

基于矩阵的工作流逻辑网模型的化简及验证

郑 红 邓文轩 邓 晓 卢兴见

(华东理工大学信息科学与工程学院 上海 200237)

摘 要 在分析工作流时,Petri 网是一种有效的建模工具,但在处理复杂的工作流时,其易引发“状态空间爆炸”问题。工作流逻辑作为工作流路径的逻辑框架,可实现对工作流网的进一步抽象。为了对规模较大的工作流的畅通性进行验证,利用 Petri 网对工作流对应的逻辑网进行建模,在此基础上,提出了一个基于矩阵的工作流逻辑网化简算法,为大规模的工作流自动化化简提供了理论基础。通过对银行选址的工作流逻辑网畅通性的验证,表明了该算法在解决实际问题时的有效性。

关键词 工作流,关联矩阵,化简,银行选址

中图分类号 TP391.4 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2018.07.052

Simplification and Verification of Matrix-based Workflow Logic Net Model

ZHENG Hong DENG Wen-xuan DENG Xiao LU Xing-jian

(School of Information Science and Engineering, East China University of Science and Technology, Shanghai 200237, China)

Abstract Petri net is used as an effective modeling tool when analyzing workflows, but it is easy to cause “state space explosion” problem when dealing with complex workflow. Workflow logic, as a logical framework for workflow paths, enables further abstraction of workflow networks. In order to verify the smoothness of the larger workflow, the Petri net is used to model the logic network corresponding to the workflow. On this basis, a matrix-based workflow logic algorithm was proposed, which provides a theoretical basis for automatic simplification of the large-scale workflow. At last, this algorithm was applied to the bank location to verify the smoothness of its workflow logic net, which reflects the effectiveness of the algorithm in solving the practical problem.

Keywords Workflow, Incidence matrix, Simplification, Bank location

1 引言

工作流通过将工作活动分解成定义良好的任务、角色、规则和过程来完成执行和监控,以达到提高生产组织水平和工作效率的目的,是实现流程执行和控制管理的一条有效途径。工作流作为业务流程的形式化描述,包含描述任务间依赖关系的工作流逻辑以及在此逻辑上增加了显性内容的工作流语义。工作流逻辑与案例属性无关,用于描述所有案例共享的操作路径框架。而工作流语义则是针对具体案例的,其依靠显性内容,在工作流逻辑定义下的路径框架上选择该案例的实际路径^[1]。工作流逻辑为工作流的核心,其合理性成为了针对特定案例的实际路径的先决条件。

目前,实现规模较大或结构较复杂的工作流模型的可行性及其验证存在很大的困难,因此从理论上验证所建立的工作流模型的合理性,研究如何评价和优化模型的方法显得尤为重要。由于工作流可分为工作流逻辑和依赖特定显性内容

的工作流语义,因此可将复杂的工作流分析和验证问题分解为工作流逻辑网和针对特定工作流语义的分析和验证问题^[2]。

Petri 网作为一种图形化的语言,支持工作流建模所需要的所有原语,便于学习交互,其坚实的数学理论基础为工作流模型的验证和分析提供了理论依据。因此,可以通过 Petri 网的关联矩阵对工作流逻辑网进行自动化分析和验证。工作流逻辑网只关心任务间的依赖关系,并不关心任务的具体操作内容,使复杂的工作流分析与繁琐的显性工作流语义分离开来。而从 Petri 网的角度来看,工作流逻辑网的变迁集由任务集构成,变迁由库所和流关系连接,通过托肯的流动体现顺序、并发和冲突等关联关系。

文献[3]提出了一种基于 Petri 网图形化简规则的业务过程模型分析与验证方法,但是随着工作流规模的增长,工作流模型的复杂程度迅速提升。文献[4]将工作流网在语义上等价转化为一种完全由矩阵和向量来表示的数学模型,利用矩

到稿日期:2017-05-25 返修日期:2017-07-07 本文受国家自然科学基金(61103115)资助。

郑 红(1973—),女,博士,副教授,主要研究方向为普适计算、Petri 网应用,E-mail:zhenghong@ecust.edu.cn(通信作者);邓文轩 男,硕士,主要研究方向为 Petri 网应用,E-mail:827963986@qq.com;邓 晓 男,硕士,主要研究方向为 Petri 网理论及应用,E-mail:745125420@qq.com;卢兴见 男,博士,讲师,主要研究方向为云计算,E-mail:luxj@ecust.edu.cn。

阵和向量的运算来验证其合理性。但在构建矩阵之前,需要添加额外的标记表来示与分支、或分支、与合并、或合并,以及建立输入/输出库所集合和输入/输出变迁集合的矩阵,其时间复杂度与模型可能出现的状态数成正比,对比较复杂的工作流网而言,甚至可能呈指数级增长。文献[5]通过 Petri 网关联矩阵的秩来判定 Petri 网的公平性。文献[6]对 Petri 网模型的结构有界性、守恒性、可重复性、协调性进行分析。上述文献都是基于 Petri 网关联矩阵的性质对模型结构进行分析,但对于有错的 Petri 网模型,虽然能够验证模型出现的错误,但却无法发现错误所在。文献[7]借助矩阵运算或矩阵上的初等变换来表示工作流逻辑网的化简规则,但未深入探究各规则的特性、规则间的联系和冲突以及化简规则的判定和化简顺序对工作流化简的影响。文献[8]和文献[9]分别探究了基于矩阵的化简在不同方面的应用。文献[10-12]分别对不同情形下的工作流网的合理性进行了验证。文献[13]提出了任务最短跟随距离矩阵的概念,基于该矩阵提出的行为相似性算法能够完整支持流程模型各类结构。目前,Petri 网的化简技术的研究重点多在 Petri 网化简理论的研究和模型验证上,但工作流逻辑网的化简目标是对规模较大或者结构复杂的场景进行化简分析,因此实现工作流逻辑网的自动化化简方案是很有必要的。本文将对工作流逻辑网的自动化化简方案进行算法设计。

本文第 2 节介绍了相关概念;第 3 节分别利用关联矩阵及预定义的数据结构对化简规则的判定和化简进行刻画,并根据哈夫曼思想确定对工作流逻辑网的化简规则的执行次序,以提高算法的执行效率;第 4 节给出了验证实例;最后总结全文。

2 相关概念

定义 1^[14] 设 $\Sigma=(S, T; F, K, W, M_0)$ 为有限 P/T 系统,并假定其基网 $(S, T; F)$ 是单纯的,即 $\forall x \in S \cup T: \cdot x \cap x \cdot = \emptyset$ 。由于 Σ 是有限的,不妨令 $S = \{s_1, s_2, s_3, \dots, s_n\}, T = \{t_1, t_2, \dots, t_n\}$,即所有库所和变迁均已排好顺序。

