升后下降的变化趋势，证明 CF 不能满足性质 2。

性质 3 跨度负相关性。如图 8 所示，在原模型上增加互斥分支，分支跨越的变迁数越多，新模型与原模型的相似性越小，即 $Sim(N_1, N_2) > Sim(N_1, N_3)$。

实验模型如图 8 所示，将 N_1 中的变迁数增加

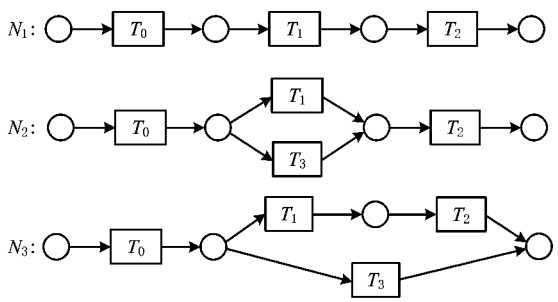


图8 跨度负相关性

至 10 个，然后分别将新增加的互斥分支从只跨越第一个变迁开始，慢慢改变其跨越的长度，即模型 N_2 只跨越 T_0 一个变迁，模型 N_3 跨越 T_0 和 T_1 两个变迁，模型 N_4 跨越 T_0，T_1 和 T_2 三个变迁，以此类推，直至 N_{11} 跨越全部 10 个变迁，得到 $N_2 \sim N_{11}$ 共 10 个模型，分别比较 N_1 和 $N_2 \sim N_{11}$ 的相似性值，计算结果如表 6 所示。

表 6 不同算法性质 3 的比较结果

	N_2	N_3	N_4	N_5	N_6	N_7	N_8	N_9	N_{10}	N_{11}
TAR	0.900	0.900	0.900	0.900	0.900	0.900	0.900	0.900	0.900	1.000
CF	0.825	0.795	0.775	0.764	0.763	0.770	0.786	0.812	0.849	0.898
BP	0.801	0.758	0.727	0.704	0.688	0.677	0.669	0.664	0.662	0.661
PTS	0.967	0.933	0.900	0.867	0.833	0.800	0.767	0.733	0.700	0.667

续表 6

SSDT	0.909	0.901	0.893	0.884	0.876	0.868	0.860	0.851	0.843	0.835
TAR++	0.833	0.792	0.750	0.708	0.667	0.625	0.583	0.542	0.500	0.458

由表 6 可以看出,随着新增加互斥分支跨越变迁数的逐渐增多,形成的新模型和原模型的 TAR++相似性值在不断降低,由此验证了 TAR++相似性满足跨度负相关性。观察其他几个算法,PTS 和 SSDT 同样也满足该性质;TAR 算法依然在流程模型的终止变迁处出现相似性升高的问题;CF 算法相似性随着互斥分支跨越变迁数的增多呈现先下降再上升的趋势,不满足性质 3;BP 的情况比较特殊,虽然从图 8 可以看出 BP 算法可以满足跨度负相关性,但实际上这只是个特例。再次考察图 8 所示的模型 N_1,N_2 和 N_3,并且用 N_1 和 N_2 的相似性与 N_1 和 N_3 的相似性进行比较,计算上述 6 个算法在该数据集合上的相似性变化情况,如表 7 所示。

表 7　图 8 中模型相似性计算结果

	$Sim(N_1,N_2)$	$Sim(N_1,N_3)$
TAR	0.500	0.667
CF	0.722	0.744
BP	0.460	0.525
PTS	0.889	0.778
SSDT	0.750	0.688
TAR++	0.600	0.500

通过表 7 不难发现,在该数据集上,模型 N_1 与 N_3 的相似性小于 N_1 与 N_2 的相似性,由此可见 BP 算法并不满足跨度负相关性。同样,不满足跨度负相关性的依然包括 TAR 和 CF 算法。

性质 4　非替代无关递减性。如图 9 所示,在原模型同一个变迁上增加分支,增加的分支数越多,新模型与原模型的相似性越小,即 $Sim(N_1,N_2)>Sim(N_1,N_3)$。

