一种基于禁忌搜索算法的流程挖掘方法

白雪骢 朱 焱
(西南交通大学信息科学与技术学院 成都 611756)

摘 要 为了满足高效率的自动化生产需要,支持流程控制的工作流管理系统的应用越来越广泛。流程挖掘可以使用事件日志等历史数据生成抽象流程模型,为工作流系统的部署提供有利条件。首先总结归纳了一种较通用的基于启发式优化算法的流程挖掘框架;然后依照该流程挖掘框架将禁忌搜索算法用于流程挖掘领域,针对禁忌搜索中程序初始化、邻域构建方法和禁忌表构造等几个关键问题进行了详细阐述和论证;最后将算法实现为ProM的插件并进行了对比实验。实验验证了该流程挖掘框架的正确性,表明了禁忌搜索流程挖掘方法对不同流程结构具有良好支持,对数据噪声具有较强的鲁棒性和更少的时间消耗。

关键词 流程挖掘,禁忌搜索,Petri网,工作流网
中图法分类号 TP311.1 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.4.044

Process Mining Approach Based on Tabu Search Algorithm

BAI Xue-cong ZHU Yan
(School of Information Science and Technology, Southwest Jiaotong University, Chengdu 611756, China)

Abstract Workflow management systems which support process control are widely used in order to meet the needs of the high efficient automatic production. Process mining can use historical data, such as the event logs, to generate abstract process model, and provide favorable conditions for the deployment of workflow system. This paper presented a general process mining framework based on heuristic optimization algorithm, and then applied the tabu search algorithm to process mining task. Some key problems, such as program initialization, creation of tabu list and neighborhood, were discussed in detail. Finally, the algorithm was implemented as a plug-in of ProM. The experiment verifies the correctness of the process mining framework, and the tabu search process mining approach can deal with different flow structures and has robustness to noise and less time consuming.

Keywords Process mining, Tabu search, Petri net, Workflow net

大量企业开始使用信息系统支持和管理其业务流程,例如企业资源规划系统(ERP)以及工作流管理系统(WFM)等。但在实际生产应用中,这些智能化系统的部署工作困难重重,主要有3方面的原因:1)流程的设计及建模需要系统应用领域的专业人员,并且业务流程的初始建立依赖于建模者对业务流程的主观理解,难免存在偏差;2)不同应用领域的业务流程千差万别,不能通用;3)业务流程在设计与使用中经常发生修改与调整。

流程挖掘技术是一种从流程日志中发现流程模型的有效技术。从技术方面来说,流程挖掘综合了基于模型的过程分析技术(例如流程仿真)和以数据为基础的数据挖掘技术(例如机器学习)。流程挖掘能揭示蕴藏在事件日志数据中的模型规律,关注的重点是始端至末端的流程模型。从应用方面来说,流程挖掘是跨越智能商业任务和IT技术之间鸿沟的有效方法[1]。稍早期的流程挖掘方法有Cook等人基于马尔科夫概率转移的挖掘方法,Greco等人基于分层树型结构的挖掘方法,Van der Aalst等人发明的算法[2]。这些算法存在对流程结构的支持不完整或者对事件日志中的噪音敏感的缺陷,而针对流程挖掘的遗传算法[3]的时间开销由于其算法本身复杂的原因而变得很大。

本文首先总结并论证了一种基于因果关系矩阵方法的通用启发式流程挖掘框架,在此基础上提出并实现了针对流程挖掘的禁忌搜索算法,讨论了禁忌搜索用于流程挖掘时的参数选择与优化问题。与基于概率或基于统计的流程挖掘算法相比较,该算法支持完整的流程结构,具有较强的抗噪声性和较低的算法时间消耗。

1 相关工作

Cook等人早在1995年就提出了流程挖掘的概念,当时称其为流程发现(process discovery)。Cook等人提出了马尔科夫方法等若干流程挖掘方法,开辟了流程挖掘这一研究领域,而后又提出了一系列流程挖掘方面的度量指标,如熵、事件类型数和流程任务因果关系等概念。

Van der Aalst等人发明的α算法[2]介绍了系统的构建工

到稿日期:2015-01-01 返修日期:2015-04-21
白雪骢(1990—),硕士生,主要研究方向为工作流建模与流程挖掘,E-mail:bxc110900@126.com;朱 焱(1965—),博士,教授,主要研究方向为数据挖掘、Web资源质量管理、数据仓库系统,E-mail:yzhu@swjtu.edu.cn。

