软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2015,26(3):550−561 [doi: 10.13328/j.cnki.jos.004770]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

一种适用于多样性环境的业务流程挖掘方法*

杨丽琴[1,2,3], 康国胜[1,2], 郭立鹏[1,2], 田朝阳[1,2], 张 亮[1,2], 张笑楠[4], 高 翔[4]

[1](复旦大学 计算机科学技术学院,上海 201203)
[2](上海市数据科学重点实验室(复旦大学),上海 201203)
[3](上海中医药大学 图书信息中心,上海 201203)
[4](中国移动有限公司 信息系统管理部,北京 100084)
通讯作者: 张亮, E-mail: lzhang@fudan.edu.cn

摘 要: 从运行日志挖掘业务流程模型的流程挖掘方法研究方兴未艾,然而,复杂多变的运行环境使流程日志也不可避免地呈现出多样性.传统的流程挖掘算法各有其适用对象,因此,如何挑选适合多样性流程日志的流程挖掘算法成为了一项挑战.提出一种适用于多样性环境的业务流程挖掘方法 SoFi(survival of fittest integrator).该方法基于领域知识对日志进行分类,使用多种现有的挖掘算法对每一类子日志产生一组流程模型作为遗传算法的初始种群,借助遗传算法的优化能力,从中整合得到高质量的业务流程模型.针对模拟日志和某通信公司真实日志的实验结果表明:相对于任何单一的挖掘算法,SoFi 产生的流程模型具有更高的综合质量,即重现度、精确度、通用性和简单性.

关键词: 流程挖掘;流程整合;遗传算法;日志分类;ProM
中图法分类号: TP18

中文引用格式: 杨丽琴,康国胜,郭立鹏,田朝阳,张亮,张笑楠,高翔.一种适用于多样性环境的业务流程挖掘方法.软件学报, 2015,26(3):550−561. http://www.jos.org.cn/1000-9825/4770.htm

英文引用格式: Yang LQ, Kang GS, Guo LP, Tian ZY, Zhang L, Zhang XN, Gao X. Process mining approach for diverse application environments. Ruan Jian Xue Bao/Journal of Software, 2015,26(3):550−561 (in Chinese). http://www.jos.org.cn/1000-9825/4770.htm

Process Mining Approach for Diverse Application Environments

YANG Li-Qin[1,2,3], KANG Guo-Sheng[1,2], GUO Li-Peng[1,2], TIAN Zhao-Yang[1,2], ZHANG Liang[1,2], ZHANG Xiao-Nan[4], GAO Xiang[4]

[1](School of Computer Science, Fudan University, Shanghai 201203, China)
[2](Shanghai Key Laboratory of Data Science (Fudan University), Shanghai 201203, China)
[3](Library and Information Center, Shanghai University of Traditional Chinese Medicine, Shanghai 201203, China)
[4](Department of Management Information System, China Mobile Communications Corporation, Beijing 100084, China)

Abstract: Mining business process models from running logs is in its ascendant. Inevitably, the ever changing operational environment makes these log records diverse. Considering every mining algorithm has its pros and cons, this paper focuses on the challenge to apply a best mining algorithm against diverse logs. A novel approach, SoFi (survival of fittest integrator), is proposed to mine business process models effectively in such a diverse environment. SoFi tackles the diversity issue by utilizing domain knowledge to classify the cases in a log and applying various mining algorithms on these categories to obtain comprehensive process models as candidates for optimization. A genetic algorithm (GA) based optimizer takes these candidates as initial population for purpose of both genetic quality as well as genetic diversity. Under the principle of survival of fittest, the GA optimizer can aggregate best process fragments with context into the final

* 基金项目: 国家自然科学基金(60873115); 教育部-中国移动科研基金(MCM20123011); 上海市科技发展基金(13dz2260200, 13511504300); 上海中医药大学预算内项目(2013JW30)

收稿时间: 2014-07-01; 修改时间: 2014-09-30; 定稿时间: 2014-11-21

process model for the entire log. Experiments on synthetic data and real cases from a telecommunication firm demonstrate the effectiveness of SoFi and comprehensive quality of mined process models in terms of replay fitness, accuracy, generalization, and simplicity.

Key words: process mining; process consolidation; genetic algorithm; log classification; ProM

目前,大多数业务流程的执行通过信息系统来实现.与人工设计流程不同,流程挖掘根据信息系统中的运行日志来提取流程模型,产生的流程模型更具有客观性[1,2].因此,流程挖掘对实现业务流程的优化和智能管理有着十分重要的意义.

流程挖掘方面已经出现了许多挖掘算法和工具,它们都各有其特色和适用对象.例如:Aalst 等人[3]提出的α算法适合于不带短循环和隐式库所的结构化流程模型,它不能处理非局部选择结构;Wen 等人[4]提出的α^{++}算法能够处理短循环和非局部选择结构,但是不能处理日志噪声;Weijters 等人[5,6]提出的启发式挖掘算法能够处理日志噪声,但是不能处理非局部的选择结构和重复活动;Van Dongen 等人[7,8]提出的两阶段挖掘算法能够挖掘顺序和选择结构,但不能挖掘循环结构;Medeiros 等人[2,9,10]提出的基于遗传算法(genetic algorithm,简称 GA)的流程挖掘方法能够处理多种常用结构和日志噪声,但因其初始种群均是随机产生,因此可能得不到优质的流程模型.这些传统挖掘算法在某种特定应用场景下挖掘效果较好,但在复杂多变的应用环境中无法处理变化多样的日志.以中国移动公司的业务流程为例,其 31 家分公司独立地运行和维护着各自的业务流程.为了方便管理和控制,需要将这些分公司的业务流程整合成统一的业务流程.各分公司由于规章制度和组织结构不同等原因导致业务流程各异,使得日志具有多样性特征.因此,如何挑选适合多样性日志的流程挖掘算法成为了一项挑战.一种方法是采用聚类方法[11−13]将运行日志中的执行实例进行分类,然后对每一类执行实例分别采用已有的算法挖掘各自的流程模型.然而,采用这种方式获得的流程都是局部的业务流程,如何将这些局部的业务流程整合成完整的业务流程模型,是有待解决的问题.