- 1)以库所集 S 为序标集的列向量 $V: S \rightarrow Z$,称做 Σ 的 S 向量,其中 Z 是整数集。
- 2)以变迁集 T 为序标集的列向量 $U: T \rightarrow Z$,称做 Σ 的 T 向量。
- 3)以 $S \times T$ 为序标集的矩阵 $C: S \times T \rightarrow Z$,称做 Σ 的关联矩阵,其矩阵元素 $C(s_i, t_j) = W(t_j, s_i) - W(s_i, t_j)$

定义 2 称 $p = (T_1, T_2, (a_1, a_2))$ 为同步器,其中 $T_1 = \{t_{11}, \dots, t_{1m_1}\}$ 和 $T_2 = \{t_{21}, \dots, t_{2m_2}\}$ 分别为 m_1 和 m_2 个任务的集合。如果 $T_1 = \cdot p, T_2 = p \cdot, K(p) = a_1 \times a_2, 1 \leq a_1 \leq m_1, 1 \leq a_2 \leq m_2, a_1$ 是 T_1 中能被 p 触发的变迁个数, a_2 是 T_2 中能被 p 触发的变迁个数,且 $W(t_1, p) = a_2, W(p, t_2) = a_1$,其中 $t_1 \in T_1, t_2 \in T_2$ 。

定义 3 有向 Petri 网 $PN=(P, T; F)$ 是工作流网络,当且仅当:

- 1) PN 中存在两个特殊库所:库所 i 是输入库所, $\cdot i = H$;库所 o 是输出库所, $o \cdot = H$ 。

- 2)如果在 PN 中库所 o 和 i 之间增加一个变迁 t_* 。(即 $t_* \cdot = \{o\}$ 并且 $\cdot t_* = \{i\}$),则得到的 PN' 是一个强连通的 Petri 网。

定义 4^[1] 称加权有向图 $WN=(P, T; F, K, W)$ 为同步网,如果 $(P, T; F)$ 为有限有向网,且 $\forall t \in T: \cdot t \neq \emptyset \wedge t \cdot \neq \emptyset \wedge \forall p \in P: \cdot p \neq \emptyset \wedge p \cdot \neq \emptyset \rightarrow p = (T_1, T_2, (a_1, a_2)) \wedge |\{p \mid \cdot p = \Phi\}| = 1 \wedge F^+ \cap (F^{-1})^+ = \emptyset \wedge \forall p: \cdot p = \emptyset \vee p \cdot = \emptyset \rightarrow K(p) = 1 \wedge \forall t \in \cdot p: W(t, p) = 1 \wedge \forall t \in p \cdot: W(p, t) = 1$ 。则只要 $\cdot p \neq \emptyset \wedge p \cdot \neq \emptyset, p$ 必须是同步器。其中 $|\{p \mid \cdot p = \Phi\}| = 1$ 表示只有一个前集为空的库所(称为入口库所,即逻辑上的唯一入口), $F^+ \cap (F^{-1})^+ = \emptyset$ 表示 WN 中无有向环。

定义 5^[1] 称同步网 $WN=(P, T; F, K, W)$ 为工作流逻辑网,如果令 $\langle \cdot = \{(t, t') \mid t, t' \in T \wedge E p \in P: t \in \cdot p \wedge t' \in p \cdot\}$,则 $\langle \cdot$ 满足: $\langle \cdot \cap \langle \cdot^2 = \emptyset$,且 $(T, \langle \cdot)$ 是一个连通图。

定义 6^[1] 称 $S=(P, T; F, K, W, M_0)$ 为任务集 $(TASK, \langle \cdot)$ 的工作流逻辑,记作 $WF_logic(TASK)$,或简称为 WF_logic ,如果 $(P, T; F, K, W)$ 为工作流网,则 $\forall p \in P: (\cdot p = \emptyset \rightarrow M_0(p) = 1) \wedge (\cdot p = \emptyset \rightarrow M_0(p) = 0)$,而且 $(T, \langle \cdot) = (TASK, \langle \cdot')$ 。

定义 7^[1] 工作流逻辑网 $S=(P, T; F, K, W, M_0)$ 的变迁规则是基于 P/T 系统的变迁规则添加两条约束:

- 1)每个变迁至多只能发生一次;
- 2)同步器 $p = (T_1, T_2, (a_1, a_2))$ 只有在 $M(p) = a_1 \times a_2$ 时才能同步授权给 T_2 变迁发生权。

由同步网的定义可知,针对工作流逻辑网的结构特征,除去入口和终点库所外,其他的均为同步器。通过采用局部化的方法,即通过找出畅通的结构,可以将工作流逻辑网化简为一个库所或者一个变迁,化简的过程不会改变原来的畅通性。如果最终的工作流逻辑网能够化简为一个库所,那么该工作流逻辑网一定是畅通的。

3 模型化简方法及其在关联矩阵中的化简

为实现化简规则的自动化执行,本文利用 Petri 网的关联矩阵来实现各个化简规则的描述和论证。本节给出了 Petri 网的关联矩阵的相关定义以及各化简规则在关联矩阵中的表现形式。

本节首先通过关联矩阵和关联矩阵定义的其他数据结构,对各化简规则在关联矩阵中的表现形式、判定和化简进行刻画,并利用哈夫曼编码思想确定各化简规则的执行顺序,以此设计工作流逻辑网化简算法的执行流程图。

3.1 基础数据结构的设计

- 1)工作流逻辑网关联矩阵 WL_net_Array

工作流逻辑 $WF_logic \Sigma=(P, T; F, K, W, M_0)$ 中基网 $(P, T; F)$ 为简单网,其中 $P = \{p_1, p_2, p_3, \dots, p_n\}, T = \{t_1, t_2, t_3, \dots, t_m\}$ 。以 $P \times T$ 为序标集的矩阵 A 称作 $WF_logic \Sigma$ 的关联矩阵,其矩阵元素为:

$$a_{ij} = A(p_i, t_j) = W(t_j, p_i) - W(p_i, t_j) \tag{1}$$

在关联矩阵 A 中,若 $a_{ij} > 0$ 表示 $W(t_j, p_i) > W(p_i, t_j)$, 则 p_i 为 t_j 的前驱,或 t_j 为 p_i 的后继;反之,若 $a_{ij} < 0$ 表示 $W(t_j, p_i) < W(p_i, t_j)$, 则 p_i 为 t_j 的后继,或 t_j 为 p_i 的前驱。为提高算法的效率以及对化简规则的畅通性进行验证,定义库所的前驱向量和后继向量,以及变迁的前驱向量和后继向量。因为库所前驱集 P_{pre} 和变迁后继集 T_{sub} 相等,库所后继集 P_{sub} 和变迁前驱集 T_{pre} 相等,所以只给出 P_{pre} 和 T_{sub} 的实现方式。