作流模型的一系列步骤，该算法可以将工作流模型的结构以工作流网的结构表示出来，而工作流网是 Petri 网的一种特殊形式，是 Petri 网概念的子集。很多流程挖掘数据集上的实验已经验证 α 算法是正确的，但是该算法对数据集中的噪声数据较为敏感，抗噪声能力弱；该算法的另一个缺陷是对流程结构的支持不完善。正因为如此，有许多学者在改进 α 算法方面做了许多工作。闻立杰提出的 α++算法可以有效处理流程结构中的非自由选择结构；而闻立杰提出的 $\alpha\#$ 算法可以处理任务流程中的不可见任务[4]。

流程挖掘遗传算法是由 W. M. P. van der Aalst 和 A. K. Alves de Medeiros 等人设计的一种基于 Petri 网理论的启发式流程挖掘方法[5]，其将 Petri 网抽象为因果关系矩阵，这种流程模型的表示方法可以适应真实流程中的各种结构，在算法计算过程中随机初始化流程模型种群并使用遗传算法中的交叉与变异两种操作来保持工作流模型种群的多样性，并且使用一定的精英选择策略选择保留种群中适应值高的个体。但是由于遗传算法本身复杂度较高，该算法在每一步的计算中种群不断迭代计算，这不仅要依据一定的交叉概率和变异概率对流程模型进行变异操作，还要在变化完成后计算种群中每一个个体的适应值。正是这些原因导致算法在种群较大的情况下的执行异常耗时。

意大利的 Greco 等人使用层次树作为过程模型的表示方式，在挖掘过程中力求能够从各个层次抽象表示流程模型[6]。层次树的根节点表现的是在事件日志数据中最常见的序列模式，代表了最一般化的流程模型，具有流程模型的公共特征；叶节点表现的是不同流程实例之间的差异性特点；根节点与叶节点之间的节点表示的是不同层次的抽象流程模型的一般序列模式。

清华大学范玉顺教授是国内研究工作流建模领域开始较早的研究者，他提出了一种基于协调理论和反馈机制的工作流建模方法[7]。清华大学的王建民和闻立杰利用着色 Petri 网对工作流模式建模[8]，这两位学者译著[9] W. M. P. van der Aalst 所著的第一本过程挖掘专著，面向的读者广泛，内容涉及过程挖掘的各个方面。

除算法研究之外，文献[10]使用流程挖掘技术优化服务流程；文献[11]根据不同流程模型选择最优的挖掘算法的智能方法；文献[12]设计实现了一种两步方法来平衡流程挖掘中欠拟合和过拟合的平衡问题。

与上述方法不同的是，本文设计的针对流程挖掘的禁忌搜索算法在具有完整流程表达能力的基础上，以因果关系矩阵为基础，采用禁忌搜索寻优算法，在流程挖掘能力和计算时间消耗之间取得良好的平衡。

2 概念及定义

定义 1(Prtri 网) Petri 网是一个三元组，设 $PN=(P,T,F)$，其中：

(1)P 是一个集合，代表有限个库所；

(2)T 是一个集合，代表有限个变迁并且 $P\cap T=\emptyset$；

(3)F 是 PN 中有向弧的集合，表示为 $(P\times T)\cup(T\times P)$。

Petri 网是现有的流程定义语言的基础，具有严格的数学证明，又可以图形化表示，具有多角度系统描述方式和成熟的系统分析技术，为计算机技术方面的流程模型建立和数学方面的流程抽象提供了便利。

定义 2(工作流网) 工作流网是一个三元组，设 $WFN=(P,T,F)$ 是一个 Petri 网，则 WFN 是一个工作流网当且仅当：

(1)WFN 存在一个库所 i 且满足 $\cdot i=\emptyset$，即 i 是整个有向网络的起始库所，没有前驱节点；

(2)WFN 存在一个库所 o 且满足 $o\cdot=\emptyset$，即 o 是整个有向网络的终止库所，没有后继节点；

(3)如果在 WFN 中加入一个新的变迁 t，使 t 连接库所 o 与 i，即 $\cdot t=\{0\}$，$t\cdot=\{i\}$，这时所得的 WFN 是强连接的。

由工作流网的定义可知，工作流网的起始库所的托肯(token)代表一个流程实例的开始，而终止库所的托肯代表了一个流程实例的结束。在工作流网中不能存在处于孤立状态的库所，所有活动的库所和变迁均位于由起始库所到终止库所的通路上。一个工作流网的简单实例如图 1 所示。

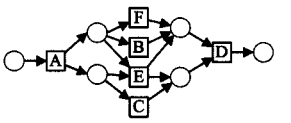

图 1 一个工作流网的简单实例

定义 3(因果关系矩阵) 因果关系矩阵可以用一个四元组来表示，设 $CM=(A,C,I,O)$，其中：

(1)A 是一个集合，代表有限个库所；

(2)$C\subseteq A\times A$ 表示活动之间存在的关系；

(3)$I:A\to\mathcal{P}(\mathcal{P}(A))$ 是输入条件函数，$\mathcal{P}(A)$ 表示集合 A 的幂集；

(4)$O:A\to\mathcal{P}(\mathcal{P}(A))$ 是输出条件函数。

而对于活动之间存在的关系 C，又有：

(1)$C=\{(a_1,a_2)\in A\times A\mid a_1\in\cup I(a_2)\}$，其中 $\cup I(a_2)$ 表示 $I(a_2)$ 元素的并集；