针对以上需求,本文提出一种兼顾日志多样性和流程模型优化的业务流程挖掘方法.首先,通过分析日志中的数据信息,利用领域知识将日志中的执行实例进行分类;然后,对分类后的各子日志使用各种已有的挖掘算法产生系列流程模型;最后,将这些流程模型作为遗传算法的初始种群,借助遗传算法的优化能力从中整合出高质量的完整业务流程模型.

本文第 1 节提出一种基于适者生存思想的业务流程挖掘方法 SoFi(survival of fittest integrator).第 2 节介绍基于领域知识的日志分类方法.第 3 节介绍利用多种挖掘算法为遗传算法挖掘准备优质初始种群.第 4 节介绍基于遗传算法的流程模型整合方法.第 5 节利用模拟日志和某通信公司真实日志进行实验,分析验证本文方法的可行性和有效性.第 6 节介绍相关工作.第 7 节对本文进行总结.

1 SoFi:业务流程模型挖掘方法

多样性环境下的业务流程挖掘面临以下几个重要问题:(1) 环境的复杂性造成了运行日志的多样性,给挖掘工作带来了困难;(2) 流程挖掘算法均有各自的适用范围和凸显特征,缺乏普适性的挖掘算法;(3) 难以平衡挖掘结果的多样性和针对性质量要求.

为此,本文提出了一种流程模型挖掘方法 SoFi 如图 1 所示.该方法的输入为流程运行日志,输出为针对该日志挖掘出来的流程模型.首先,借助领域知识对日志中的执行实例进行分类,满足相同条件的执行实例被归到同一类,整个日志最后将被划分成N个子日志,从而解决日志的多样性问题,简化挖掘算法的应用环境,让挖掘算法的特征和优势得到充分发挥;然后,针对算法适用性困难,直接对分类后的每一个子日志实施多种流程挖掘算法.假设对每一个子日志使用M种流程挖掘算法,则将总共产生$N\times M$个挖掘结果(流程模型).这种盲目匹配子日志和算法的策略在凸显算法特征的同时,必将带来挖掘结果数量繁多、部分结果质量低劣的负面效应,这可留给后期的遗传算法按照适者生存的原则逐步剔除.将多种算法作用于各子日志,然后施用遗传算法整合还带来额外的好处是:(1) 初始种群的遗传多样性足够丰富;(2) 相对于传统的随机初始种群,遗传算法直接获得了先前挖掘结果(流程模型)的许多优良基因;(3) 基于适应值函数的优化易于综合多方面的目标,如重现度、精确度、

通用性和简单性.

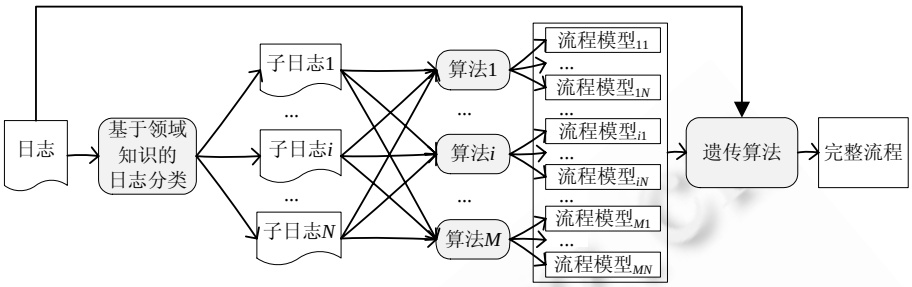


Fig.1 Process mining approach SoFi

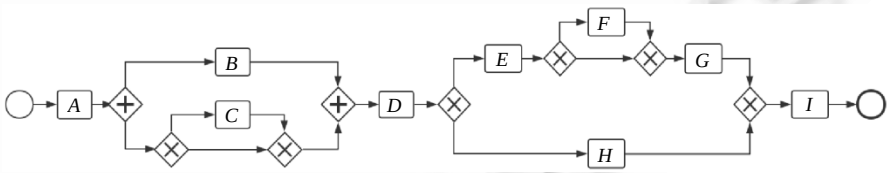
图 1 业务流程模型挖掘方法 SoFi

2 基于领域知识的日志分类

SoFi 方法的第 1 步就是日志分类.日志分类是根据分类条件将日志中的执行实例分组,从而形成多个子日志.通常,运行日志中记录了与活动有关的信息,如活动的执行者、执行时间或所处理的数据信息.这些数据信息对流程日志分类具有重要的指导意义[14,15].因此,本节将通过分析日志中的数据信息,利用领域知识对日志进行分类.下面通过一个具体流程来介绍该日志分类方法.

2.1 考虑数据的日志分类方法

某电子公司提供移动电话和导航仪的维修服务,其报修流程[11]的 BPMN 模型如图 2 所示.首先,公司收到顾客发出的报修请求和维修产品(*A*),然后,由技术人员对产品进行检测(*B*),如果报修的产品是移动电话,则检查顾客的保修单(*C*).凭检测结果和保修单,计算修理费用并告知顾客(*D*);如果报修的产品是导航仪,则无需检查保修单,直接计算修理费用.如果顾客决定修理,则为其修理产品(*E*)并收取费用(*G*).如果修理的产品是导航仪,则在修理之后还需进行实地测试(*F*).若顾客放弃修理产品,发出取消请求(*H*),产品寄回给顾客(*I*),流程结束.



A:接收报修请求和产品;*B*:检测产品;*C*:检查保修单;*D*:计算费用通知顾客;
E:修理产品;*F*:测试产品;*G*:收取费用;*H*:发出取消请求;*I*:寄回产品

Fig.2 Repair process using the BPMN notation

图 2 报修流程的 BPMN 模型

该流程所处理的数据对象(data object)包含 8 个必须的属性,各属性及其取值范围见表 1.其中,RepairType 为修理产品的类型,取值为 Mobile 或 Navigation;CustomerID 为顾客 ID;CustomerAddress 为顾客的地址;FailureType 为产品检测到的故障类型,共分为 8 种类型,取值为 *F*0~*F*7;WarrantID 为保修单号,Payment 为修理所需的费用,取值为大于 0 的数值;Canceled 表示是否发出取消请求,Paid 表示费用是否支付成功,取值为 Yes 或 No.