2) 库所前驱集 P_{pre} 及变迁后继集 T_{sub}

P_{pre_i} 为库所 p_i 对应的前驱向量,对于 $\forall j \in (1, m)$, if $a_{ij} > 0, P_{pre_{ij}} = 1, \text{ else } P_{pre_{ij}} = 0$ 。

3) 库所后继集 P_{sub} 及变迁前驱集 T_{pre}

P_{sub_i} 为库所 p_i 对应的后继向量,对于 $\forall j \in (1, m)$, if $a_{ij} < 0, P_{sub_{ij}} = 1, \text{ else } P_{sub_{ij}} = 0$ 。

4) 同步器(synchronizer)及类别向量 place-type

工作流逻辑网中,起始库所和终点库所由于均不连接业务流程中的环节,即起始库所无前驱变迁,终点库所无后继变迁,因此无法用同步器来刻画。其他库所前后均连接业务流程,都可使用同步器来刻画。因此在关联矩阵中,适当地描述同步器的定义是工作流网建模中至关重要的一环。

对于关联矩阵中任意的库所 p_i ,令 p_i 的前驱变迁数量 $m_{1i} = |P_{pre_i}|$, p_i 的后继变迁数量 $m_{2i} = |P_{sub_i}|$,如果 $m_{1i} > 0 \wedge m_{2i} > 0$,对于 $\forall j \in (1, m)$,如果 $P_{pre_{ij}} = 1$,有 $a_{ij} = s_{2i}$ 。对于 $\forall j \in (1, m)$,如果 $P_{sub_{ij}} = 1$,有 $a_{ij} = s_{1i}$,库所 p_i 可以用同步器表示为:

$$p_i = (T_{pre_i}, T_{sub_i}, (s_{1i}, s_{2i}))$$

其中, T_{pre_i} 为库所 p_i 的前驱变迁集合,因为 $T = [t_1, t_2, t_3, \dots, t_m]^T$,可知 $T_{pre_i} = T \cdot P_{pre_i}$ 。 T_{sub_i} 为库所 p_i 的后继变迁集合,且有 $T_{sub_i} = T \cdot P_{sub_i}$ 。

同步器分类的定义^[1]:

1) 串行同步器

若 $m_{1i} = m_{2i} = 1$,则同步器 p_i 为串行同步器。

2) 叉形器

若 $m_{1i} = 1, m_{2i} > 1$,则同步器 p_i 为叉形器。

①若 $1 < s_{2i} < m_{2i}$,则此叉形器为叉形选择器,简称叉选,

即 OR_{split} 。

②若 $s_{i2} = m_{2i}$,则此叉形器为叉形并行器,简称叉并,即

AND_{split} 。

3) 汇型器

若 $m_{1i} > 1, m_{2i} = 1$,则同步器 p_i 为汇型器。

①若 $1 < s_{1i} < m_{1i}$,则此汇型器为汇型选择器,简称汇并,

即 OR_{join} 。

②若 $s_{1i} = m_{1i}$,则此汇型器为汇型并行器,简称汇并,即

AND_{join} 。

4) 混型器

若 $m_{1i} > 1, m_{2i} > 1$,则同步器 p_i 为混型器。

①若 $s_{1i} < m_{1i} \wedge s_{2i} < m_{2i}$,则此混型器为双选混合,简称双选。

②若 $s_{1i} = m_{1i} \wedge s_{2i} < m_{2i}$,则此混型器为并选混合,简称并选。

③若 $s_{1i} < m_{1i} \wedge s_{2i} = m_{2i}$,则此混型器为选并混合,简称选并。

④若 $s_{1i} = m_{1i} \wedge s_{2i} = m_{2i}$,则此混型器为双并混合,简称双并。

对于同步器的类型,可定义枚举类向量 $type = \{ \text{null}, \text{串行器}, OR_{split}, AND_{split}, OR_{join}, AND_{join}, \text{双选}, \text{并选}, \text{选并}, \text{双并} \}$,依据上述同步器分类方式,可对工作流逻辑网中所有的库所进行分类,生成库所同步器类别向量 $place_type$ 。

3.2 化简算法的设计

根据上文定义的数据结构,工作流逻辑网的化简规则^[1]在矩阵中的化简形式、化简条件的判断和化简后的矩阵如下:

化简规则 1 令 $T_1 = \{t_{11}, \dots, t_{1m_1}\}$ 和 $T_2 = \{t_{21}, \dots, t_{2m_2}\}$ 为 m_1 和 m_2 个任务的集合,若 $p_1 = (T_1, T, (a, b))$, $p_2 = (T, T_2, (b, c))$,则可以化简为 $p = (T_1, T_2, (a, c))$ 。

在矩阵中,顺序对比库所的后继变迁集矩阵 P_{sub} 和前驱变迁集矩阵 P_{pre} ,若在 P_{sub} 中存在行 P_{sub_i} 与 P_{pre} 中的行 P_{pre_j} 相同的情况,则可对关联矩阵 A 进行化简:

1) 删除原关联矩阵 A 中 P_{sub_i} 或 P_{pre_j} 向量中元素值为 1 对应的列。

2) 通过相加和合并关联矩阵 A 中的行 i 和行 j 来替换原来的行 i ,删除行 j ,形成新的化简后的关联矩阵 A' 。

合并后库所 p_i 的前驱变迁向量 $P_{pre_i}' = P_{pre_i}$,后继变迁向量 $P_{sub_i}' = P_{sub_i}$ 。且对于化简后的关联矩阵 A' , $\forall k \in (1, m)$,如果 $P_{pre_{ik}} = 1$, $a_{ij}' = s_{2k}$;同理,对于 $\forall k \in (1, m)$,如果 $P_{sub_{ik}} = 1$,化简后的关联矩阵 A' 中 $a_{ij}' = s_{1k}$ 。

通过上述化简方法可知,化简的时间消耗主要在匹配 P_{sub} 和 P_{pre} 中相同的向量上, P_{sub} 和 P_{pre} 均为 $n \times m$ 型的矩阵,因此总的时间复杂度为 $O(n^3)$ 。满足化简规则 1 的关联矩阵及化简后的关联矩阵如表 1 和表 2 所列。

表 1 满足化简规则 1 的关联矩阵

Table 1 Incidence matrix satisfying simplification rule 1

	T_{11}	...	T_{1m}	T_1	...	T_k	T_{21}	...	T_{2n}
P_1	b		b	$-a$		$-a$	0		0
P_2	0		0	c		c	$-b$		$-b$

表 2 利用化简规则 1 化简后的关联矩阵

Table 2 Incidence matrix simplified with rule 1

	T_{11}	...	T_{1m}	T_{21}	...	T_{2n}
P	c		c	$-a$		$-a$

化简规则 2 $p = (\{t_1\}, \{t_2\}, (1, 1))$,只要 $t_1 \cap \cdot t_2 = \{p\}$,即可以消去 p ,将 t_1 和 t_2 合并为 t , $\cdot t = \cdot t_1 \cup \cdot t_2 - \{p\}$, $t \cdot = t_1 \cup t_2 - \{p\}$ 。

