软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2018,29(2):396−416 [doi: 10.13328/j.cnki.jos.005253]

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

基于混合人工免疫算法的流程挖掘事件日志融合方法*

徐 杨[1], 袁 峰[2], 林 琪[1], 汤德佑[1], 李 东[1]

[1](华南理工大学 软件学院,广东 广州 510006)

[2](广州中国科学院 软件应用技术研究所,广东 广州 511458)

通讯作者: 李东, E-mail: cslidong@scut.edu.cn

摘 要: 流程挖掘是流程管理和数据挖掘交叉领域中的一个研究热点.在实际业务环境中,流程执行的数据往往分散记录到不同的事件日志中,需要将这些事件日志融合成单一事件日志文件,才能应用当前基于单一事件日志的流程挖掘技术.然而,由于流程日志间存在着执行实例的多对多匹配关系、融合所需信息可能缺失等问题,导致事件日志融合问题具有较高的挑战性.对事件日志融合问题进行了形式化定义,指出该问题是一个搜索优化问题,并提出了一种基于混合人工免疫算法的事件日志融合方法:以启发式方法生成初始种群,以人工免疫系统的克隆选择理论作为基础,通过免疫进化获得"最佳"的融合解,从而支持包含多对多的实例匹配关系的日志融合;考虑两个实例级别的因素——流程执行路径出现的频次和流程实例间的时间匹配关系,分别从"量"匹配和"时间"匹配两个维度来评价进化中的个体;通过设置免疫记忆库、引入模拟退火机制,保证新一代种群的多样性,减少进化早熟几率.实验结果表明:该方法能够实现多对多的实例匹配关系的事件日志融合的目标,相对于随机方法生成初始种群,启发式方法能够加快免疫进化的速度.另外,针对利用分布式技术提高事件日志融合性能,探讨了大规模事件日志分布式融合中的数据划分问题.

关键词: 事件日志融合;流程挖掘;人工免疫系统;日志预处理

中图法分类号: TP181

中文引用格式: 徐杨,袁峰,林琪,汤德佑,李东.基于混合人工免疫算法的流程挖掘事件日志融合方法.软件学报,2018,29(2):396−416. http://www.jos.org.cn/1000-9825/5253.htm

英文引用格式: Xu Y, Yuan F, Lin Q, Tang DY, Li D. Merging event logs for process mining with a hybrid artificial immune algorithm. Ruan Jian Xue Bao/Journal of Software, 2018,29(2):396−416 (in Chinese). http://www.jos.org.cn/1000-9825/5253.htm

Merging Event Logs for Process Mining with a Hybrid Artificial Immune Algorithm

XU Yang[1], YUAN Feng[2], LIN Qi[1], TANG De-You[1], LI Dong[1]

[1](School of Software Engineering, South China University of Technology, Guangzhou 510006, China)

[2](Institute of Software Application Technology, Guangzhou & Chinese Academy of Sciences, Guangzhou 511458, China)

Abstract: Process mining is an active research topic in the cross field of process management and data mining. In an actual business environment, the recorded data of a process execution that may be supported by different computer systems is scattered into different event log files. It is necessary to merge the scattered data into one single event log file when applying current process mining techniques

* 基金项目: 国家自然科学基金(71090403); 广东省科技计划(2014B090901001, 2015B010103002, 2016B090918062, 2016B050502001); 广州市科技计划(201604010127); 华南理工大学软件学院 985 学科建设基金(x2rjD615015III)

Foundation item: National Natural Science Foundation of China (71090403); Science and Technology Planning Projects of Guangdong Province (2014B090901001, 2015B010103002, 2016B090918062, 2016B050502001); Science and Technology Planning Projects of Guangzhou City (201604010127); Special Funds on "985 Project" Disciplinary Construction in School of Software Engineering of South China University of Technology (x2rjD615015III)