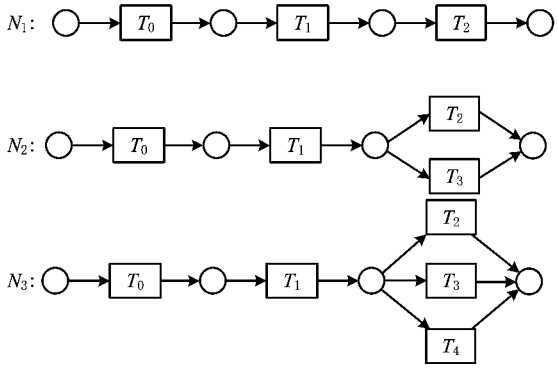


图9　非替代无关递减性

实验模型如图 9 所示,在 N_1 中 T_2 的基础上增加分支,分别增加 1,2,…,10 个分支作为 N_2 和 N_3 直至 N_{11},共 10 个模型,分别比较 N_1 和 N_2～N_{11}的相似性值,计算结果如表 8 所示。

表 8　不同算法性质 4 的比较结果

	N_2	N_3	N_4	N_5	N_6	N_7	N_8	N_9	N_{10}	N_{11}
TAR	0.667	0.500	0.400	0.333	0.286	0.250	0.222	0.200	0.182	0.167
CF	0.717	0.601	0.508	0.432	0.370	0.319	0.275	0.237	0.206	0.178
BP	0.550	0.351	0.250	0.192	0.155	0.129	0.111	0.097	0.086	0.077
PTS	0.889	0.833	0.800	0.778	0.762	0.750	0.741	0.733	0.727	0.722
SSDT	0.750	0.600	0.500	0.429	0.375	0.333	0.300	0.273	0.250	0.231
TAR++	0.600	0.500	0.429	0.412	0.400	0.391	0.385	0.379	0.375	0.371

由表 8 可以看出,随着在 T_1 上增加分支数目的增多,其形成的新模型和原模型的 TAR++相似性值不断降低,由此验证了 TAR++相似性满足非替代无关递减性。同时,TAR,CF,BP,PTS 和 SSDT 也都满足此性质。

性质 5　循环序列长度负相关性。如图 10 所示,在原模型上增加循环分支,循环分支跨越的变迁数越多,新模型与原模型的相似性越小,即 $Sim(N_1,$

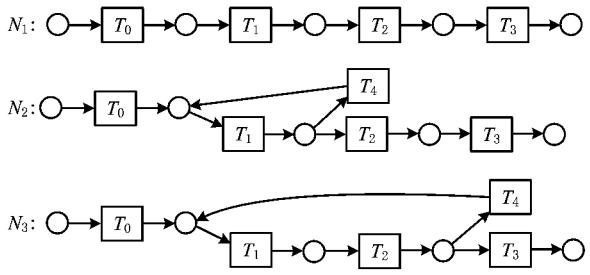


图10　循环序列长度负相关性

$N_2)>Sim(N_1,N_3)$。

实验模型如图 10 所示，将 N_1 中的变迁数增加至 10 个，然后在原模型上增加循环。让循环从跨越 T_1 变迁开始，慢慢改变其跨越的长度，即模型 N_2 只跨越 T_1 一个变迁，模型 N_3 跨越 T_1 和 T_2 两个变迁，模型 N_4 跨越 T_1，T_2 和 T_3 三个变迁，以此类推，直至 N_{10}跨越 T_1，T_2，T_3，…，T_9 共 9 个变迁，得到 N_2～N_{10}共 9 个模型，分别比较 N_1 和 N_2～N_{10}的相似性值，计算结果如表 9 所示。