在矩阵中,遍历关联矩阵 A 的同步器类别向量,若在 $place_type$ 中存在库所 P_i 为串行器,则可对行 P_i 中包含的两个分别为 1 和 -1 的元素 A_{ik} 和 A_{ij} 做如下处理:

删除原关联矩阵 A 中的第 i 行,同时将 k 列和 j 列相加合并到 k 列中,删除 j 列,形成新的化简后的关联矩阵 A' 。可知化简规则 2 消耗的时间为按行遍历关联矩阵 A 的时间,且由于矩阵 A 为 $n \times m$ 型的矩阵,因此化简规则 2 的时间复杂度为 $O(n^2)$ 。

与此同时,在化简后的关联矩阵 A' 中,相加合并后得到

的 k 列变迁 t_k' 的前驱库所集 $\bullet t_k' = \bullet t_k + \bullet t_j - p_i$, 变迁 T_k' 的后继库所集为 $t_k' \bullet = \bullet t_k + \bullet t_j - p_i$ 。其中, $\bullet t_k = P \cdot T_{pre_i}, t_k \bullet = P \cdot T_{sub_i}, P = [p_1, p_2, p_3, \dots, p_n]'$ 为对应原关联矩阵的库所向量。满足化简规则 2 的关联矩阵和化简后的关联矩阵如表 3 和表 4 所列。

表 3 满足化简规则 2 的关联矩阵

Table 3 Incidence matrix satisfying simplification rule 2

	T_1	T_2
P_1	-1	0
P_2	1	0
P_3	1	-1
P_4	0	-1
P_5	0	1

表 4 利用化简规则 2 化简后的关联矩阵

Table 4 Incidence matrix simplified with rule 2

	T
P_1	-1
P_2	1
P_3	-1
P_4	1

化简规则 3 若变迁 t 和库所 p_1, p_2 满足 $t \in p_i \wedge \{t\} = \bullet p_2$ 且 $p_2 \neq \emptyset \vee p_i = \{t\}$, 则 p_1 和 p_2 完全可以合并为 $p, \bullet p = \bullet p_1, p \bullet = p_i \cup p_2 - \{t\}$ 。若 p_1 中有托肯, 则 p 中也有。注意, 当 $p_2 = \emptyset \wedge p_i \neq \{t\}$ 时不能化简。

在矩阵中遍历关联矩阵 A , 若存在列 T_j , 仅包含 a_{mj} 和 a_{nj} 两个元素, 值分别为 1 和 -1, 且 a_{mj} 对应的库所 p_n 对应的 $P_{sub_n} \neq \vec{0}$ 或 $P_{sub_n} = 1$, 则可删除原关联矩阵 A 中的 T_j 列, 同时将 m 行和 n 行进行相加合并到 m 行中, 并删除 n 行, 形成新的化简后的关联矩阵 A' 。可知化简规则 3 消耗的时间为遍历原关联矩阵 A 的所有列所花费的时间, 因此化简规则 3 的时间复杂度为 $O(n^2)$ 。

在化简后的关联矩阵 A' 中, 通过相加合并后得到的行 m 库所 P_m' 的前驱变迁集为 $\bullet p_m' = \bullet p_m$, 后继变迁集为 $p_m' \bullet = p_m \bullet + p_n \bullet - t_j$, 其中 $\bullet p_m = T \cdot P_{pre_n}, p_m \bullet = T \cdot P_{sub_n}, p_n \bullet = T \cdot P_{sub_n}$ 。满足化简规则 3 的关联矩阵和化简后的矩阵如表 5 和表 6 所列。

表 5 满足化简规则 3 的关联矩阵

Table 5 Incidence matrix satisfying simplification rule 3

	T_i
P_i	-1
P_j	1

表 6 利用化简规则 3 化简后的关联矩阵

Table 6 Incidence matrix simplified with rule 3

	T_i
P	0

化简规则 4 令 $T_1 = \{t_1, t_2, \dots, t_a\}, T_2 = \{t_1', t_2', \dots, t_a'\}$ 。

1) 若 $p_i = (\{t_i\}, T_2, (1, b)), i = 1, 2, \dots, a$, 则这些同步器可以合并为一个同步器 $p = (T_1, T_2, (a, b))$ 。

2) 若同步器 $p_i = (T_1, \{t_1'\}, (a, 1)), i = 1, 2, \dots, b$, 则这些同步器可以合并为一个同步器 $p = (T_1, T_2, (a, b))$ 。

化简规则 4-1 对于关联矩阵 A , 若库所后继集 P_{sub} 中存

在相同的行向量, 即库所存在相同的后继变迁 T_{sub_i} , 个数为 k , 且这 k 个库所均为 $p_i = (\{t_i\}, T_{sub_i}, (1, k))$ 类同步器, 则可对原关联矩阵 A 进行如下化简从而得到新的关联矩阵 A' 。

令满足上述条件的第一个库所为 p_i , 则可将该 k 个库所所在的行均加到 p_i 所在的行, 同时删除满足上述条件的除 p_i 外的库所所在的行。

在化简后的关联矩阵 A' 中, $\bullet p_i' \bullet = \bigcup_{i=1}^k \bullet p_i, p_i' \bullet = p_i \bullet$, 其中 $p_i (i \in (1, k))$ 为原关联矩阵中满足化简条件的库所。

化简规则 4-2 在关联矩阵 A 中, 若库所前驱集 P_{pre} 中存在相同的行向量, 即库所存在相同的后继变迁 T_{pre_i} , 个数为 k , 且这 k 个库所均为 $p_i = (T_{pre_i}, \{t_i\}, (k, 1))$ 类同步器, 则可进行与化简规则 4-1 相同的化简, 将原关联矩阵 A 化简为新的关联矩阵 A' 。且 $\bullet p_i' = \bullet p_i, p_i' \bullet = \bigcup_{i=1}^k p_i \bullet$, 其中 $p_i (i \in (1, k))$ 为原关联矩阵中满足化简条件的库所。可知化简规则 4 化简耗费的时间与在后继集 P_{sub} 中存在相同的行向量的个数成正比, 时间复杂度为 $O(n^3)$ 。满足化简规则 4 化简条件的关联矩阵以及化简后的矩阵如表 7-表 10 所列。