收稿时间: 2016-10-10; 修改时间: 2016-12-12; 采用时间: 2017-01-07; jos 在线出版时间: 2017-03-24

CNKI 网络优先出版: 2017-03-24 17:09:33, http://kns.cnki.net/kcms/detail/11.2560.TP.20170324.1709.012.html

and tools for process mining. This mission is still challenging, however, because of the complex relationships between cases in two logs and the possible lack of information for the merging. In this paper, event log merging for process mining is regard as a type of search and optimization problems based on the formal definition, and a merging approach with a hybrid artificial immune algorithm is presented in order to achieve the event log merging with many to many relationship between cases in the two event logs. In the merging approach, the clonal selection principle is selected as its underlying principle, which requires the matching process to undergo iterations of clonal selection, hypermutation and receptor editing in order to get the best solution. The algorithm starts from an initial population produced with a heuristic approach. Two factors, occurrence frequency and temporal relation, are designed in the affinity function to evaluate the individuals in the population. In addition, immunological memory and simulated annealing are exploited to make the artificial immune merging jumping out from the trap of local optima. Experimental results show that the hybrid algorithm has good performance in merging logs with complex cases relationships, and the heuristic approach for initial population can speed the process of the evolution. This paper also discusses the data distribution methods in which the log merging problems can be distributed.

Key words: event log merging; process mining; artificial immune system; log preprocessing

随着 Web 服务技术、Web 2.0 技术和云计算技术的广泛应用,组织的业务变得越来越灵活,业务流程的执行路径根据变化的业务需求、客户需求和人员技能在运行时动态地变化,不能通过建模方法预先明确,业务流程越来越呈现非结构化和半结构化的特性.目前,大多数流程分析研究工作都假设流程是结构良好(well-structured)的.这种情形下,流程的理解和流程执行数据的分析变得十分困难[1].流程挖掘(process mining)[2]结合流程建模分析技术和数据挖掘技术,将信息系统执行过程中产生的流程日志看作是一个事件网,从这个事件网中抽取流程知识,从而发现流程模型,识别流程执行的瓶颈,提供对流程执行的量化理解,为流程分析提供了一条新的途径[3].

在企业实际业务环境中,信息系统往往用于支撑流程的执行.但大部分情况下,这些信息系统并不管理整个业务流程,而是处理流程中的某些活动.因此,流程执行数据被分散记录到不同的日志中,结果是日志数据逐渐呈现出大数据的特性:物理分布的广泛性、数据格式的多样性、数据模型的非标准化以及语义的异构性等[4].然而,当前的流程挖掘技术或工具,比如 Heuristics Miner[5]、Generic Miner[6]、Distributed Process Mining[7]和 Conformance Checking[8]都是基于单个事件日志文件的.因此,对于分散在不同信息系统的日志数据,需要将其集成为能够完整描述整个业务流程的单一日志数据文件,才能应用这些研究成果进行流程分析.

流程挖掘技术是以事件日志(event log)作为输入.在事件日志中,一个事件(event)表示流程中一个活动的执行,每个事件都归属于某个流程执行实例,而且流程执行实例中的事件是有序的.一个事件日志包含而且仅包含与某一个流程相关的事件.从数据源获得的原始流程日志需要经过事件抽取、实例识别等操作转换成事件日志.事件日志的融合就是将多个事件日志文件融合成单一日志文件的过程,主要步骤包括:识别出不同日志中记录的两个执行实例是不是属于同一业务流程的执行实例(本文称为全局流程执行实例),如果属于同一全局流程执行实例,则根据流程执行实例中活动发生的时间先后顺序合并成一个全局流程执行实例,并将其写入新的日志文件中.其中的关键是如何识别两个流程执行实例是不是属于同一全局流程执行实例,即如何在给定两个流程执行实例集合上建立执行实例之间的匹配关系.

直观上看,可以通过比较某个或几个两个执行实例的属性来判断是否具有匹配关系,比如通过比较执行实例的标识是否相同来确定;或者通过比较执行实例中活动属性,比如活动的时间戳来决定流程实例间是否存在匹配关系.然而,实际情况要复杂得多.