(2)$C=\{(a_1,a_2)\in A\times A\mid a_2\in\cup O(a_1)\}$，其中 $\cup O(a_1)$ 表示 $O(a_1)$ 元素的并集；

(3)$C=\{(a_o,a_i)\in A\times A\mid a_o\cdot=\emptyset\wedge\cdot a_i=\emptyset\}$ 是一个强连通图。

符合工作流网定义的 Petri 网可以被抽象为一个因果关系矩阵，并且因果关系矩阵可以被反向变换成为 Petri 网，进而转换为其它流程标记语言[3]。根据定义 3，因果矩阵中的因果关系呈对应关系，因果关系矩阵不可分割，除初始节点和终止节点外，每个节点都需要链接前驱节点和后继节点，不能出现孤立节点。因果关系矩阵有输入输出表和矩阵两种表示形式。如图 1 中的简单工作流网，可以根据定义将其抽象成为如表 1 的矩阵表示形式。在使用启发式算法对流程进行挖掘的过程中，为了易于构造算法使用的数据结构以及实现计算机方便，一般使用矩阵表示法来表示一个工作流流程模型。

表 1 工作流网的因果关系矩阵表示法

任务输出集	A	B	C	D	E	F	任务输入集
A	0	1	1	0	1	1	{}
B	0	0	0	1	0	0	{{A}}
C	0	0	0	1	0	0	{{A}}
D	0	0	0	0	0	0	{{F,B,E},{E,C}}
E	0	0	0	1	0	0	{{A}}
F	0	0	0	1	0	0	{{A}}

3 启发式流程挖掘框架

对于流程挖掘任务，本文提出的一般化流程可以完成对事件日志的流程挖掘，如图 2 所示。

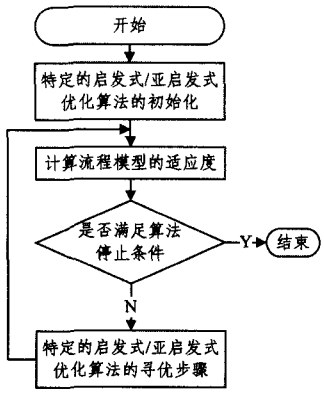

图 2 基于启发式寻优算法的流程挖掘的一般步骤

下面就该流程挖掘的框架进行详细说明。

• 启发式/亚启发式优化算法的初始化。该步骤最主要的目的是根据算法本身的原理和流程挖掘任务特点生成可供寻优操作的个体。不同算法的初始化方法不相同，如遗传算法需要在该步骤生成一个初始种群，种群中包括若干个体，每一个个体都是以一个因果关系矩阵的形式表示，个体数目根据实验的需要确定，而这种初始种群的生成一般采用随机生成的方式；又如模拟退火算法和禁忌搜索算法在该步骤仅需要生成一个初始个体，其也是由因果关系矩阵表示的，该个体是后继优化步骤的基础，所以初始个体的选择关系到后继操作的便利性和流程模型的具体形式。除此之外，一些算法如模拟退火算法需要在这一步设置初温，禁忌搜索算法需要分配禁忌表等。

• 计算流程模型的适应度。该步骤是后继步骤判断算法停止的关键因素，流程模型的评判标准主要体现在要求流程模型的准确性、完整性、简单性和通用性 4 个方面，4 者存在相互制约的关系，在 4 者中寻找到合理的平衡成为流程挖掘要解决的难点之一。不同的流程模型适应度计算方法在评判流程模型的优劣时有不同的侧重点，本文使用的适应度函数综合了对流程准确性和完整性的综合考虑。

• 算法停止的条件。算法停止的条件一般为实验设计者自行定义，针对流程挖掘任务，一般有几种常用的算法停止条件：适应度函数的值达到某一特定预设数值、算法的循环次数达到某一特定预设数值、适应度函数的值经历若干次循环（小于算法循环次数）未发生变化等。在实验实现中往往多个算法停止条件同时使用。