表 2 中列出了报修流程的部分执行日志,每一行记录表示一个事件(event),记录了与事件相关的各种信息,如事件 ID(EventID)、事件发生的时间(timestamps)、执行的活动名称(activity)、活动的执行者(resource)以及该活动处理的数据对象的属性.执行实例(case)是流程的一次执行过程,用实例 ID(CaseID)标识.每个事件属于某一个执行实例.每个执行实例只能操作属于本实例的数据对象中的属性.流程开始执行时创建数据对象并初始化其中的所有属性,例如表 2 中第 2 行 EventID 为 1001 的记录,该事件执行流程的第 1 个活动 *A*,该活动分别初

始化数据对象中的 8 个属性:顾客 ID(CID)为 CID12,顾客地址(CAdd)为 No.280 Hutu R.D.,送修产品类型(RT)为 Navigator;其他属性为默认值.后续活动如果对其中某个属性做了更新,则记录该属性更新后的值.例如,第 6 行 EventID 为 1005 的记录,该事件执行的活动是计算修理费用并告知顾客(*D*),该活动对 Pmt 属性做了更新操作,(Pmt,2000)表示修理费用为 2 000 元.

Table1 1 Attributes and allowed states in data object of repair process

表 1 报修流程数据对象中的属性及其取值范围

属性名称(缩写)	取值范围	说明
RepairType (RT)	Mobile 或 Navigation	产品类型
CustomerID (CID)	形如:CID1,CID2 等的字符串	顾客 ID 号
CustomerAddress (CAdd)	形如:No.560 Cailun R.D.等的字符串	顾客地址
FailureType (FT)	F_0~F_7	故障类型代码
WarrantID (WID)	型如:No.1024 等的字符串	保修单号
Payment (Pmt)	大等于 0 的整数	支付金额
Canceled (Can)	Yes 或 No	是否取消修理
Paid (Paid)	Yes 或 No	是否付款

Table 2 Event log fragment of the repair process

表 2 报修流程部分执行日志

Event ID	Case ID	Properties			
		Timestamps	Activity	Resource	Data object
…	…	…	…	…	…
1001	103	05-01-2014 08:10am	*A*	Clare	(CID,“CID12”) (CAdd,“No.280 Hutu R.D.”) (RT,“Navigator”)
1002	104	05-01-2014 11:00am	*A*	John	(CID,“CID16”) (CAdd,“No.560 Cailun R.D.”) (RT,“Mobile”)
1003	103	05-01-2014 14:20pm	*B*	Pete	(FT,“*F*1”)
1005	103	05-01-2014 15:01pm	*D*	Sue	(Pmt,2000)
1006	104	05-01-2014 16:21pm	*B*	Pete	(FT,“*F*2”)
1007	104	05-01-2014 17:30pm	*C*	John	(WID,“No1324”)
1008	103	06-01-2014 09:00am	*E*	Mike	(Can,“No”)
1009	104	06-01-2014 10:00am	*D*	Sue	(Pmt,4000)
1010	103	07-01-2014 12:04pm	*F*	Pete	
1011	103	07-01-2014 14:00pm	*G*	Jane	(Paid,“Yes”)
1012	103	07-01-2014 16:10pm	*I*	Sue	
1013	104	07-01-2014 16:20pm	*H*	Clare	(Can,“Yes”)
…	…	…	…	…	…

分析表 2 日志中的数据对象,根据领域知识将日志中的执行实例分成 3 类:数据对象中,属性 RT 为 Navigator 且属性 Can 为 No 的执行实例归为第 1 类;属性 RT 为 Mobile 且属性 Can 为 No 的执行实例归为第 2 类;属性 RT 为 Navigator 或 Mobile,且属性 Can 为 Yes 的执行实例归为第 3 类.分类后,第 1 个子日志中包含的是修理导航仪的执行实例,第 2 个子日志中包含的是修理移动电话的执行实例,第 3 个子日志中包含的是报修导航仪或移动电话后又取消修理的执行实例.

2.2 日志分类算法

本节介绍基于日志数据信息和领域知识的日志分类算法,见算法 1.算法的输入为一个运行日志 Log 和一组基于领域知识的分类条件 Conditions.算法的输出为分类后的一组子日志 SLogs.算法第 1 行~第 4 行创建并初始化子日志.因为 Conditions 中的每个分类条件对应一个子日志,因此,子日志的个数等于分类条件的个数.第 5 行~第 12 行扫描 Log 中的执行实例,并将它们归到对应的子日志中.第 6 行 *d*=*getDataObject*(*a*)是对于日志中的每一个执行实例 *a* 获取其对应的数据对象 *d*.第 8 行~第 10 行判断 *d* 是否满足某个分类条件,若满足,则将执行实例 *a* 加入到对应的子日志中.

算法 1. 考虑数据的日志分类算法.

Input: Log, Conditions //输入:日志,分类条件

Output: Slogs //输出:分类后的一组子日志

1: **for** each condition c_i in Conditions

```
2:      Create slog_i; Set slog_i=∅;          //初始化各子日志
3:      SLogs=SLogs∪slog_i;
4:   end for
5:   for each trace a in Process Log          //对每一个执行实例
6:      d=getDataObjects(a)                   //获取该执行实例的数据对象
7:      for each condition c_i in conditions
8:         if d satisfies c_i
9:            slog_i=slog_i∪a                 //若 d 满足第 i 个分类条件,则将 d 所
10:        end if                             //属的执行实例 a 加入到第 i 类子日志
11:     end for
12: end for
```

3 利用多种挖掘算法准备优质种群

基于领域知识的分类解决了日志的多样性问题,而为每一类子日志选择适合的挖掘算法依然是一个挑战,因为各种流程挖掘算法均有各自的适用范围,得到的流程模型也各有特色.

为了解决这个问题,SoFi 将多种流程挖掘算法施加于各类子日志,以增强算法遇到合适日志的可能性.挖掘结果通过后期的遗传算法进行整合与优化.正如第 1 节指出,多种挖掘结果既提高了遗传算法初始种群的质量,加速了遗传算法的收敛;而且初始种群具备遗传多样性,避免遗传操作的近亲结缘,从而提高了最终挖掘结果的质量.例如,选用α算法[3]、Heuristic 算法[5,6]和 Region-Based 挖掘算法[16]来为各子日志挖掘流程模型.α算法利用活动之间的二元关系来构造 Petri 网模型,模型中不带重复活动和不可见活动,因此结果模型相对简单. Heuristic 算法的优点是可以处理日志噪声,其关键是阈值的设定.由于 Heuristic 算法只能根据活动的出现频率来判断噪声,因此有些正确的活动可能会被当作噪声过滤掉,导致重现度降低.Region-Based 挖掘算法产生的模型侧重反映日志中出现过的执行实例[1],因此该算法得到的流程模型的精确度较高.以报修流程为例,日志分类后的其中一个子日志中包含修理移动电话的执行实例.同时使用α算法、Heuristic 算法和 Region-based 算法对该子日志进行挖掘,得到的结果模型分别如图 3(a)~图 3(c)所示.