表 7 满足化简规则 4-1 的关联矩阵

Table 7 Incidence matrix satisfying simplification rule 4-1

	T_1	...	T_a	t_1	...	t_b
P_1	b		0	-1		-1
...						
P_a	0		b	-1		-1

表 8 利用化简规则 4-1 化简后的关联矩阵

Table 8 Incidence matrix simplified with rule 4-1

	T_1	...	T_a	t_1	...	t_b
P	b		b	$-a$		$-a$

表 9 满足化简规则 4-2 的关联矩阵

Table 9 Incidence matrix satisfying simplification rule 4-2

	T_1	...	T_a	t_1	...	t_b
P_1	1		1	$-a$		0
...						
P_b	1		1	0		$-a$

表 10 利用化简规则 4-2 化简后的关联矩阵

Table 10 Incidence matrix simplified with rule 4-2

	T_1	...	T_a	t_1	...	t_b
P	b		b	$-a$		$-a$

化简规则 5 令 $T = \{t_1, t_2, \dots, t_a\}, T_i = \{t_{i1}, t_{i2}, \dots, t_{ib}\}, i = 1, 2, \dots, a$ 。

化简规则 5-1 同步器 $p_i = (\{t_i\}, T_i, (1, b))$, 只要 T 中的所有变迁属于一个同步器 $p' = (T', T'', (a_1, a_2))$ 及 $T' \leq T''$, 则这些同步器可以化简为库所 $p, K(p) = a \cdot b, W(t_i, p) = b, i = 1, 2, \dots, a, W(p, t_{ij}) = 1, j = 1, 2, \dots, b$, 简写为 $p = (T, \bigcup T_i, (1, b))$ 。

化简规则 5-2 同样地, 若 $p_i = (T_i, \{t_i\}, (b, 1))$ 为并行器, 而且 $\bigcup_{i=1}^a T_i$ 属于并行器 $p' = (T', T'', (a_1, a_2))$, 使得 $\bigcup_{i=1}^a T_i \leq T''$, 则上述的 p_i 可以合并为 $p = (\bigcup T_i, T, (b, 1)), K(p) = a \cdot b$ 。

即在原关联矩阵 A 中,若存在连续的行满足以下条件:

1) p_i 行中只存在 1 个 b (或 $-b$), b 个 -1 (或 b 个 1),其他全为 0,且 $b \geq 2$ 。

2) 在关联矩阵满足上述条件的行向量中,当 $i \neq j$ 时,对于 $\forall k$ 均满足:如果 $s_{ik} \neq 0$,则 $s_{jk} = 0$ 。

则可对原关联矩阵做如下化简:令满足上述条件的第一个库所 p_i ,对于满足上述条件的 k 行,将各行相加合并到行 p_i ,同时删除 p_i 行以外的其他满足条件的 $k-1$ 行,生成新的关联矩阵 A' 。

在化简后的关联矩阵 A' 中,通过合并化简后生成的新的库所 p_i 的前驱变迁集为 $\cdot p_i' = \bigcup_{i=1}^k \cdot p_i$,后继变迁集为 $p_i \cdot' = \bigcup_{i=1}^k p_i \cdot$,其中 $p_i (i \in (1, k))$ 为原关联矩阵中满足化简条件的 k 个库所。可知化简规则 5 中,化简所需要花费的时间为 $O(n^3)$ 。满足化简规则 5 的关联矩阵和化简后的关联矩阵如表 11—表 14 所列。

表 11 满足化简规则 5-1 的关联矩阵

Table 11 Incidence matrix satisfying simplification rule 5-1

	T_1	...	T_a	T_{11}	...	T_{1b}	T_{a1}	...	T_{ab}
P_1	b		0	-1		-1	0		0
...									
P_a			b	0	0	0	-1		-1

表 12 利用化简规则 5-1 化简后的关联矩阵

Table 12 Incidence matrix simplified with rule 5-1

	T_1	...	T_a	T_{11}	...	T_{1b}	T_{a1}	...	T_{ab}
P_1	b		b	-1		-1	-1		-1

表 13 满足化简规则 5-2 的关联矩阵

Table 13 Incidence matrix satisfying simplification rule 5-2

	T_{11}	...	T_{1b}	T_{a1}	...	T_{ab}	T_1	...	T_a
P_1	1		1	0		0	$-b$		0
...									
P_a	0		0	1		1	0		$-b$

表 14 利用化简规则 5-2 化简后的关联矩阵

Table 14 Incidence matrix simplified with rule 5-2

	T_{11}	...	T_{1b}	T_{a1}	...	T_{ab}	T_1	...	T_a
P_1	1		1	1		1	$-b$		$-b$

化简规则 6

化简规则 6-1 同步器 $p_i = (\{t\}, \{t_i\}, (1, 1)), i = 1, 2, \dots, a$, 可以合并为同步器 $p = (\{t\}, \{t_1, t_2, \dots, t_a\}, (1, a))$ 。

化简规则 6-2 同步器 $p_i = (\{t_i\}, \{t\}, (1, 1)), i = 1, 2, \dots, a$, 可以合并为同步器 $p = (\{t_1, t_2, \dots, t_a\}, \{t\}, (a, 1))$ 。

即在原关联矩阵 A 中遍历库所前驱集 P_{pre} , 查找相同的行向量,若存在且满足条件的行向量对应的库所 p_i 均为串行同步器,则可对满足条件的库所在原关联矩阵中对应的行做如下化简:令 p_i 为原关联矩阵 A 中第一个满足条件的库所,则可将其他满足条件的行相加合并到 p_i 所在的行,并删除 p_i 之外的其他行,得到化简后的关联矩阵 A' 。

在新的关联矩阵 A' 中,新生成的库所 p_i 的前驱变迁为 $\cdot p_i' = \cdot p_i$,后继变迁为 $p_i \cdot' = \bigcup_{i=1}^k p_i \cdot$ 。其中 $p_i (i \in (1, k))$ 为满足原关联矩阵化简条件的库所。满足化简规则 6 的关联

矩阵及化简后的关联矩阵如表 15—表 18 所列。

表 15 满足化简规则 6-1 的关联矩阵

Table 15 Incidence matrix satisfying simplification rule 6-1

	T	T_1	...	T_a
P_1	-1	1		0
...				
P_a	-1	0		1

表 16 利用化简规则 6-1 化简后的关联矩阵

Table 16 Incidence matrix simplified with rule 6-1

	T	T_1	...	T_a
P	a	-1		-1

表 17 满足化简规则 6-2 的关联矩阵

Table 17 Incidence matrix satisfying simplification rule 6-2

	T_1	...	T_a	T
P_1	-1		0	1
...				
P_a	0		-1	1

表 18 利用化简规则 6-2 化简后的关联矩阵

Table 18 Incidence matrix simplified with rule 6-2

	T_1	...	T_a	T
P	-1		-1	a

变迁规则中,除化简规则 1 以外,其他规则涉及的都是串行器(顺序)和并行器(AND),因为串行器和并行器都是局部畅通的,所以都是可以化简的,而引起搁浅和遗留的情况都涉及选择器(OR),这是由选择器之间不匹配或者选择器与并行器之间不匹配引起的。为实现更高效的化简,针对矩阵运算的特点,在化简过程中优先使用能够快速判断关联矩阵中行列是否满足化简条件的化简规则。同时,在化简规则 6 中,当 $a=2$ 时,两个库所均满足化简规则 2 的条件,但如果优先使用化简规则 2,则会破坏原有结构,从而无法实现化简,因此在使用化简规则 2 之前须先查验关联矩阵能否使用化简规则 6。在利用关联矩阵化简的过程中,化简规则 4 和化简规则 5 在判定条件中的第一个步骤是一样的,因此在对原关联矩阵进行化简时,对于满足化简规则 4 和化简规则 5 第一步判定条件的行,依次判定是否符合化简规则 4 或化简规则 5。综上所述,本文将化简规则的顺序设定为化简规则 3→化简规则 6→化简规则 2→化简规则 4→化简规则 5→化简规则 1。