• 启发式/亚启发式优化算法的寻优操作。该步骤是整个流程挖掘工作的核心部分，主要目的是在算法初始化个体的基础上通过不同算法的不同寻优方式使个体或群体中某一个体的适应度提高。不同算法的寻优方式不同是算法间的主要区别，也是该步骤的难点，由于要将这些算法应用于流程挖掘领域，因此需要对算法进行修改，将流程挖掘任务中的概念根据算法的不同进行抽象。例如遗传算法应用于流程挖掘时，遗传算法中的个体被重新定义为一个因果关系矩阵，种群被重新定义为一个有若干因果关系矩阵的集合，在寻优操作这一步，参照遗传算法的原理，进行遗传繁殖时的基因交叉、基因变异和精英选择策略都将被重新定义为针对因果关系矩阵的操作，而这些操作的合理性和有效性成为寻优效果优劣的主要影响因素。另外，算法中的参数需要针对流程挖掘任务的特点通过大量实验确定。

4 基于禁忌搜索的流程挖掘算法

禁忌搜索算法是针对局部寻优方法进行改进的一种全局逐步寻优算法。禁忌搜索算法最大的特点是模拟了一种记忆方式，即一定次数内避免对已搜索过的解空间重复搜索。因为遍历整个解空间是 NP 完全问题[7]，禁忌搜索算法从解空间中的一个解出发，根据人为设置的规则选择一些移动方向，对每个方向上移动后的新解计算适应度函数，选择适应度高的移动方向。为了避免陷入局部最优解，禁忌搜索算法采取了一种叫做禁忌表的记忆方式，即对已经进行过搜索的解空间进行记录并以此为依据指导算法下一次移动时方向的选择。禁忌表中保存的是程序最近若干次的移动，禁忌表中保存移动结果的多少取决于禁忌表的长度。凡是处于禁忌表中被禁的结果，是不允许在下一次移动中采用的，正是由于这种策略使得算法不能连续几次迭代都使用同一结果，从而迫使算法向未探索的解空间移动，防止迭代循环陷入局部最优。下面就针对流程挖掘的禁忌搜索算法的几个关键点进行阐述。

4.1 基于禁忌搜索的流程挖掘算法初始化

禁忌搜索算法的初始化是确定待解问题的解空间中的一个初始解，本文对处理任务进行抽象，形成因果矩阵。因为初始解的质量对禁忌搜索算法的收敛速度和在解空间中的搜索范围都有一定影响，一般的做法是先使用其他算法生成一个较高质量的解，以此解作为禁忌搜索算法的初始解以提高禁忌搜索算法的搜索效率。在流程挖掘领域，可以采用 α 算法等基于 Petri 网的简单算法作为提供初始解的算法。

4.2 邻域结构和邻域解

良好的邻域结构是保证禁忌搜索算法在解空间中有足够搜索能力的基础。邻域移动是指由当前解出发，按照规定的移动策略产生若干新解，这些新解被称为邻域解，新解的数目称为邻域规模。本文构造邻域结构的移动策略分为 3 种：交换事件因果关系，即某一活动与矩阵位置上下左右 4 个方向的元素之一交换关系；新增事件因果关系，即因果关系矩阵中随机位置上一个无因果关系的活动（表示为 0）变为有因果关系（表示为 1）；减少事件因果关系，即因果关系矩阵中随机位置上一个有因果关系的活动（表示为 1）变为无因果关系（表示为 0）。以一个简单因果关系矩阵为例，3 种邻域构建方法的具体操作如图 3 所示。通过 3 种移动变化策略，可以保证算法有足够的搜索能力在解空间中以可接受的时间代价寻得较优解。

$$\begin{bmatrix}1&0&0\\0&1&0\\1&0&0\end{bmatrix}\quad\begin{bmatrix}1&0&0\\0&0&0\\1&1&0\end{bmatrix}\quad\begin{bmatrix}1&0&1\\0&1&0\\1&0&0\end{bmatrix}\quad\begin{bmatrix}1&0&0\\0&0&0\\1&0&0\end{bmatrix}$$

原始矩阵　交换操作　新增操作　减少操作

图 3 邻域构建方法

4.3 禁忌表

禁忌表的结构是一个先入先出的队列，禁忌表的长度是禁忌表的最关键参数。禁忌表中储存的是被禁忌的对象，每

一个禁忌对象就是一个因果关系矩阵，而禁忌表的长度是在不考虑特赦准则的情况下进入禁忌表的因果关系矩阵被禁止选择的最大迭代次数。考虑到运算时间和存储资源的开销，禁忌表应该尽可能小，但是禁忌表过小会导致陷入局部最优而无法完成全局搜索，所以禁忌表的长度设置是定义禁忌表的关键问题。禁忌表大小随着处理问题的规模不同而变化，本文中算法的禁忌表大小为待处理问题规模的二倍。