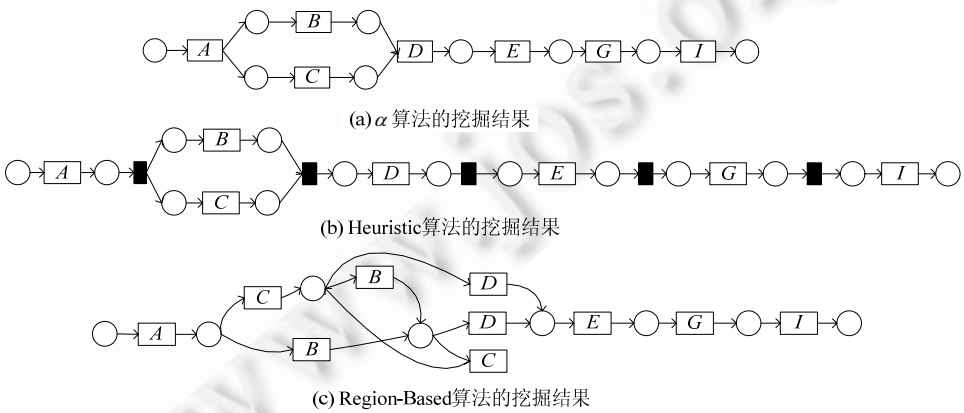


Fig.3 Three results adhering to sublog of preparing mobile phones using different algorithms

图 3 修理移动电话子日志的 3 种挖掘结果

3 个结果模型中:Heuristic 算法得到的流程模型结构正确,但是存在一些不必要的不可见活动(如图 3(b)中的黑色矩形);Region-Based 挖掘算法得到的流程模型能够重现日志,但流程的复杂度较高.本例中,α算法得到的流程模型结构最好,但这只是一次巧合,对于其他日志,α算法不一定能得到完全正确的流程模型.尽管这 3 种算

法对同一个子日志的挖掘结果不同,但是将它们连同其他子日志的挖掘结果一起作为遗传算法的初始种群,利用遗传算法的优化能力最终挖掘得到完整的高质量流程模型.

4 基于遗传算法的流程模型整合

准备好优质种群后,借助遗传算法[9,10,17,18]的优化能力剔除劣质流程,整合优质流程最终得到优化的业务流程模型.遗传算法的关键是流程模型的表示方式、评价流程模型的适应值函数以及遗传算子(杂交、变异)的定义.下面详细介绍遗传算法整合方法.

4.1 流程模型表示方式

本文使用流程树[17]作为流程模型的表示方式,流程树中的节点分为叶子节点和非叶子节点:叶子节点(也称为活动节点)表示执行的活动;非叶子节点(也称为操作节点)表示流程的控制流结构,如顺序、(或)选择、(互斥)选择、并行和循环等.为了简化流程树结构,规定每个节点最多包含两个叶子节点.使用流程树表示的流程模型是一种块结构的流程模型,其最大的好处是流程可避免死锁.5 种控制流结构的流程树表示方法如图 4 所示,其中,→,∨,×,∧,↻分别表示顺序、(或)选择、(互斥)选择、并行和循环结构.

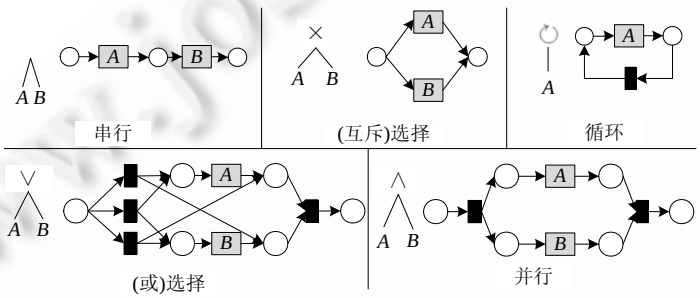


Fig.4 Five control-flow structures using process tree

图 4 5 种控制流结构的流程树表示

4.2 流程模型的质量评价指标

通常从 4 个方面来评价流程模型的质量[19]:重现度(replay fitness)、精确度(precise)、通用性(generalization)和简单性(simplicity).

- 重现度:指流程模型对日志中执行实例的可重现程度.给定一个流程模型和一个执行实例,如果执行实例可通过执行该流程获得,则称该流程可重现该执行实例.流程可重现的执行实例越多,对日志的重现度就越高.计算重现度的基本思想[20]是:首先找到日志与流程模型之间的最佳对应,利用代价函数计算额外活动或遗漏活动产生的代价,即为日志和流程模型之间的偏离度.偏离度越大,流程模型对日志的重现度就越小;
- 精确度:流程模型的精确度也是评价流程模型质量的重要指标之一.如果通过执行流程模型可以产生许多日志中不包含的执行实例,那么该流程模型的精确度就较低.计算精确度的基本思想[19]是:对于模型的每一个状态,如果该状态下能够执行的活动数大大多于日志中实际执行的活动数,则模型的精确度就低;
- 通用性:指未来将要出现的执行实例符合流程模型描述的概率,概率越高,说明流程模型的通用性越好,流程模型应该具备一定的通用性.如果流程模型与日志过度拟合,就可能造成流程模型通用性下降.通用性较难计算,因为人们无法预计未来流程如何执行.计算通用性的基本思想[17]是:在流程树上重现日志后,如果各节点的访问频率都较高,则流程模型的通用性就较好;反之,如果部分节点被访问的频率很小,则流程模型的通用性就较差;

- 简单性:在保证其他 3 个指标的情况下,流程模型显然越简单越好.在评价简单性方面,本文同时考虑了以下 3 个方面[17]:
  (1) 重复活动和遗漏活动越少,流程模型越简单;
  (2) 两个操作节点间的交替变换次数越少,流程模型越简单;
  (3) 循环节点和选择节点越少,流程模型越简单.