按照上述的工作流逻辑网化简规则的执行顺序,工作流逻辑网关联矩阵化简算法的执行流程如下:

Step1 输入工作流逻辑网关联矩阵 A 。

Step2 初始化关联矩阵 A 的库所前驱集矩阵 P_{pre} 、库所后继集矩阵 P_{sub} 、变迁前驱集矩阵 T_{pre} 及变迁后继集矩阵 T_{sub} ,同时对矩阵 A 中除起始库所和终点库所外的其他库所所属的同步器类别进行分类。

Step3 按照预先指定的工作流化简规则的执行顺序,判断关联矩阵 A 中是否存在符合化简规则的行和列。如果存在可执行的化简规则,则更新化简后的关联矩阵 A ,并跳至 Step2。如果不存在符合化简规则的行列,则执行 Step4。

Step4 判断关联矩阵 A 是否为 $\begin{pmatrix} -1 \\ 1 \end{pmatrix}$ 形式,若是则表明该工作流逻辑网是畅通的,否则表示该工作流逻辑网不畅通。

workflows logic net simplification process as shown in Figure 1.

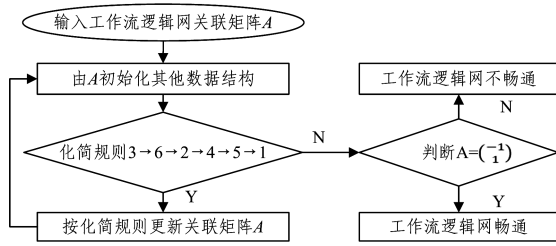


图1 workflows logic net simplification process flowchart

Fig. 1 Flowchart of WF logic net simplification

4 Model verification example

Bank branches are the most important resources for the development of banking business, especially for urban commercial banks. Due to its market scale, geography and technology, etc., urban commercial banks are more dependent on branches. However, urban commercial banks need to consider many factors when selecting branches, such as market capacity, population attributes (including population gender ratio, age level, consumption level, etc.) and consumption market; at the same time, other competitive bank branches, safety, etc. negative factors, how to choose more suitable branches among multiple candidates has an important influence. In addition, in the existing selected bank branches, how to evaluate the performance of bank branches, how to compare the performance of similar branches, etc. can be abstracted as the same operation path framework. For different situations, you can choose the actual path from the path framework, so as to realize the more efficient distribution of bank branches. According to the above situation, a workflow logic net can be established, as shown in Figure 2.

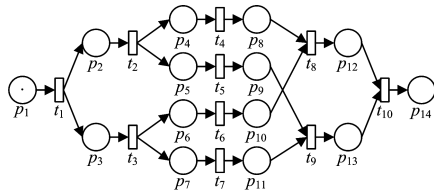


图2 银行选址的工作流逻辑网

Fig. 2 WF logic net of bank location

According to the above workflow logic net simplification process, the workflow logic net is simplified.

1) According to the workflow logic net in Figure 2, the corresponding incidence matrix A_0 is listed in Table 19.

表19 原工作流逻辑网的关联矩阵 A_0

Table 19 Incidence matrix A_0 of origin WF logic net

	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}
P_1	-1	0	0	0	0	0	0	0	0	0
P_2	1	-1	0	0	0	0	0	0	0	0
P_3	1	0	-1	0	0	0	0	0	0	0
P_4	0	1	0	-1	0	0	0	0	0	0
P_5	0	1	0	0	0	0	0	0	0	0
P_6	0	0	1	0	0	0	0	0	0	0
P_7	0	0	1	0	0	0	0	0	0	0
P_8	0	0	0	1	0	0	0	0	0	0
P_9	0	0	0	0	1	0	0	0	0	0
P_{10}	0	0	0	0	0	1	0	0	0	0
P_{11}	0	0	0	0	0	0	1	0	0	0
P_{12}	0	0	0	0	0	0	0	1	0	0
P_{13}	0	0	0	0	0	0	0	0	1	0
P_{14}	0	0	0	0	0	0	0	0	0	1

2) Initialize the precursor set matrix P_{pre} , the successor set matrix P_{sub} , the precursor set matrix T_{pre} and the successor set matrix T_{sub} . The initialization results are listed in Table 20, Table 21. Since the precursor set matrix T_{pre} and the successor set matrix T_{sub} are equal to P_{pre} and P_{sub} respectively, therefore this paper does not give them. For the incidence matrix A_0 , except for the starting place and the ending place, other places are classified into different categories, the classification results are listed in Table 22. $Type$ is an enumeration array recording the type of the place, the value of $type$ corresponds to the type in the $Type$ array. At this point, the initialization of the related data structure of the incidence matrix A_0 is completed.

表20 关联矩阵 A_0 的前驱集矩阵 P_{pre}

Table 20 Precursor set matrix P_{pre} of incidence matrix A_0

	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}
P_1	0	0	0	0	0	0	0	0	0	0
P_2	1	0	0	0	0	0	0	0	0	0
P_3	1	0	0	0	0	0	0	0	0	0
P_4	0	1	0	0	0	0	0	0	0	0
P_5	0	1	0	0	0	0	0	0	0	0
P_6	0	0	1	0	0	0	0	0	0	0
P_7	0	0	1	0	0	0	0	0	0	0
P_8	0	0	0	1	0	0	0	0	0	0
P_9	0	0	0	0	1	0	0	0	0	0
P_{10}	0	0	0	0	0	1	0	0	0	0
P_{11}	0	0	0	0	0	0	1	0	0	0
P_{12}	0	0	0	0	0	0	0	1	0	0
P_{13}	0	0	0	0	0	0	0	0	1	0
P_{14}	0	0	0	0	0	0	0	0	0	1