4.4 适应度函数

流程挖掘的适应度函数用来判断流程模型是否符合需求。本文采用的适应度函数分为两个部分：对流程模型的完整性考察和精确性考察。流程模型的完整性指流程模型能够尽可能多地表达日志数据中的实例，流程模型的精确性是指流程模型能够尽量少地表达日志数据当中不存在的流程实例。而流程模型的完整性越高，流程模型所能表达的日志数据之外的流程实例也就越多，流程模型的精确性就会下降，流程模型会出现欠拟合情况；相反地，如果流程实例的精确性越高，则流程模型越有可能不能完整表达日志中的实例流程，二者是相互制约关系[8]。本文中使用的适应度函数定义如下。

定义 4（流程模型的完整性函数 $F_{complete}$） L 表示一个非空的事件日志，a 表示 L 中的一个实例，M 表示一个因果关系矩阵，则：

$$F_{complete}(L,M)=\frac{\sum_{i=1}^{unmTraces(L)}(NumActivitiesParsed(a,M))-PenaltyFactor}{\sum_{i=1}^{unmTraces(L)}(NumActivities(a))}$$

其中：

$$PenaltyFactor=\frac{\sum_{i=1}^{unmTraces(L)}(NumMissingTokens(a,M))}{unmTrace(L)-numTraceMissingTokens(L,M)+1}+\frac{\sum_{i=1}^{unmTraces(L)}(NumExtraTokens(a,M))}{unmTrace(L)-numTraceExtraTokens(L,M)+1}$$

$unmTraces(L)$ 表示事件日志 L 中实例的个数；$NumActivities(a)$ 表示实例 a 中的活动事件数；$NumActivitiesParsed(a,M)$ 表示因果关系矩阵 M 能表达实例 a 中的活动事件的个数；$NumMissingTokens(a,M)$ 表示因果关系矩阵 M 在表达实例 a 时丢失的托肯数；$NumExtraTokens(a,M)$ 表示因果矩阵 M 在表达实例 a 时额外的托肯数。

定义 5（流程模型的精确性函数 $F_{precise}$） L 表示一个非空的事件日志，a 表示 L 中的一个实例，M 表示一个因果关系矩阵，$M[]$ 表示包含 M 的因果关系矩阵的集合，则：

$$F_{precise}(L,M,M[])=\frac{\sum_{i=1}^{unmTraces(L)}(NumEanbleActivities(a,M))}{\max(\sum_{i=1}^{unmTraces(L)}(NumEanbleActivities(a,M[])))}$$

$NumEanbleActivities(a,M)$ 表示因果关系矩阵 M 在表达实例 a 时激活的活动事件数。

定义 6（流程模型的适应度函数 F） L 表示一个非空的事件日志，a 表示 L 中的一个实例，M 表示一个因果关系矩阵，$M[]$ 表示包含 M 的因果关系矩阵的集合，$F_{complete}$ 和 $F_{precise}$ 是在定义 4 和定义 5 中定义的流程模型完整性和精确性函数，γ 是一个实数，$\gamma\in(0,1]$，则：

$$F(L,M,M[])=F_{complete}(L,M)-\gamma\times F_{precise}(L,M,M[])$$

由适应度函数值的公式可知，若一个因果关系矩阵对日志事件数据中的流程的表达能力相同，即 $F_{complete}(L,M)$ 相同，则表达事件日志数据之外的额外行为能力弱，即 $F_{precise}(L,M,M[])$ 更小，因果关系矩阵获得更高的适应度值。

4.5 特赦准则

特赦准则也被称为藐视准则，在流程挖掘过程中，当出现候选解全部处于禁忌表中或者当前候选解中存在优于当前最优解的解时它就发挥作用。特赦准则会选择当前候选解中适应度函数值最高的因果关系矩阵，用该因果关系矩阵代替当前的最优解，若该因果关系矩阵存在于禁忌表中，则将其从禁忌表中解禁后重新添加为禁忌表的第一个元素。

4.6 代码描述

```
1. read event log data
2. M←M0, Mbest←M, TabuList←null
3. While(not StopCondition())
4.   CandidateList←null
5.   For(every Mcandidate in Mneighborhood)
6.     If(Tabulist not contains Mcandidate)
7.       CandidateList←CandidateList+Mcandidate
8.     End if
9.   End for
10. Mcandidate←LocateBestCandidate(CandidateList)
11.   If(fitness(Mcandidate)>fitness(Mbest))
12.     TabuList←TabuList+Mcandidate
13.     mbest←Mcandidate
14.     while(size(TabuList)>maxTabuListSize)
15.       Release(TabuList)
16.     End while
17.   End if
18. End while
19. Return(Mbest)
```

M 代表一个因果关系矩阵，每一轮迭代生成新的候选解 $M_{neighborhood}$ 的方法是以一定概率采用邻域结构中提出的 3 种操作。在 $M_{neighborhood}$ 这个候选解集合中判断候选解是否存在于禁忌表中，在候选解中存在适应度值高于当前最优解的情况下，使用该解替换当前最优解，并将它加入禁忌表，进行下一次迭代，直至达到算法停止条件，返回的当前最优解即为结果。在算法运行过程中随着迭代的进行，禁忌表以先入先出的原则不断更新。