表 9 不同算法性质 5 的比较结果

	N_2	N_3	N_4	N_5	N_6	N_7	N_8	N_9	N_{10}
TAR	0.818	0.818	0.818	0.818	0.818	0.818	0.818	0.818	0.818
CF	0.779	0.773	0.775	0.780	0.789	0.799	0.809	0.820	0.732
BP	0.948	0.916	0.889	0.864	0.842	0.823	0.805	0.789	0.774
PTS	0.525	0.525	0.525	0.525	0.525	0.525	0.525	0.525	0.525
SSDT	0.893	0.868	0.835	0.793	0.744	0.686	0.620	0.545	0.463
TAR++	0.846	0.786	0.733	0.688	0.647	0.611	0.579	0.550	0.524

由表 9 可以看出，随着新增加的循环分支跨越变迁数的增加，其形成的模型和原模型的 TAR++相似性值不断降低，由此验证了 TAR++相似性满足循环序列长度负相关性。观察其他几个算法的情况，TAR 相似性和 PTS 相似性都随着循环分支跨越变迁数的增加保持不变，CF 相似性则先上升再下降，这些算法都不满足该性质；除了 TAR++外，只有 SSDT 和 BP 算法满足性质 5。

综上所述，得到 TAR，PTS，SSDT，CF 和 TAR++算法对 5 个性质的满足情况如表 10 所示。

表 10 各算法对 5 个性质的满足情况

性质	顺序结构漂移不变性	互斥结构漂移不变性	跨度负相关性	非替代无关递减性	循环序列长度负相关性
TAR	×	×	×	√	×
PTS	√	√	√	√	×
SSDT	√	√	√	√	√
CF	√	×	×	√	×
BP	√	√	×	√	√
TAR++	√	√	√	√	√

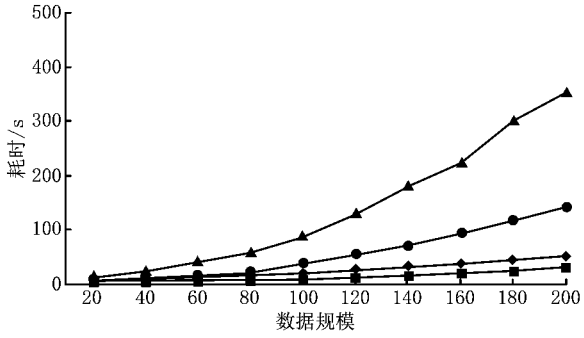
由表 10 不难发现，除了 SSDT 和 TAR++相似性，其他算法都不能满足所有 5 个性质。其中，TAR 算法只满足性质 4，主要因为 TAR 算法仅使用变迁之间最简单的两两邻接关系来构造比较集合；PTS 算法可以满足大部分性质，只有循环序列长度负相关性不能满足，这主要源自 PTS 算法对循环这种结构处理得不够好；CF 算法也是只能满足性质 1 和性质 4。除了 TAR++，唯一能满足所有性质的 SSDT 算法在性能和灵活性上多有不便，将在下文详述。

3.4 算法性能比较

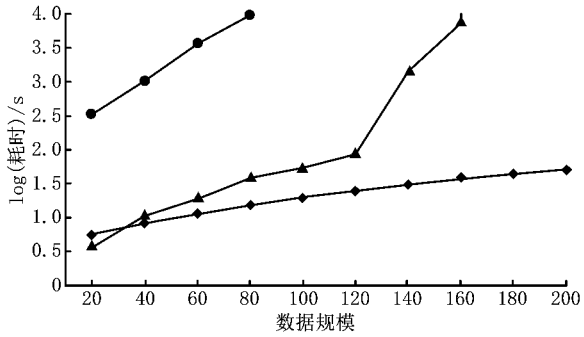
下面通过实验测试算法在实际数据集上的性能表现。

将本章用到的数据集一～数据集三作为测试性能的基准数据集，这三个数据集的大小分别为 80，36 和 63 共 179 个模型。另外，再从之前进行相似性五个性质测试的人工模型中去除 21 个模型，共 200 个模型。为了更直观地展现各个算法的性能优劣，将上述 200 个流程模型分别用 TAR，CF，BP，PTS，SSDT 和 TAR++算法进行相似性计算，并作图表示。由于实验结果中 CF 算法和 PTS 算法的时间消耗明显大于其他算法，并且在模型数量为 200 时都不可等待(大于 10 000 s)，为了清晰表示，首先在同一个图上绘制时间消耗在一个数量级上的