表21 关联矩阵 A_0 的库所后继集矩阵 P_{sub}

Table 21 Follow set matrix P_{sub} of incidence matrix A_0

	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}
P_1	1	0	0	0	0	0	0	0	0	0
P_2	0	1	0	0	0	0	0	0	0	0
P_3	0	0	1	0	0	0	0	0	0	0
P_4	0	0	0	1	0	0	0	0	0	0
P_5	0	0	0	0	1	0	0	0	0	0
P_6	0	0	0	0	0	1	0	0	0	0
P_7	0	0	0	0	0	0	1	0	0	0
P_8	0	0	0	0	0	0	0	1	0	0
P_9	0	0	0	0	0	0	0	0	1	0
P_{10}	0	0	0	0	0	0	0	0	1	0
P_{11}	0	0	0	0	0	0	0	0	0	1
P_{12}	0	0	0	0	0	0	0	0	0	1
P_{13}	0	0	0	0	0	0	0	0	0	1
P_{14}	0	0	0	0	0	0	0	0	0	0

表22 关联矩阵 A_0 中的库所分类向量

Table 22 Place class vectors of incidence matrix A_0

	$Type$
P_1	0
P_2	2
P_3	2
P_4	2
P_5	2
P_6	2
P_7	2
P_8	2
P_9	2
P_{10}	2
P_{11}	2
P_{12}	2
P_{13}	2
P_{14}	0

In Table 22, $type = \{ \text{null, 串行器, } OR_{split}, AND_{split}, OR_{join}, AND_{join}, \text{双选, 并选, 选并, 双并} \}$.

3)根据化简规则的顺序进行依次判断,可知原关联矩阵 A_0 中的 T_4, T_5, T_6, T_7 列均满足化简规则 3,因此可通过重复执行化简规则 3 来对原关联矩阵化简。与此同时,前驱集矩阵 P_{pre} 中的 P_{pre_2} 和 P_{pre_3} 相等,且 P_{pre} 中的 P_{pre_2} 和 P_{pre_3} 同步器类型均为串行器,满足化简规则 6,可再根据化简规则 6 的化简方式对上一步化简后的关联矩阵继续进行化简。原关联矩阵通过上述化简后生成新的关联矩阵 A_1 ,如表 23 所列。返回步骤 2),根据新生成的关联矩阵 A_1 重新初始化相关数据结构。

表 23 关联矩阵 A_0 经化简后的关联矩阵 A_1

Table 23 Incidence matrix A_1 by simplifying A_0

	T_1	T_2	T_3	T_4	T_5	T_6
P_1	-1	0	0	0	0	0
P_2	2	-1	-1	0	0	0
P_3	0	2	0	-1	-1	0
P_4	0	0	2	-1	-1	0
P_5	0	0	0	1	1	-2
P_6	0	0	0	0	0	1

按照相同的化简逻辑,可发现化简后的关联矩阵 A_1 对应的库所后继集 P_{sub} 中存在相同的行向量 P_{sub_1} 和 P_{sub_2} ,个数为 2,同时库所 P_3 和 P_4 均为 $s_{13} = s_{14} = 1$ 且 $s_{23} = s_{24} = 2$,满足化简规则 4。因此可对关联矩阵 A_1 化简,化简结果如表 24 所列。

表 24 关联矩阵 A_1 化简后的关联矩阵 A_2

Table 24 Incidence matrix A_2 by simplifying A_1

	T_1	T_2	T_3	T_4	T_5	T_6
P_1	-1	0	0	0	0	0
P_2	2	-1	-1	0	0	0
P_3	0	2	2	-2	-2	0
P_4	0	0	0	1	1	-2
P_5	0	0	0	0	0	1

在关联矩阵 A_2 对应的库所后继变迁集矩阵 P_{sub} 和前驱变迁集矩阵 P_{pre} 中存在 $P_{sub_2} = P_{pre_3}$,即存在库所 P_2 的后继向量和库所 P_3 的前驱向量相等,且 $s_{22} = 1, s_{13} = 2$,则化简后的关联矩阵 A_3 如表 25 所列。

表 25 关联矩阵 A_2 化简后的关联矩阵 A_3

Table 25 Incidence matrix A_3 by simplifying A_2

	T_1	T_2	T_3	T_4
P_1	-1	0	0	0
P_2	2	-1	-1	0
P_3	0	1	1	-2
P_4	0	0	0	1

关联矩阵 A_3 对应的库所后继变迁集矩阵 P_{sub} 和前驱变迁集矩阵 P_{pre} 中存在 $P_{sub_2} = P_{pre_3}$,即存在库所 P_2 的后继向量和库所 P_3 的前驱向量相等,且 $s_{22} = 1, s_{13} = 1$,则化简后的关联矩阵 A_4 如表 26 所列。

表 26 关联矩阵 A_3 化简后的关联矩阵 A_4

Table 26 Incidence matrix A_4 by simplifying A_3

	T_1	T_2
P_1	-1	0
P_2	1	-1
P_3	0	1

在关联矩阵 A_4 中, T_1 列对应的行 P_1 和 P_2 满足化简规则 3,则可继续对关联矩阵 A_4 化简,最后可得关联矩阵 $A_5 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$,如表 27 所列。至此,退出循环,上述工作流逻辑网是畅通的。

表 27 关联矩阵 A_4 化简后的关联矩阵 A_5

Table 27 Incidence matrix A_5 by simplifying A_4

	T_1
P_1	-1
P_2	1

结束语 工作流网被广泛地应用于企业建模,是实现流程执行和控制管理的一条有效途径,因此合理的工作流网对企业的重要性不言而喻。但由于复杂性等原因,实现工作流模型的可行性及其验证存在很大的困难,Petri 网作为一种拥有严谨的数学理论基础的图形化语言,经常被用于工作流网的建模和可行性分析与验证。随着工作流复杂程度的提高,构建的 Petri 网模型的复杂度也相应增长。工作流逻辑网作为工作流任务间依赖关系的描述,将工作流分解为工作流逻辑和在此逻辑上添加显性内容的工作流语义,可降低复杂工作流问题的分析和验证难度,对工作流逻辑网的畅通性验证可视为工作流合理性验证的先决条件。本文通过对工作流逻辑网化简规则的分析,利用工作流逻辑网的关联矩阵特性,使用关联矩阵及预定义的数据结构刻画各个化简规则的判断条件及化简方式,为工作流逻辑网化简的编程自动化实现提供了依据。最后,通过对银行选址问题构建的工作流逻辑网模型的验证,证明了本文方法的正确性。

参 考 文 献

[1] 袁崇义. Petri 网原理与应用[M]. 北京:电子工业出版社,2005: 230-238.

[2] HUANG Y Y, LI K Q. Survey: Legitimacy Verification of Workflow Based on Petri Nets[J]. Application Research of Computers, 2013, 30(1): 26-29. (in Chinese)
黄瑜岳, 李克清. 基于 Petri 网的工作流合法性验证综述[J]. 计算机应用研究, 2013, 30(1): 26-29.