5 实验结果及分析

实验在 2.5GHz 双核 CPU、8GB 内存、Windows8.1 操作系统的微型计算机上实现。数据集来自 processmining.org 网站提供的事件日志数据。实验搭建在流程挖掘平台 ProM 上，将基于禁忌搜索的流程挖掘算法实现为该平台的一个插件。

5.1 初始解的选择对比实验

禁忌搜索算法对初始解较为敏感，所以第一个实验用于验证在流程挖掘方面算法的初始解对算法性能的影响。一组实验使用随机方法生成一个因果关系矩阵作为程序的初始解，另一组使用 α 算法先对事件日志进行一次处理，将 α 算法的结果作为程序的初始解。禁忌表长度采取二倍问题规模即待挖掘事件日志中活动种类数的两倍；在邻域构建时，以 0.6 的概率交换事件因果关系，以 0.2 的概率新增事件因果关系

和减少事件因果关系，邻域的规模设定为 50。由于随机初始解的制造和禁忌搜索构建邻域都存在一定的随机性，因此实验分别都进行了 10 次，每次进行 25 次迭代。实验的结果如图 4 所示。

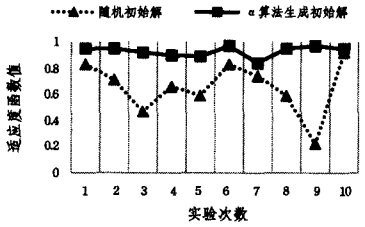

图 4　初始解选择对比实验

使用随机初始解方法的禁忌搜索算法的结果很不稳定，原因是由于初始解的随机性，有可能使算法在禁忌表长度等参数的限制下寻优能力不足以突破局部最优，或者在一定的迭代次数里还没有接近整个解空间中的较优解。而使用算法生成初始解的方法可以使算法的解适应度保持在较高的水平。通过实验可以说明，禁忌搜索在处理流程挖掘任务时依然是对初始解敏感的，单独的禁忌搜索算法表现不稳定。为了提高禁忌搜索算法对于流程挖掘任务的效率，有必要给定一个适应度较高的初始解。

5.2　禁忌搜索参数选择实验

禁忌搜索算法最关键的参数是禁忌表的长度，禁忌表长度过短会导致禁忌搜索算法被局部最优解吸引而无法跳出；禁忌表长度过长会增加系统空间与运算量消耗，收敛速度变慢。本文设计以规模为 8 个事件的过程模型为例，分别设置禁忌表长度为问题规模的 1 倍、2 倍、4 倍(即 8、16 和 32)完成对比实验，以说明禁忌表长度对算法挖掘效率的影响。实验结果如图 5 所示。

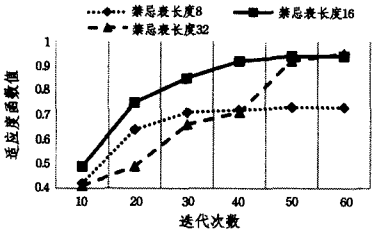

图 5　禁忌表长度对实验结果的影响

通过这组实验发现，禁忌表长度为 8 的实验组在进行 30 余次迭代后陷入一个局部最优解，并且经过多次(数十次)迭代均未跳出局部最优解的吸引，在这次实验中没能成功挖掘过程模型。而禁忌表长度为 16 和 32 的两组实验均在一定次数的迭代后得到了较为满意的结果，并且从图中数据可知，前 50 次迭代中禁忌表长度为 16 的实验组的适应度函数值提升的速度更快，算法收敛速度较禁忌表长度为 32 的实验组更快，并且考虑到禁忌表长度越大系统开销越大，本文选择禁忌表长度为 16 作为事件日志 2 的禁忌表长度参数。依据该实验，在选择禁忌搜索算法的过程挖掘方法中以禁忌表长度为事件日志中事件种类数目的二倍作为默认参数。

5.3　与遗传算法的对比实验

遗传算法在大规模组合优化问题上有其独特优势。在流程挖掘领域，遗传算法借助由 Petri 网抽象的因果关系矩阵这种流程模型表示方式克服了之前许多流程挖掘算法的缺点，但是由于自身算法的计算过程较复杂，涉及到种群中大量个体数目的计算时间消耗过长。实验中，遗传算法使用文献[3]中设计实验的基本参数，种群包含 100 个个体，最多进行 1000 次迭代，优秀个体选择比例为 0.02，根据定义 6 计算适应度，惩罚系数为 0.025，个体交叉的概率为 0.8，个体发生变异的概率为 0.2；禁忌搜索算法的实验参数保持与上述实验不变。实验在 3 个不同事件日志数据上完成，当算法的输出结果适应度函数值到达 0.95 即停止计时，每个时间数据均为重复 10 次实验所取的平均值。