事件日志融合问题复杂性的一个重要表现就是,实际的业务流程本身的灵活性导致两个日志中的执行实例的匹配关系不是简单的一对一关系.表 1 是一个简化了的 IT 事件管理系统的日志片断 log1.在 IT 事件管理流程中,IT 系统的用户向 IT 事件管理系统提交 IT 事件(incident),IT 事件经过登记、分析后,就分派给 1 个或多个 IT 运维任务进行处理.运维任务分派后就被纳入任务管理系统进行管理.表 2 是一个简化了的任务管理系统的日志片断 log2.一个任务在其生命周期中经历新建、分配、执行、完成、评价及关闭多个阶段.

在 IT 事件管理流程中,通过分析,登记的不同 IT 事件如果产生的原因相同,则会将这些事件合并分派给同一个任务来处置.表 1 中 IT 事件 INCID2016060102(“无法访问邮件系统”)和 INCID2016060113(“无法登录某业

务系统")产生的原因相同("网络交换机 SW021101 故障"),则把这两个 IT 事件合并,向任务管理系统发送一个任务请求.任务管理系统接到事件管理系统的任务请求后则会产生一个运维任务 TK000005(见表 2).这样,log1 中的执行实例 IN000001 和 IN000003 与 log2 中的执行实例 TK000005 形成多对一的匹配关系.log1 与 log2 日志融合时,需要分别将 IN000001,IN000003 与 TK000005 进行融合.表 1 中的流程执行实例 IN000045 则和表 2 的流程执行实例 TK000012 形成一对一的匹配关系.实际上,IT 事件管理流程中,除了多对一和一对一的关系外,还存在一个 IT 事件由多个任务处置的一对多以及多个 IT 事件由多个任务处置的多对多的匹配关系.因此,在进行事件日志的融合时,需要考虑执行实例的多对多匹配关系,提高了日志融合问题的复杂度,而且日志规模越大,该复杂度就越高.

Table 1 Log1 for the simplified incident management process (fragments)

表 1 简化的事件管理流程日志 log1(片段)

CaseID	IncidentID	Activity	TimeStamp
IN000001	INCID2016060102	INCIDENT_REGISTER	2016-06-06T23:08:04
IN000001	INCID2016060102	INCIDENT_ASSIGN	2016-06-07T09:18:31
IN000001	INCID2016060102	INCIDENT_HANDLE	2016-06-07T09:28:58
IN000001	INCID2016060102	INCIDENT_SOLVE	2016-06-07T11:42:08
IN000001	INCID2016060102	INCIDENT_CONFIRM	2016-06-07T12:52:35
IN000001	INCID2016060102	INCIDENT_CLOSE	2016-06-07T14:03:02
IN000003	INCID2016060113	INCIDENT_REGISTER	2016-06-07T01:13:29
IN000003	INCID2016060113	INCIDENT_ASSIGN	2016-06-07T09:18:31
IN000003	INCID2016060113	INCIDENT_HANDLE	2016-06-07T09:28:58
IN000003	INCID2016060113	INCIDENT_SOLVE	2016-06-07T11:42:08
IN000003	INCID2016060113	INCIDENT_CONFIRM	2016-06-07T12:52:35
IN000003	INCID2016060113	INCIDENT_CLOSE	2016-06-07T14:03:02
IN000045	INCID2016060463	INCIDENT_REGISTER	2016-06-13T10:37:07
IN000045	INCID2016060463	INCIDENT_ASSIGN	2016-06-13T10:47:34
IN000045	INCID2016060463	INCIDENT_HANDLE	2016-06-13T11:58:01
IN000045	INCID2016060463	INCIDENT_SUSPEND	2016-06-13T12:08:28
IN000045	INCID2016060463	INCIDENT_HANDLE	2016-06-13T13:08:28
IN000045	INCID2016060463	INCIDENT_SOLVE	2016-06-13T14:29:22
IN000045	INCID2016060463	INCIDENT_CONFIRM	2016-06-13T14:39:49
IN000045	INCID2016060463	INCIDENT_CLOSE	2016-06-13T14:50:17