4.3 流程模型整合过程

首先,对初始种群中的流程模型(流程树)使用适应值函数计算每个流程模型的质量,按照一定比例选择其中质量最优的多个流程模型,无需任何改变直接保留到下一代.其余流程模型使用锦标赛方法选出并进行杂交、变异后进入下一代,没有被选中的质量较差的模型被淘汰.继续使用适应值函数计算流程模型的质量,与前面的过程一样,高质量的流程模型直接保留到下一代,其余模型使用遗传操作产生.如此迭代下去,直到满足终止条件,挖掘过程停止.通过这种精英选择和遗传操作,每一代种群中的最优流程模型的质量会变得越来越好,末代种群中质量最高的流程模型即是最终挖掘结果.下面详细介绍适应值函数和遗传算子的定义.

4.3.1 适应值函数

一般的流程挖掘算法只能兼顾某些方面的质量指标,例如,使用 Region-Based 挖掘算法产生的流程模型的重现度和精确度较好,但是模型的通用性和简单性较差[6].而遗传算法通过适应值函数,能在挖掘的过程中监控流程模型 4 个方面的质量指标.本文采用设置权重的方式[2,17]将 4 个质量指标(见第 4.2 节)综合起来,使得产生的结果模型具有较高的综合质量,即重现度、精确度、通用性和简单性.适应值函数的计算公式为

$$fitness=w_1\times Fr+w_2\times Pe+w_3\times Gn+w_4\times Sm \tag{1}$$

其中,Fr,Pe,Gn 和 Sm 分别为流程模型在重现度、精确度、通用性和简单性这 4 方面的计算值;w_1,w_2,w_3 和 w_4 分别是 4 个质量指标的权重.用户可以根据自己的偏好设置流程模型在这 4 方面的权重.

4.3.2 适用于流程模型表示的遗传算子

利用适应值函数计算当前所有流程模型的适应值,按照一定比例,将适应值最高的多个流程模型直接保留到下一代.其余的流程使用锦标赛方法选出并通过杂交变异[17]产生.具体方法如下:

(1) 杂交

参与杂交的两棵流程树随机选择各自的子树进行交换.已知两棵流程树 $P1$,$P2$,随机选中各自的一棵子树,如图 5 所示,$P1$ 选中子树 $st1$,$P2$ 选中子树 $st2$.将两棵子树交换后,得到两棵新的流程树 $P1'$,$P2'$.

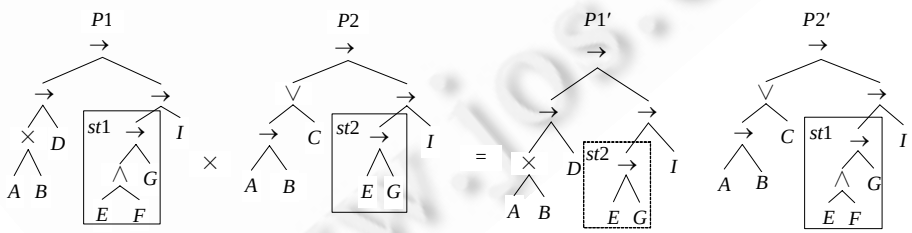


Fig.5 Crossover of two process trees

图 5 两棵流程树的杂交

(2) 变异

变异分为 3 种情况:节点变异、删除节点、添加活动节点.

- 节点变异包括操作节点(非叶子节点)变异和活动节点(叶子节点)变异.对于操作节点,改变其代表的控制流结构类型;对于活动节点,改变其代表的活动类型.例如:将流程树 $P1$ 中的代表并行结构的操作节点(节点 E 的父节点)变成顺序结构,变异结果如图 6(b)所示;将 $P1$ 中活动节点 D 的活动类型变成 C,变异结果如图 6(c)所示;

- 删除节点指随机选中一个节点,将其连同所有子节点一起删除.为了保证流程树的正确结构,有时需要同时删除其父节点.例如,要将流程树 $P1$ 中的活动节点 E 删除,需要同时删除它的父节点,并将节点 F 变成节点 G 的兄弟节点,变异结果如图 6(d)所示.
- 添加活动节点指随机产生一个活动节点,将它添加到任意一个操作节点下.为了保证流程树的正确结构,有时需要同时添加父节点,父节点的控制流类型随机指定.例如,在流程树 $P1$ 中,将活动节点 C 添加到活动节点 D 的父节点下.在该父节点下创建一个同样代表顺序结构的操作节点,将节点 C 和 D 添加到该操作节点下,变异结果如图 6(e)所示.

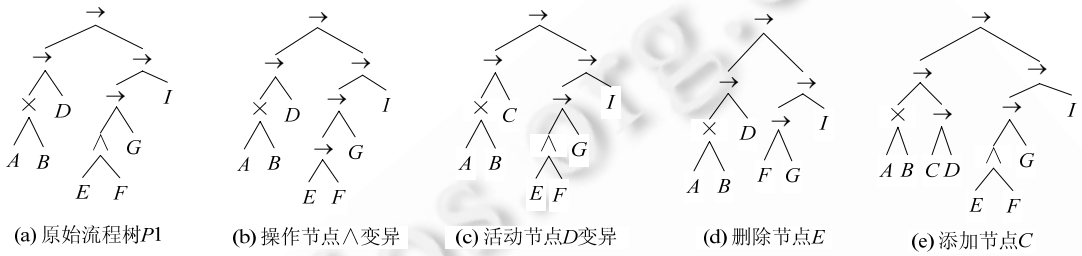


Fig.6 Mutation of two process trees

图 6 两棵流程树的变异

5 实验与结果分析

针对本文提出的流程模型挖掘方法 SoFi,本节进行了两组实验来分析验证该方法的有效性:第 1 组实验采用报修流程的模拟日志,第 2 组实验采用中国移动公司 3 个省市的发文流程日志.对每组实验,设计了 5 个实验方案,使用 ProM[22]工具挖掘优质初始种群,实现了 GA 优化器和其中的适应值函数.下面对实验结果进行分析和比较.