TAR,BP,SSDT 和 TAR++算法(如图 11a),再在另一个图上绘制取对数的 CF,PTS,TAR++算法的时间消耗(如图 11b)。



a TAR++与TAR, BP, SSDT的性能比较



b TAR++与CF, PTS的性能比较

图11　各算法性能比较

从图 11 不难看出,CF 和 PTS 的效率明显低于其他算法,其中 PTS 算法在模型数据量比较小时性能表现很好,但是在模型数量从 100 个增加到 120 个时产生了比较大的阶跃,即产生了数量级的变化,这主要是因为增加的这 20 个流程中大多数包括了很多并发和循环结构,而 PTS 处理并发结构的效率非常低。从 TAR++与其他算法的比较来看,只有 TAR 算法比 TAR++算法性能更好,主要是因为 TAR++在进行工作流网边系数确定时多进行了一次遍历。尽管如此,TAR++算法性能还是优于除 TAR 以外的其他算法。

4　结束语

本文介绍了一种基于 TAR 重要性的流程相似性算法,该算法不但满足距离度量空间的四个特性,还满足相似性算法应该满足的五个基本性质;同时,该算法也能处理包含诸如重名任务和非自主选择结构等特殊结构的模型。算法的复杂度和工作流网中具有最大回边数 N 的节点相关,为 $O(E+V+N!)$。对于相似性算法评估,本文借鉴前人的工作,总结出了相似性算法应该满足的顺序结构漂移不变性、互斥结构漂移不变性、跨度负相关性和循环序列长度负相关性 5 个特性,一方面从理论上说明了算法确实满足这些特性,另一方面通过具体实验数据和图表的形式直观验证了这一特征。

在未来工作中,还需进一步挖掘相似性度量的特征,扩展相似性算法应该满足的性质,并据此对 TAR++算法进行优化,使得相似度度量数值更加贴近领域专家的评估;另外,还需要在企业界对 TAR++算法进行推广。

参考文献:

[1] SHAFER S M, SMITH H J, LINDER J C. The power of business models [J]. Business Horizons, 2005, 48(3): 199-207.

[2] YAN Z, DIJKMAN R, GREFEN P. Fast business process similarity search with feature-based similarity estimation [M]//On the Move to Meaningful Internet Systems: OTM 2010. Berlin, Germany: Springer-Verlag, 2010: 60-77.

[3] BECKERA M, LAUEB R. A comparative survey of business process similarity measures[J]. Computers in Industry, 2012, 63(2): 148-167.

[4] WANG Shuhao, WEN Lijie, WEI Daisen, et al. SSDT matrix-based behavioral similarity algorithm for process model[J]. Computer Integrated Manufacturing Systems, 2013, 19(8): 1822-1831(in Chinese). [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.]

[5] ZHA Haiping, WANG Jianmin, WEN Lijie, et al. A workflow net similarity measure based on transition adjacency relations[J]. Computers in Industry, 2010, 61(5): 463-471.

[6] VAN DONGEN B F, MENDLING J, VAN DER AALST W M P. Structural patterns for soundness of business process models[C]//Proceedings of the 10th IEEE International Conference on Enterprise Distributed Object Computing. Washington, D. C., USA: IEEE, 2006: 116-128.

[7] WEIDLICH M, ELLIGER F, WESKE M. Generalised computation of behavioural profiles based on petri-net unfoldings [M]//Web Services and Formal Methods. Berlin, Germany: Springer-Verlag, 2011: 101-115.

[8] WANG J, HE T, WEN L, et al. A behavioral similarity measure between labeled Petri nets based on principle transition sequences[M]//On the Move to Meaningful Internet Systems: OTM 2010. Berlin, Germany: Springer-Verlag, 2010: 394-401.