Table 2 Log2 for the simplified task management process (fragments)

表 2 简化的任务管理流程日志 log2(片段)

CaseID	Activity	RelatedIncidentID	TimeStamp
TK000005	TASK_NEW	INCID2016060102, INCID2016060113	2016-06-07T09:18:32
TK000005	TASK_ASSIGN	INCID2016060102, INCID2016060113	2016-06-07T09:20:52
TK000005	TASK_EXECUTE	INCID2016060102, INCID2016060113	2016-06-07T09:28:58
TK000005	TASK_COMPLETE	INCID2016060102, INCID2016060113	2016-06-07T11:42:08
TK000005	TASK_EVALUATE	INCID2016060102, INCID2016060113	2016-06-07T16:50:47
TK000005	TASK_CLOSE	INCID2016060102, INCID2016060113	2016-06-07T16:55:37
TK000012	TASK_NEW	NULL	2016-06-20T10:47:35
TK000012	TASK_ASSIGN	NULL	2016-06-18T10:55:18
TK000012	TASK_EXECUTE	INCID2016060463	2016-06-13T11:58:00
TK000012	TASK_DISPATCH	INCID2016060463	2016-06-13T12:15:44
TK000012	TASK_EXECUTE	INCID2016060463	2016-06-13T13:08:27

日志融合需要面临的另一个问题是流程日志融合所需数据或信息缺失问题.事件日志中的实例标识是用于区分唯一实例的重要属性,通过分析两个日志中的实例标识是否一致来判定两个实例是否可以融合[9].但这个属性可能会缺失;或者,即使两个事件日志中存在实例标识,但两个标识是异构的,比如表 1 和表 2 中的 CaseID 情况.因而,不能用实例标识来判定两个实例是否匹配.又如,两个实例可以通过某些关联属性来判断它们是否匹配,如表 2 中实例 TK000005 中的 RelatedIncidentID 属性关联了表 1 中的实例 IN000001 的 IncidentID,可通过两者是否匹配来判断两实例是否属于全局流程执行实例.但可能出现的情况是,如表 2 中实例 TK000012 中,缺失了关联属性 RelatedIncidentID 的属性值.这类问题本文都称为融合所需信息缺失问题.

日志融合需要面临的第 3 个问题是事件日志的"噪声".实际上,"噪声"问题是流程挖掘技术在各个阶段都面临的问题.在日志融合中,"噪声"主要体现在:由于日志系统的错误或传输通道的不可靠等问题导致的日志中

记录的活动和实际执行活动不一致;由于活动开始和(或)结束时间的被延迟记录等问题而产生的活动执行序列与实际不一致.这些都会影响对两个实例是否匹配的分析结果的正确性.

目前,有关事件日志融合问题的研究成果还不多见.本文把相关研究分为两类:一类是事件级别的日志融合;另一类是实例级别的日志融合.

虽然没有文献明确提出事件级别的日志融合方法,但可以利用现有的事件关联(event correlation)技术[10,11]进行日志融合.事件关联技术是用于将流程日志转换成结构化事件日志的一种自动化技术,这类技术主要是通过分析记录在日志中的活动属性之间的关联关系,将属于同一个流程实例的活动进行聚类,从而识别出日志中流程实例.对于日志融合问题,可以将待融合的两个日志中的活动集合合成一个大的活动集合,再利用事件关联技术将属于同一流程执行的事件归集到同一实例中,从而完成两个日志的融合,形成新的日志.事件关联技术处理的数据主要是与流程活动(事件)相关的属性,因而称其为事件级别的融合技术.事件级别的融合技术的问题在于:当两个日志的活动集合合在一起时,其处理的活动空间将迅速扩大,事件关联算法性能下降快.