实验结果如图 6 所示。由实验结果可知，本文提出的基于禁忌搜索的流程挖掘方法在时间消耗上较遗传算法有优势，随着 3 个不同事件日志数据的复杂度和问题规模增大，算法时间消耗的规模变大。遗传算法在每次迭代时的种群交叉变异操作较本文方法花费更多时间，并且由于算法种群规模更大，计算适应度函数值的过程也更加耗时。

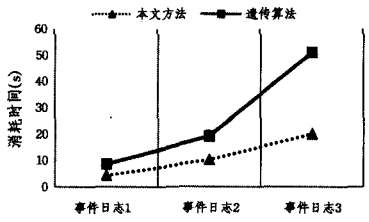

图 6　本文方法与遗传算法的比较实验

结束语　本文就使用启发式优化算法在流程挖掘方面的应用做了讨论，并提出使用禁忌搜索算法完成流程挖掘任务。本文的流程挖掘方法不仅可以应对各种流程形式，而且跟遗传算法一样具有较强的噪声鲁棒性，并且具有较遗传算法更少的时间消耗。

因为本文方法基于由禁忌表辅助记忆的少量因果关系矩阵，所以对解空间的搜索范围略小于遗传算法，但是在流程模型规模不是太大的情况下，本文方法在更少的时间内可以达到遗传算法的计算效果；在问题规模较大的情况下，禁忌搜索算法的局限性开始体现，算法时间和存储代价的制约以及邻域的构建方式和结构限制导致禁忌搜索算法在该类问题上的全局搜索能力不足。使用搜索能力更强的全局搜索算法和效率更高的局部搜索算法构成的组合算法来弥补这种不足是今后工作的主要挑战。

参 考 文 献

[1] Van der Aalst W M P. Using Process Mining to Bridge the Gap between BI and BPM [J]. IEEE Computer, 2011, 44(12): 77-80

[2] Van der Aalst W M P, Weijters T, Maruster L. Workflow mining: Discovering process models from event logs [J]. IEEE Transactions on Knowledge and Data Engineering, 2004, 16(9): 1128-1142

[3] de Medeiros A K A, Weijters A J M M, Van der Aalst W M P. Genetic process mining: an experimental evaluation [J]. Data Mining and Knowledge Discovery, 2007, 14(2): 245-304

[4] Wen Li-jie. Studies on Algorithms for Process Mining based on WF-net[D]. Beijing: Tsinghua University, 2007(in Chinese)
闻立杰. 基于工作流网的过程挖掘算法研究[D]. 北京：清华大学，2007

(下转第 240 页)

选取目标用户的推荐对象集出发，结合推荐用户的信任度和目标项目的受欢迎度完成对目标用户的推荐，并给出了 SRN 算法和 CF-ODDA 算法的具体实现过程。上述算法有效地改善了邻居选取片面性的问题，同时也提高了算法的推荐精度。

在协同过滤推荐算法中，冷启动也是其中一个主要问题。当某个新项目没有被任何用户评价过或新用户最初向系统提供的自己的信息非常有限时，推荐系统就无法发挥其作用。此时，我们可以考虑运用跨平台机制，整合该用户或项目在各个平台上的信息，根据其进行推荐。因此，怎样缓解协同过滤中的冷启动问题将是我们下一步的研究工作。

参 考 文 献

[1] Jia Dong-yan, Zhang Fu-zhi. A collaborative filtering recommendation algorithm based on double neighbour choosing strategy [J]. Journal of Computer Research and Development, 2013, 50 (5): 1076-1084(in Chinese)
贾冬艳，张付志. 基于双重邻居选取策略的协同过滤推荐算法[J]. 计算机研究与发展，2013，50(5)：1076-1084

[2] Yamashita A, Kawanura H, Suzuki K. Adaptive fusion method for user-based and item-based collaborative filtering [J]. Advances in Complex Systems, 2011, 14(2): 133-149

[3] Wen Jun-hao, Zhou Wei. An improved item-based collaborative filtering algorithm based on clustering method [J]. Journal of Computational Information Systems, 2012, 8(2): 571-578

[4] Zhang Liang, Deepak A, Chen Bee-chung. Generalizing matrix factorization through flexible regression priors[C]// Proceedings of the 5th ACM Conference on Recommender Systems. Chicago, USA, 2011: 13-20

[5] Gao Hui-ji, Tang Ji-liang, Hu Xia, et al. Exploring temporal effects for location recommendation on location-based social networks[C]// Proceedings of the 7th ACM Conference on Recommender Systems. New York, USA, 2013: 93-100