5.1 实验设计与参数设置

第 1 组实验的日志来自第 2 节介绍的报修流程,利用日志产生工具 PLG(process log generator)[23]产生 1 000 个执行实例,共 6 760 条事件信息.每个执行实例都包含各自的数据对象,根据这些数据对象提供的信息将执行实例分成 3 类:第 1 类包含 390 个执行实例,第 2 类包含 450 个执行实例,第 3 类包含 160 个执行实例.第 2 组实验的日志采用中国移动公司在北京、上海、湖南 3 个省市的发文流程的运行日志,该日志共包含 1 800 个执行实例,14 190 条事件信息.根据数据对象提供的信息将日志分成 3 类:第 1 类包含 550 个执行实例,第 2 类包含 420 个执行实例,第 3 类包含 630 个执行实例.对每组实验设计实验方案如下:

(1) 按照本文提出的 SoFi 方法,为分类后的各子日志分别施用α算法、Heuristic 算法和 Region-based 算法为 GA 优化器准备初始种群;

(2) 对分类后的各子日志施用同一种挖掘算法为 GA 优化器准备初始种群.使用α算法、Heuristic 算法和 Region-based 算法各进行一次.3 次实验分别用 AlphaForGA、HeuForGA 和 RegForGA 表示;

(3) 对原始日志直接施用α算法、Heuristic 算法和 Region-based 算法为 GA 优化器准备初始种群.用 SoFiNoClassify 表示;

(4) 对原始日志直接施用 GA 算法进行挖掘;

(5) 对原始日志直接施用α算法、Heuristic 算法和 Region-based 算法,对挖掘结果利用公式(1)计算模型的综合质量.

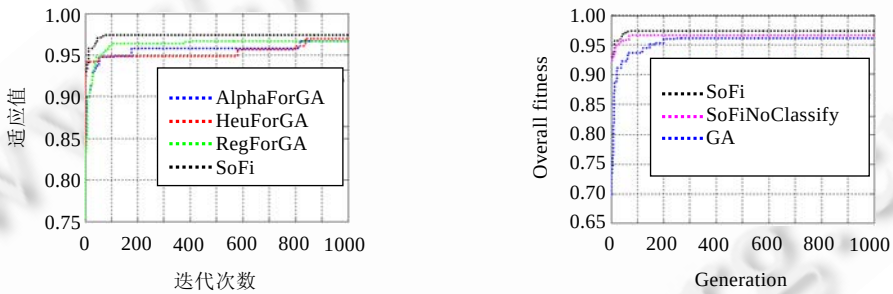
在两组实验中,为 GA 优化器设置种群大小为 100,精英选择的比例为 0.3.因为变异操作比杂交操作对于获得高质量流程模型的贡献更大[24],因此,本实验设参与杂交和变异的流程模型的比例分别为 0.1 和 0.5.对于适应值函数中的各权重设置,因为重现度和精确度是评价流程质量最重要的两个指标[19],所以这两方面的权重应设

得高一些.因此,适应值函数中各权重分别设为 w_1=0.4,w_2=0.4,w_3=0.1,w_4=0.1.设置迭代次数达到 1 000 代时,挖掘过程终止.

5.2 实验结果与分析

对两组日志分别进行上述 5 个实验.方案 1 的 SoFi 方法和方案 2 中的 AlphaForGA,HeuForGA 和 RegForGA 方法的流程模型适应值变化过程如图 7(a)和图 8(a)所示.图中,横坐标表示 GA 优化器迭代的次数,纵坐标表示流程模型的适应值.与方案 1 的 SoFi 方法不同,方案 2 对分类后的各子日志施用单一挖掘算法为 GA 优化器准备初始种群.如图 7(a)和图 8(a)所示,SoFi 方法的流程模型适应值比 AlphaForGA,HeuForGA 和 RegForGA 方法的流程模型均更快地收敛,并且最终结果模型的适应值优于 AlphaForGA,HeuForGA 和 RegForGA 方法的结果模型.这是因为相比单一的挖掘算法,SoFi 方法对每个子日志施用各种不同的挖掘算法增加了子日志遇到合适算法的可能性,提高了初始种群的质量.

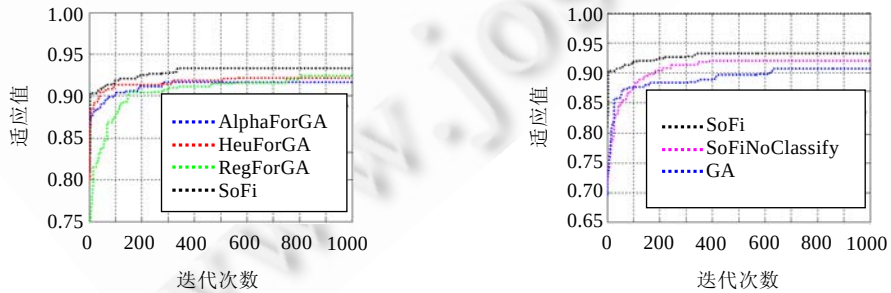
方案 1、方案 3 和方案 4 的流程模型适应值变化过程如图 7(b)和图 8(b)所示.图中,SoFi 方法的流程模型适应值比 SoFiNoClassify 和 GA 方法的流程模型均更快地收敛,并且最终结果模型的适应值优于 SoFiNoClassify 和 GA 方法的结果模型.与 SoFi 方法不同,SoFiNoClassify 方法没有对原日志进行分类,直接使用α,Heuristic 和 Region-based 这 3 种不同的算法为 GA 优化器准备初始种群.而 SoFi 方法对原始日志分类的好处是降低了日志多样性,使传统挖掘算法的特征和优势得到充分发挥,从而得到更好的初始种群,因此最终流程模型的综合质量更好.



(a) SoFi,AlphaForGA,HeuForGA 和 RegForGA 适应值变化过程 (b) SoFi,SoFiNoClassify 和 GA 适应值变化过程

Fig.7 Experimental results using synthetic log of repair process

图 7 采用报修流程日志的实验结果



(a) SoFi,AlphaForGA,HeuForGA 和 RegForGA 适应值变化过程 (b) SoFi,SoFiNoClassify 和 GA 适应值变化过程

Fig.8 Experimental results using real-life log of document issue process

图 8 采用发文流程日志的实验结果

方案 5 对原始日志直接施用α,Heuristic 和 Region-based 算法得到 3 个流程模型.先计算它们 4 个方面的模型质量,再使用公式(1)计算得到流程模型综合质量,计算结果见表 3.