实例级别的日志融合技术的典型代表是 Claes 等人的研究[9,12,13].Claes 将遗传算法和人工免疫算法进化算法引入到日志融合里,将日志融合问题转换为搜索与寻优问题,其核心是寻找两个日志之间最优的实例匹配集,给日志数据融合提供了一种较好的解决思路.然而,Claes 的研究成果主要针对事件日志中仅包含实例之间一对一的匹配关系,不能处理复杂的一对多、多对一或多对多的实例匹配关系.另外,在 Claes 的人工免疫方法(artificial immune algorithm,简称 AIA)[9]中假定了实例标识是事件日志中实例应具备的属性,并在对算法解的评价中假定了两个日志采用统一的实例标识方法.这些都限制了 Claes 的 AIA 方法的适应性.

本文基于人工免疫系统的克隆选择理论,提出了一种事件日志融合的自动化方法——混合人工免疫算法(hybrid artificial immune algorithm,简称 HAIA).这种方法不仅可以完成包含一对一实例匹配关系的事件日志融合问题,还支持包含一对多、多对一或多对多的实例匹配关系的日志融合.本文主要工作包括:(1) 探讨了事件日志融合问题的形式化描述,把实例级别的日志融合问题的解表达为一个的匹配关系矩阵,分析了事件日志中包含实例一对一、一对多、多对一和多对多匹配关系对应的匹配关系矩阵的特征;(2) 针对随机方法对于生成包含多对多实例匹配关系的初始种群存在的问题,提出了一种构建人工免疫进化算法初始种群的启发式方法;(3) 亲和度函数的设计是人工免疫进化算法的关键,HAIA 算法考虑两个实例级别的因素来评价个体——流程执行路径出现的频次、流程实例间的时间匹配关系,分别从“量”匹配和“时间”匹配两个维度来考察个体;(4) 通过设置免疫记忆库、引入模拟退火机制,保证新一代种群的多样性,解决免疫进化因单一基因浓度过高导致的早熟问题.

本文第 1 节对事件日志融合问题相关的概念进行形式化定义,采用匹配矩阵表达问题解.第 2 节对 HAIA 算法进行介绍.第 3 节通过实验对 HAIA 算法的融合质量和融合效率进行评估.第 4 节对分布式 HAIA 中的数据划分问题进行了探讨.最后总结全文,提出未来的研究方向.

1 相关概念

在流程挖掘领域中,Petri Net[14]常常用来定义一个流程.采用 Petri Net 来表达流程,需要确定每个活动的输入和输出库所(place),而这不是本文的关注点,本文主要关注流程中活动间的关系.因此,本文采用定义 1 来定义一个流程(这里采用类似于文献[6]中对因果矩阵的定义来表达流程.文献[6]中论证了这种表示与 Petri Net 在表达活动间依赖的关系时是等价的).

定义 1(**流程**). 流程 $P=(A,C,I,O,a_0,a_e)$,其中,

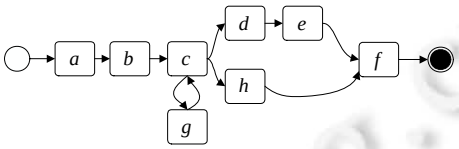
- A 是活动的有限集合;
- I 表示活动执行的前置活动,$I:A\rightarrow\mathcal{P}(\mathcal{P}(A))$($\mathcal{P}(A)$表示集合 A 的幂集);
- O 表示活动执行的后置活动,$O:A\rightarrow\mathcal{P}(\mathcal{P}(A))$;
- $a_0\in A$ 是开始活动,$I(a_0)=\{\varnothing\}$;
- $a_e\in A$ 是结束活动,$O(a_e)=\{\varnothing\}$;

- C 是一个强连通得图, $C=\{(a_1,a_2)\in A\times A \mid a_1\in\bigcup I(a_2)\}\cup\{(a_1,a_2)\in A\times A \mid a_2\in\bigcup O(a_1)\}$.