[6] Zhou Ke, Yang Shuang-hong, Zha Hong-yuan. Functional matrix factorizations for cold-start recommendation [C]// Proceedings of the 34th Int ACM SIGIR Conference on Research and Development in Information Retrieval. Beijing, China, 2011: 315-324

[7] Ma Hao, Liu Chao, Irwin K, et al. Probabilistic factor models for Web site recommendation [C] // Proceedings of the 34th Int ACM SIGIR Conf on Research and Development in Information Retrieval. Beijing, China, 2011: 265-274

[8] Wang Hong-ning, He Xiao-dong, Chang Ming-wei, et al. Personalized ranking model adaptation for web search[C]// Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval. New York, USA, 2013: 323-332

[9] Yang Xi-wang, Harald S, Liu Yong. Circle-based recommendation in online social networks [C] // Proceedings of the 18th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. New York, USA, 2012: 1267-1275

[10] Joseph N, Scott S, Khoi-Nguyen T, et al. New objective functions for social collaborative filtering[C]// Proceedings of the 21st International Conference on World Wide Web. Lyon, France: WWW, 2012: 859-868

[11] Guo Lei, Ma Jun, Chen Zhu-min, et al. Incorporating Item Relations for Social Recommendation[J]. Chinese Journal of Computers, 2014, 37(1): 219-288(in Chinese)
郭磊，马军，陈竹敏，等. 一种结合推荐对象间关联关系的社会化推荐算法[J]. 计算机学报，2014，37(1)：219-288

[12] Zhang Bin, Zhang Yin, Gao Ke-ning, et al. Combining Relation and Content Analysis for Social Tagging Recommendation[J]. Journal of Software, 2012, 23(3): 476-488(in Chinese)
张斌，张引，高可宁，等. 融合关系与内容分析的社会标签推荐[J]. 软件学报，2012，23(3)：476-488

[13] Jiang Meng, Cui Peng, Liu Rui, et al. Social contextual recommendation[C]// Proceedings of the 21st ACM Conference on Information and Knowledge Management. Maui, USA, 2012: 45-54

[14] Yu Xiao, Ren Xiang, Sun Yi-zhou, et al. Personalized entity recommendation: A heterogeneous information network approach [C]// Proceedings of the 7th ACM International Conference on Web Search and Data Mining. New York, USA, 2014: 283-292

[15] Wang Yu, Gao Lin. Social circle-based algorithm for friend recommendation in online social networks[J]. Chinese Journal of Computers, 2014, 37(4): 801-808(in Chinese)
王珏，高琳. 基于社交圈的在线社交网络朋友推荐算法[J]. 计算机学报，2014，37(4)：801-808

[16] Huang Chuang-guang, Yin Jian, Wang Jing, et al. Uncertain neighbours' collaborative filtering recommendation algorithm[J]. Chinese Journal of Computers, 2010, 33(8): 1369-1377(in Chinese)
黄创光，印鉴，汪静，等. 不确定近邻的协同过滤推荐算法[J]. 计算机学报，2010，33(8)：1369-1377

(上接第 218 页)

[5] Van der Aalst W M P, de Medeiros A K A, Weijters A. Genetic process mining [M] // Applications and Theory of Petri Nets 2005. Springer Berlin Heidelberg, 2005: 48-69

[6] Greco G, Guzzo A, Pontieri L. Mining hierarchies of models: From abstract views to concrete specifications [M] // Business Process Management. Springer Berlin Heidelberg, 2005: 32-47

[7] Fan Yu-shun. The basis of workflow management technology [M]. Beijing: Tsinghua University Press, 2001(in Chinese)
范玉顺. 工作流管理技术基础[M]. 北京：清华大学出版社，2001

[8] Wen Li-jie, Wang Jian-min, Sun Jia-guang. Modeling workflow patterns using coloured petri nets[J]. Computer Science, 2006, 33(6): 135-139(in Chinese)
闻立杰，王建民，孙家广. 用着色 Petri 网建模工作流模式[J]. 计算机科学，2006，33(6)：135-139

[9] Van der Aalst W M P. 过程挖掘：业务过程的发现、合规和改进[M]. 王建民，闻立杰，等译. 北京：清华大学出版社，2014

[10] Van der Aalst W M P. Service Mining: Using Process Mining to Discover, Check, and Improve Service Behavior[J]. IEEE Transactions on Services Computing, 2013, 6: 525-535

[11] Wang J, Wong R K, Ding J, et al. Efficient Selection of Process Mining Algorithms[J]. IEEE Transactions on Services Computing, 2013, 6: 484-496

[12] Van der Aalst W M P, Rubin V, Verbeek H M W, et al. Process mining: a two-step approach to balance between underfitting and overfitting [J]. Software & Systems Modeling, 2010, 9(1): 87-111