Table 3 Comparing the qualities of mined process models using α, Heuristic, Region-based and SoFi algorithms

表 3 α,Heuristic,Region-based 与 SoFi 方法的挖掘结果比较

	实验 1				实验 2			
	α	Heuristic	Region-Based	SoFi	α	Heuristic	Region-Based	SoFi
重现度	0.856 8	0.757 3	0.799 8	**0.979 5**	0.565 9	0.766 9	0.770 7	**0.920 3**
精确性	**1.000 0**	0.935 6	0.756 3	0.983 6	0.864 4	0.884 4	0.854 2	**0.974 5**
通用性	0.873 4	0.754 6	0.497 6	**0.882 8**	0.649 9	0.649 5	0.523 5	**0.837 1**
简单性	**1.000 0**	0.997 6	0.936 5	0.988 1	**0.996 0**	0.876 0	0.774 5	0.898 6
适应值	0.930 1	0.852 4	0.765 9	**0.971 1**	0.812 9	0.813 1	0.779 8	**0.931 5**

实验 1 中:α算法获得的模型简单性和精确性均最高,为 1.000;Region-Based 算法获得的模型简单性最低,为 0.936 5;Heuristic 算法获得的模型重现度最低,为 0.757 3;而 SoFi 方法获得的流程模型虽然在个别质量维度上的计算值小于以上 3 种挖掘算法,但综合适应值优于这 3 种挖掘算法.实验 2 中:α算法获得的模型的重现度最低,为 0.565 9,因为真实日志的多样性更佳显著,α算法不能一次处理所有的复杂结构;Region-Based 算法获得的模型的简单性最低,为 0.774 5;而 SoFi 方法获得的流程模型除简单性外,其余维度的质量计算值均优于以上 3 种挖掘算法,综合适应值也优于这 3 种挖掘算法.

以上实验结果表明,采用 SoFi 方法先对日志进行分类降低日志多样性,再对分类后的子日志施用各种不同的挖掘算法,能够为 GA 优化器提供高质量的初始种群.而 GA 优化器能够整合各种挖掘算法的优势,使最终流程模型的综合质量均优于任何参与其中的单一挖掘算法.

6 相关工作

流程挖掘是从系统的执行日志中发现流程模型,在这方面已出现了许多挖掘算法和工具[1,25].Aalst 等人[3]提出的α算法通过发现活动间的二元关系构建控制流结构,该算法适合于不带短循环和隐藏库所的结构化流程模型,它不能处理非局部选择结构.Wen 等人[4]提出的α^{++}算法是对α算法的扩展,它能够处理短循环和非局部选择结构,但是不能处理日志噪声.Weijters 等人[5,6]提出的启发式挖掘算法考虑了活动间跟随关系的频率,因此该算法能够处理日志噪声,但是不能处理非局部选择结构和重复活动.Van Dongen 等人[7,8]提出的两阶段挖掘算法先在日志层建立活动的二元关系模型,然后再将这些模型合并形成顺序和选择结构,该算法不能挖掘循环结构.Region-Based 挖掘算法[6]先从日志产生变迁系统,然后再转换成 Petri 网模型.流程模型的重现度和精确度较好,但是模型往往过于复杂以至于无法理解.基于 GA 的挖掘算法根据模型表示方法不同又分为两种:一种是基于因果矩阵的挖掘算法[2];另一种是基于流程树的挖掘算法[17].前一种方法挖掘出来的模型可能存在死锁;而后一种方法能够保证流程模型的正确性,避免死锁情况的产生.这些传统的流程挖掘算法均有各自的特色和适用对象,无法适用于复杂多变的应用环境.而本文提出的 SoFi 方法能够解决日志的多样性问题,并充分发挥各种挖掘算法的特征和优势.

评价挖掘得到的流程模型是否正确,反映了实际的执行情况属于一致性检测问题[20].通常从 4 个方面[19]——重现度、精确度、通用性和简单性来评价流程模型的质量.文献[26]的方法是在模型上重现日志中的执行实例,统计重现过程中添加或去除的令牌数,然后计算 4 个方面的评价值.文献[21]的方法是找到日志与模型的最佳对应方案,利用对应状态机计算流程模型的质量.文献[20]针对大规模日志提出了基于 A*算法的日志与模型对应方法,通过识别遗漏活动和额外活动来计算流程模型质量.

处理大规模多样性日志方面,一种方法是对日志进行分类预处理.借助数据挖掘中的聚类方法,可以对日志中的执行实例进行分类.文献[11]根据活动构造执行实例的特征向量.文献[12]考虑了活动的上下文,根据执行模式构造特征向量,然后使用常用的聚类算法对执行实例进行聚类.文献[13]通过计算各执行实例间的编辑距离来进行聚类.这些方法都是从活动角度提取执行实例的特征,而 SoFi 方法充分利用日志中的数据信息结合领域知识对日志进行分类.

7 结 论

复杂多变的运行环境,使得相应的日志呈现出多样性.传统的挖掘算法各有其适用对象,因此,如何挑选适合多样性流程日志的流程挖掘算法成为了一项挑战.本文提出了一种适用于多样性应用环境的业务流程挖掘方法.先根据日志中的数据信息利用领域知识对日志进行分类,采用多种传统流程挖掘算法对每一个子日志进行挖掘得到多种流程模型.最后,采用基于遗传算法的流程挖掘方法将这些流程模型的优点整合起来,通过设置适应值函数,最终得到综合质量较高的完整流程模型.实验结果表明,采用 SoFi 方法不仅能够加快算法的收敛速度,而且最终产生的流程模型的综合质量均优于直接使用一种挖掘算法所产生的流程模型.

本文使用流程树作为 GA 优化器的流程模型表示方法,因为流程树表示的流程不会产生死锁,流程具有更高的正确性.但是,尝试多种挖掘算法得到的流程模型结构不一定完全能用流程树表示.本文将多种挖掘算法得到的流程模型尽可能地转换成流程树.未来将考虑使用因果矩阵作为 GA 优化器的流程模型表示方法,但如何保证流程模型的正确性是需要进一步解决的问题.

References:

[1] Van der Aalst WMP. Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer-Verlag, 2011.

[2] Van der Aalst WMP, De Medeiros AKA, Weijters AJMM. Genetic process mining. In: Proc. of the Applications and Theory of Petri Nets 2005. LNCS 3536, Springer-Verlag, 2005. 48−69.