对于流程 $P'=(A',I',O',A'_0,A'_e)$, 如果 $A'\subseteq A$,而且 $I'\subseteq I,O'\subseteq O$,那么称 P'是 P 的子流程,记作 $P'\subseteq P$,称 P 是 P'的全局流程.

定义 2(流程路径). 给定流程 $P=(A,I,O,A_0,A_e)$,一个活动序列$\sigma\in A^*$称为一个流程路径,当且仅当存在 $a_1,\ldots,a_n\in A,\sigma=\langle a_1,\ldots,a_n\rangle$,对于所有 $i(1<i<n)$,I_i和 O_i是活动 a_i执行的前置活动和后置活动,有 $I_i\subseteq\mathcal{P}(\mathcal{P}(\{a_1,\ldots,a_{i-1}\}))$,而且 $O_i\subseteq\mathcal{P}(\mathcal{P}(\{a_{i-1},\ldots,a_n\}))$.

如图 1 所示,活动序列$\langle a,b,c,d,e,f\rangle$,$\langle a,b,c,h,f\rangle$,$\langle a,b,c,g,c,d,e,f\rangle$,$\langle a,b,c,g,c,g,c,d,e,f\rangle$就是 IT 事件管理流程可能的流程路径.



a:登记事件; *b*:分派事件; *c*:处置事件; *d*:事件解决
e:确认事件; *f*:关闭事件; *g*:挂起事件; *h*:取消事件

Fig.1 An IT incident management process

图 1 一个 IT 事件管理流程

定义 3(日志模式). 日志模式 S 是一个有限集合$\{D_1,\ldots,D_n\}$,其中,$D_i(i=1,\ldots,n)$称为属性域.

表 1 中 log1 的日志模式为{*CaseID*,*IncidentID*,*Activity*,*Timestamp*},它规定了描述日志 log1 的元数据,其中,*CaseID*,*Activity* 是属性域.

定义 4(事件). 给定一个流程 P 和一个日志模式 S,一个事件$(d_1,d_2,\ldots,d_n)\in D_1\times D_2\times\ldots\times D_n$是在流程 P 对应的活动集合 A 上关于日志模式 S 的一个解释,即日志模式的一个实例.

在流程挖掘领域中,一个事件代表一个流程活动的执行.表 1 中第 1 行记录(IN000001,INCID2016060102, INCIDENT_REGISTER,2016-06-06T23:08:04)就是一个事件,对应着 IT 事件管理流程中活动“事件登记”的一个执行.

事件是流程日志的最小构成单元.事件可由属性来描述,这些属性则由日志模式来规定.本文中,用记号 e 表示事件,用 $e.d$ 表示事件 e 的属性 d 的值.表 1 中,事件(IN000001,INCID2016060102,INCIDENT_REGISTER,2016-06-06T23:08:04)的属性 timestamp 值为 2016-06-06T23:08:04.

在本文研究过程中,假设所有事件都具有可靠的和可比较的时间戳,即 timestamp 属性.这样,流程中的事件按时间戳属性值排序,即$(\omega,\leqslant_{\text{timestamp}})$是偏序集.对于事件 $e_i,e_j\in\omega,1\leqslant i<j\leqslant|\omega|,e_i\leqslant_{\text{timestamp}}e_j$.

定义 5(流程实例). 给定一个流程 P 和日志模式 S,对于一个有限事件集合$\omega\subseteq D_1\times D_2\times\ldots\times D_n$,当且仅当满足以下条件时,称$\omega$为流程 P 的一个流程实例.

1) ω中的每个事件只出现 1 次,即对于事件 $e_i,e_j\in\omega,1\leqslant i<j\leqslant|\omega|,e_i\neq e_j$;
2) ω中的事件是有序的;
3) ω的所有事件构成的序列是流程 P 的某个流程路径的一个实际执行,即是某个流程路径的一个实例;

表 1 中流程实例 IN000001,IN000003 是流程路径$\langle a,b,c,d,e,f\rangle$的一个实例,IN000045 是流程路径$\langle a,b,c,g,c,d,e,f\rangle$的一个实例.