[3] LIU L. Research on Workflow Modeling Method and its Application Based on Petri Networks[D]. Hangzhou: Zhejiang Science and Technology University, 2013. (in Chinese)
刘磊. 基于 Petri 网的工作流模型化方法及其应用研究[D]. 杭州: 浙江理工大学, 2013.

[4] GAO J, WU H R. Research of Matrix-based Workflow Model Rationality Validation Algorithm[J]. Computer Engineering and Design, 2010, 31(11): 2621-2624, 2628. (in Chinese)
高捷, 吴华瑞. 基于矩阵模型的工作流合理性验证算法研究[J]. 计算机工程与设计, 2010, 31(11): 2621-2624, 2628.

[5] WANG P L, JIANG C J. Use of Rank of Incidence Matrix for Judgement Fairness of Petri Nets[J]. Journal of Software, 1994 (12): 24-29. (in Chinese)
王培良, 蒋昌俊. 利用关联矩阵的秩判断 Petri 网的公平性[J]. 软件学报, 1994(12): 24-29.

- [6] LIAO J J, WANG M Z. Eigenvalues of Incidence Matrices Applied to the Analysis of Petri Net Structures[J]. *Journal of Applied Sciences*, 2010, 28(4): 417-423. (in Chinese)
廖晶静, 王明哲. 用关联矩阵特征值分析 Petri 网模型结构[J]. *应用科学学报*, 2010, 28(4): 417-423.
- [7] MENG Y G, SONG W, YE J H. Reducing of Workflow Logic Net Based on Petri Nets by Using Matrix [J]. *Control & Automation*, 2006, 22(21): 49-51. (in Chinese)
孟永刚, 宋文, 叶剑虹. 基于 Petri 网的工作流逻辑化简规则的矩阵表示[J]. *微计算机信息*, 2006, 22(21): 49-51.
- [8] CHEN W, DA SILVA R F, DEELMAN E, et al. Balanced task clustering in scientific workflows[C]// 2013 IEEE 9th International Conference on eScience (eScience). IEEE, 2013: 188-195.
- [9] CHEN X, ZHOU Y, YUE X B, et al. Research about New Technology for Petri Net's Simplification [J]. *Computer Engineering and Applications*, 2012, 48(5): 47-50. (in Chinese)
陈曦, 周彦, 乐晓波, 等. Petri 网化简新技术研究[J]. *计算机工程与应用*, 2012, 48(5): 47-50.
- [10] YUAN G, SUN R Z, XIANG Y, et al. Semantic Validation of Workflow Based Condition-restricted and its Application[J]. *Journal of Chinese Computer Systems*, 2013, 34(9): 2051-2055. (in Chinese)
袁钢, 孙瑞志, 向勇, 等. 基于条件约束的工作流语义验证方法及应用[J]. *小型微型计算机系统*, 2013, 34(9): 2051-2055.
- [11] LI Y. Rationality Analysis of Enterprise Business Process Based on Simplification Rules and Process Evaluation [J]. *China CIO News*, 2014(4): 16-17, 19. (in Chinese)
李燕. 基于化简规则与过程评价的企业业务过程合理性分析[J]. *信息系统工程*, 2014(4): 16-17, 19.
- [12] BOUCHEKIR R, BOUKHEDOUA S, BOUKALA M C. Symbolic probabilistic analysis and verification of inter-organizational workflow[C]// 2016 International Conference on Information Technology for Organizations Development (IT4OD). IEEE, 2016: 1-8.
- [13] WANG S H, WEN L J, WEI D S, et al. SSDT Matrix-based Behavioral Similarity Algorithm for Process Models [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.
- (上接第 298 页)
- [21] FRISTON K J. Statistical Parametric Mapping: The Analysis of Functional Brain Images [J]. *Neurosurgery*, 2013, 61(1): 216-216.
- [22] TZOURIOMAZOYER N, LANDEAU B, PAPATHANASSIOU D, et al. Automated anatomical labeling of activations in SPM using a macroscopic anatomical parcellation of the MNI MRI single-subject brain[J]. *Neuroimage*, 2002, 15(1): 273-289.
- [23] TEWARI EP, VAN D E, HILLEBRAND A, et al. The minimum spanning tree: an unbiased method for brain network analysis [J]. *Neuroimage*, 2015, 104: 177-188.
- [24] KRUSKAL J B. On the shortest spanning subtree of a graph and the traveling salesman problem[J]. *Proceedings of the American Mathematical Society*, 1956, 7(1): 48-50.
- [25] LOPEZ H C. Kolmogorov-Smirnov Test[J]. *International Encyclopedia of Statistical Science*, 2008, 10(1): 718-720.
- [26] BENJAMINI, YOAV, HOCHBERG Y, et al. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing [J]. *Royal Statistical Society. Series B (Methodological)*, 1995, 57(1): 289-300.
- [27] POLAJNAR M, DEMSAR J. Small network completion using frequent subnetworks [J]. *Intelligent Data Analysis*, 2015, 19(1): 89-108.
- [28] YAN X, HAN J. gSpan: Graph-Based Substructure Pattern Mining[C]// IEEE International Conference on Data Mining. IEEE Computer Society, 2002: 721.
- [29] LANCKRIETG R G, CRISTIANINI N, BARTLETT P, et al. Learning the Kernel Matrix with Semi-Definite Programming. [C]// Nineteenth International Conference on Machine Learning. Morgan Kaufmann Publishers Inc. 2002: 323-330.
- [30] SHERVASHIDZE N, SCHWEITZER P, VAN LEEUWEN E J, et al. Weisfeiler-Lehman Graph Kernels [J]. *Journal of Machine Learning Research*, 2011, 12(3): 2539-2561.
- [31] GUO H, LIU W Z, LIU Z F, et al. Difference index analysis on resting state functional brain network and its application in major depressive disorder classification [J]. *Computer Applications & Software*, 2014(12): 85-88. (in Chinese)
郭浩, 刘文钊, 刘志芬, 等. 静息态功能脑网络差异指标分析及抑郁症分类应用[J]. *计算机应用与软件*, 2014(12): 85-88.
- [32] SACCHETTI D, CAMACHO M C, LIVERMORE E E, et al. Accelerated aging of the putamen in patients with major depressive disorder [J]. *J Psychiatry Neurosci*, 2017, 42(3): 164.
- [33] QIAO L, ZHANG H, KIM M, et al. Estimating functional brain networks by incorporating a modularity prior [J]. *Neuroimage*, 2016, 141: 399-407.
- [34] M-L W, DONG C, ANDREEV V, et al. Prediction of susceptibility to major depression by a model of interactions of multiple functional genetic variants and environmental factors [J]. *Mol Psychiatry*, 2012, 17(6): 624-633.