[9] VAN DER AALST W M P. The application of Petri nets to workflow management[J]. Journal of Circuits, Systems and Computers, 1998, 8(1): 21-66.

[10] WANG Wenxing, WANG Jianmin. TAR*: an improved

process similarity measure based on unfolding of Petri nets [J]. Computer Integrated Manufacturing Systems, 2012, 18 (8):1774-1784.

作者简介:

殷 明(1989—),男,江苏东台人,硕士研究生,研究方向:业务流程相似性度量;

+闻立杰(1977—),男,河北唐山人,副教授,博士,研究方向:流程数据管理、工作流技术、过程挖掘,通信作者,E-mail:wenlj@tsinghua.edu.cn;

王建民(1968—),男,吉林磐石人,教授,博士生导师,研究方向:大数据管理、信息系统与工程、流程数据管理与挖掘;

肖 汉(1992—),男,北京人,本科生,研究方向:业务流程模型库;

丁子哲(1981—),男,内蒙古呼和浩特人,高级工程师,博士,研究方向:智能业务流程管理;

高 翔(1956—),男,辽宁沈阳人,教授级高工,博士,研究方向:云计算、计量经济学、智能业务流程管理。

“基于大数据和云技术的智慧制造”征文通知

随着物联网、云计算等新信息技术的广泛应用,在制造业的产品设计、生产和服务过程中将产生大量的数据,即所谓的大数据。其中物联网(如 RFID,无线传感器网络)主要用于收集无所不在的数据;云服务(如云制造服务数据)则作为服务/数据仓库为制造业提供有用的服务/数据。

充分、有效地利用大数据,可以提高制造企业的创新能力、促进网络智能制造的发展,进而为整个制造业的转型升级带来巨大推进力。然而,目前在整个产品生命周期中产生的制造大数据的类型尚属未知,如何从如此庞大的动态大数据中提取有用的信息并加以利用,是一项十分艰巨的任务,因此在云技术或云服务平台的支撑下,如何有效地管理和利用大数据并服务于制造业,是一个巨大挑战。近年来,已有诸多专家和学者对将大数据和云技术应用到工业生产中进行了大量研究。

为总结这些研究成果,《计算机集成制造系统》期刊拟开展“基于大数据和云技术的智慧制造”专刊征文活动,旨在报道大数据与云技术应用到工业中的相关方法、技术、系统等的最新研究成果和进展,热忱欢迎相关领域的专家和学者踊跃投稿。

一、征文范围

1. 智慧制造模式;
2. 云制造中的大数据;
3. 数据驱动的制造系统;
4. 基于大数据和云技术的个性化设计与制造;
5. 供应链管理中的大数据与云技术;
6. 基于云平台的 3D 打印;
7. 基于云平台的智能工厂;
8. 制造知识管理;
9. 制造服务管理;
10. 工业大数据挖掘算法;
11. 制造系统智能物联技术;
12. 基于信息物理融合系统的工业应用;
13. 基于大数据和云技术的绿色设计与制造。

二、征文要求

1. 论文必须为原创,并且未被其他会议、期刊录用或发表;
2. 论文可以是围绕主题的综述论文、技术方法论文和应用性论文,具有一定的创新性和前瞻性;
3. 论文格式请直接参照《计算机集成制造系统》期刊论文的格式要求,或访问期刊主页 http:www.cims-journal.cn 查询稿件格式要求。

三、投稿方式和截止日期

1. 通过《计算机集成制造系统》期刊网站:http://www.cims-journal.cn 进行投稿,稿件类型请务必注明“智慧制造”;
2. 投稿截止日期为 2015 年 8 月 31 日。

四、稿件录取

1. 所有来稿均需经两位以上同行专家评审,合格后方可录用。
2. 录用论文将集中在《计算机集成制造系统》以专刊的形式发表(Ei 100%收录)。

五、投稿联系方式

联系人:杨璐 通信地址:北京 2413 信箱 34 分箱 CIMS 编辑部

邮政编码:100089 联系电话:(010)68962468—3;68962479

电子邮件:cims@onet.com.cn