定义 6(事件日志). 一个事件日志 L 是流程实例的集合.对于任意两个流程实例$\omega_i,\omega_j\in L,\omega_i\cap\omega_j=\varnothing$.符号$|L|$表示日志 L 中包含的实例的数量.

事件日志是流程挖掘技术的输入,它具备的一个重要特征是流程实例被识别出来,而且实例之间没有共同的事件,即$\omega_i\cap\omega_j=\varnothing$.本文日志融合方法处理的对象就是事件日志.表 1 所示的日志就是一个事件日志,其中包含流程实例 IN000001,IN000002 和 IN000045,这 3 个流程实例中包含的事件各不相同.

如前所述,日志融合的关键是在给定两个事件日志上建立流程实例之间的匹配关系.所有实例间匹配关系构成的集合即是本文日志融合方法的解.本文用匹配矩阵来定义融合解.

定义 7(匹配矩阵). 一个匹配矩阵是一个五元组 $M=(P,P_1,P_2,R,\delta)$,其中,

- 流程 P_1,P_2 是 P 的子流程,即 $P_1\subseteq P,P_2\subseteq P$.
- $R=\{(\omega_x,\omega_y)|\omega_x\in L(P_1),\omega_y\in L(P_2)\}$是匹配关系集合,这里 $L(P_1),L(P_2)$分别是 P_1 和 P_2 的事件日志.(ω_x,ω_y)是定义在两个流程实例 ω_x,ω_y 上的二元关系.
- $\delta: L(P_1)\times L(P_2)\to\{0,1\}$是一个映射,$\delta(\omega_x,\omega_y)=\begin{cases}1, & \omega_x\cup\omega_y\in L(P)\\ 0, & \omega_x\cup\omega_y\notin L(P)\end{cases}$. $\omega_x\cup\omega_y$ 表示两个流程实例合并后构成的流程实例.如果 $\omega_x\cup\omega_y$ 属于全局流程实例,即 $\omega_x\cup\omega_y\in L(P)$,则二元关系 $(\omega^{L(P_1)},\omega^{L(P_2)})$ 为真(表示为 1);否则为假(表示为 0).

这样,可以用一个 $m\times n$ 的二值矩阵来表示一个事件日志融合的解,m,n 分别是两个事件日志中流程实例的数量.显然,实例间二元匹配关系是对称的.因此,在考察两个流程实例 ω_x 和 ω_y 的匹配关系时,只需考察(ω_x,ω_y),而不必再考察(ω_y,ω_x).考虑到流程实例是具有时间特性的,即两个匹配的实例 ω_x 和 ω_y 的开始发生在时间上有先后顺序,流程实例这种时间上的顺序关系是由事件日志对应流程的交互关系决定的.由于通过对业务的基本理解就可以判定两个事件日志对应的流程逻辑上的开始时间的先后关系,因此,本文假设(ω_x,ω_y)中 ω_x 所属事件日志对应的流程在逻辑上开始时间先于 ω_y 对应的流程.图 2 展示了表 1 和表 2 所示事件日志 log1 和 log2 实例间匹配的一个可匹配矩阵,其中,(IN000001,TK000005),(IN000003,TK000005),(IN000045,TK0000012)实例间的匹配关系为真,其他实例间的匹配关系为假.在考察两个日志的实例匹配关系时,IT 事件管理流程在业务逻辑上开始时间先于任务管理流程.

	TK 000005	TK000012
IN000001	1	0
IN000003	1	0
IN000045	0	1

Fig.2 A match matrix between log1 and log2

图 2 log1 和 log2 的一个匹配矩阵

这样,两个事件日志上建立流程实例之间的匹配关系问题转换为求解一个匹配矩阵,在这个矩阵中,匹配关系能够正确地表示两个事件日志中所有实例间匹配关系.那么,如何衡量可能解的匹配关系的正确性,是求解的关键.