[3] Van der Aalst W, Weijters T, Maruster L. Workflow mining: Discovering process models from event logs. IEEE Trans. on Knowledge and Data Engineering, 2004,16(9):1128−1142. [doi: 10.1109/TKDE.2004.47]

[4] Wen LJ, Wang JM, Sun JG. Detecting implicit dependencies between tasks from event logs. In: Proc. of the Frontiers of WWW Research and Development-APWeb 2006. Springer-Verlag, 2006. 591−603. [doi: 10.1007/11610113_52]

[5] Weijters AJMM, Ribeiro JTS. Flexible heuristics miner (FHM). In: Proc. of the IEEE Symp. on Computational Intelligence and Data Mining (CIDM). IEEE, 2011. 310−317. [doi: 10.1109/CIDM.2011.5949453]

[6] Weijters AJMM, van der Aalst WMP. Rediscovering workflow models from event-based data using little thumb. Integrated Computer Aided Engineering, 2003,10(2):151−162.

[7] Van Dongen BF, Van der Aalst WMP. Multi-Phase process mining: Aggregating instance graphs into EPCs and Petri nets. In: Proc. of the 2nd Int'l Workshop on Applications of Petri Nets to Coordination, Workflow and Business Process Management (PNCWB). Citeseer, 2005.

[8] Van Dongen BF, Van der Aalst WMP. Multi-Phase process mining: Building instance graphs. In: Proc. of the Conceptual Modeling–ER 2004. Springer-Verlag, 2004. 362−376. [doi: 10.1007/978-3-540-30464-7_29]

[9] De Medeiros AKA, Weijters AJMM, Van der Aalst WMP. Genetic process mining: an experimental evaluation. Data Mining and Knowledge Discovery, 2007,14(2):245−304. [doi: 10.1007/s10618-006-0061-7]

[10] De Medeiros AKA, Weijters AJMM, Van der Aalst WMP. Genetic process mining: A basic approach and its challenges. In: Proc. of the Business Process Management Workshops. Springer-Verlag, 2006. 203−215. [doi: 10.1007/11678564_18]

[11] Song M, Günther CW, Van der Aalst WMP. Trace clustering in process mining. In: Proc. of the Business Process Management Workshops. Springer-Verlag, 2009. 109−120. [doi: 10.1007/978-3-642-00328-8_11]

[12] Bose RPJC, Van der Aalst WMP. Trace clustering based on conserved patterns: Towards achieving better process models. In: Proc. of the Business Process Management Workshops. Springer-Verlag, 2010. 170−181. [doi: 10.1007/978-3-642-12186-9_16]

[13] Bose RPJC, Van der Aalst WMP. Context aware trace clustering: Towards improving process mining results. In: Proc. of the SDM. SIAM, 2009. 401−412.

[14] Kumaran S, Liu R, Wu FY. On the duality of information-centric and activity-centric models of business processes. In: Proc. of the Advanced Information Systems Engineering. Springer-Verlag, 2008. 32−47. [doi: 10.1007/978-3-540-69534-9_3]

[15] Nigam A, Caswell NS. Business artifacts: An approach to operational specification. IBM Systems Journal, 2003,42(3):428-445. [doi: 10.1147/sj.423.0428]

[16] Cortadella J, Kishinevsky M, Lavagno L, Yakovlev A. Deriving Petri nets from finite transition systems. IEEE Trans. on Computers, 1998,47(8):859−882. [doi: 10.1109/12.707587]

[17] Buijs J, Van Dongen BF, Van der Aalst WMP. A genetic algorithm for discovering process trees. In: Proc. of the 2012 IEEE Congress on Evolutionary Computation (CEC). IEEE, 2012. 1–8. [doi: 10.1109/CEC.2012.6256458]

[18] Aalst WMP, Medeiros AKA, Weijters AJMM. Genetic process mining. In: Proc. of the Applications and Theory of Petri Nets. 2005. 985–985. [doi: 10.1007/11494744_5]

[19] Buijs J, Van Dongen BF, Van der Aalst WMP. On the role of fitness, precision, generalization and simplicity in process discovery. In: Proc. of the Move to Meaningful Internet Systems: OTM 2012. Springer-Verlag, 2012. 305–322. [doi: 10.1007/978-3-642-33606-5_19]

[20] Adriansyah A, Van Dongen BF, Van der Aalst WMP. Conformance checking using cost-based fitness analysis. In: Proc. of the 2011 15th IEEE Int'l Enterprise Distributed Object Computing Conf. (EDOC). IEEE, 2011. 55–64. [doi: 10.1109/EDOC.2011.12]

[21] Adriansyah A, Munoz-Gama J, Carmona J, Van Dongen BF, Van der Aalst WMP. Alignment based precision checking. In: Proc. of the Business Process Management Workshops. Springer-Verlag, 2013. 137–149. [doi: 10.1007/978-3-642-36285-9_15]

[22] Van Dongen BF, De Medeiros AKA, Verbeek HMW, Weijters AJMM, Aalst WMP. The ProM framework: A new era in process mining tool support. In: Proc. of the Applications and Theory of Petri Nets 2005. Springer-Verlag, 2005. 444–454. [doi: 10.1007/11494744_25]

[23] Burattin A, Sperduti A. PLG: A framework for the generation of business process models and their execution logs. In: Proc. of the Business Process Management Workshops. Springer-Verlag, 2011. 214–219. [doi: 10.1007/978-3-642-20511-8_20]

[24] Eck M. Alignment-Based Process Model Repair and its Application to the Evolutionary Tree Miner., 2013

[25] Zeng QT, Sun SX, Huan H, Liu C, Wang HQ. Cross-Organizational collaborative workflow mining from a multi-source log. In: Proc. of the Decision Support Systems. 2013. 1280–1301. [doi: 10.1016/j.dss.2012.12.001]

[26] Rozinat A, Van Der Aalst WMP. Conformance checking of processes based on monitoring real behavior. Information Systems, 2008,33(1):64–95. [doi: 10.1016/j.is.2007.07.001]

杨丽琴(1982－),女,上海人,讲师,CCF 学生会员,主要研究领域为业务流程管理,服务计算.

康国胜(1985－),男,硕士,CCF 学生会员,主要研究领域为以数据为中心的业务流程配置,服务计算.

郭立鹏(1984－),男,学士,主要研究领域为业务流程管理,服务计算.

田朝阳(1990－),男,硕士,主要研究领域为业务流程管理.

张亮(1963－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为业务流程管理,服务计算.

张笑楠(1980－),男,工程师,主要研究领域为智能业务流程管理.

高翔(1956－),男,教授级高工,主要研究领域为云计算,计量经济学,智能业务流程管理.