正如前面所讨论的,两个事件日志的实例间并不仅仅是一对一的匹配关系,可能还存在一对多、多对一或多对多的匹配关系.本文采用匹配矩阵类型来描述这些不同类型的匹配关系.

定义 8(匹配矩阵类型). 匹配矩阵 $M=(P,P_1,P_2,R,\delta),(\omega_x,\omega_y)\in R$,且$\delta(\omega_x,\omega_y)=1$.

- 若$\forall\omega_z\in L(P_2),\omega_z\neq\omega_y$,总有$\delta(\omega_x,\omega_z)=0$,表明匹配矩阵只包含一对一的实例匹配关系,则称 M 为 0 型匹配矩阵;
- 若$\exists\omega_z\in L(P_2),\omega_z\neq\omega_y,\delta(\omega_x,\omega_z)=1$,表明匹配矩阵包含一对多的实例匹配关系,则称 M 为 I 型匹配矩阵;
- 若$\exists\omega_p\in L(P_1),\omega_p\neq\omega_x,\delta(\omega_p,\omega_y)=1$,表明匹配矩阵包含多对一的实例匹配关系,则称 M 为 II 型匹配矩阵;
- 若$\exists\omega_p\in L(P_1),\omega_z\in L(P_2),\omega_p\neq\omega_x,\omega_z\neq\omega_y,\delta(\omega_x,\omega_z)=1$ 且$\delta(\omega_p,\omega_y)=1$,表明匹配矩阵包含多对多的实例匹配关系,则称 M 为 III 型匹配矩阵.

0 型匹配矩阵表示两个事件日志的实例匹配是一对一的关系.它是一个置换矩阵,即 0 型匹配矩阵是一个每一行和每一列恰有 1 个 1 的 0-1 矩阵(这里考虑一个日志中的实例总可以在另一个日志中找相匹配的实例的情况).可以通过行和列的交换将匹配矩阵转换为一个对角矩阵.理想情况下,当 $L(P_1)$和 $L(P_2)$是一对一的实例匹配关系时,这两个事件日志包含的实例数量应是相同,即$|L(P_1)|=|L(P_2)|$.

I 型匹配矩阵是每一列恰有 1 个 1,而每一行至少有 1 个 1 的 0-1 矩阵.因此,当 $L(P_1)$和 $L(P_2)$是一对多的实例匹配关系时,$|L(P_1)|<|L(P_2)|$.

II 型匹配矩阵每一行恰有 1 个 1,而每一列至少有 1 个 1 的 0-1 矩阵,因此,$|L(P_1)|>|L(P_2)|$.

对于多对多匹配关系的 III 型匹配矩阵,每一行和每一列都至少有 1 个 1 的 0-1 矩阵.

应注意到:两个事件日志中包含的实例数量越大,无论是哪个类型的匹配矩阵,都越来越会呈现大规模稀疏 0-1 矩阵的特征.

2 基于人工免疫算法的日志融合

正如前面的论述,由于实际业务流程的灵活性、融合所需信息的缺失以及日志本身的“噪声”,仅仅从事件或流程实例的某几个属性来判断不同日志间的流程实例的匹配关系是不可靠的.构建匹配矩阵不仅仅是考虑某两个流程实例是否匹配,而是要考虑两个日志所有的实例是否匹配,因而它是一个全局优化的问题.对于这种搜索和优化的问题,进化算法(evolutionary algorithm),如遗传算法[15]或人工免疫进化算法[16],因其自组织、自适应、自学习的特性和鲁棒性高、易于并行的特点,是一个比较合适的选择.本文选取带有免疫记忆库的人工免疫系统作为日志融合方法的基础.

人工免疫系统是基于自然免疫系统方法的计算系统.免疫算法从体细胞理论和网络理论得到启发,实现了类似于生物免疫系统的抗原识别、细胞分化、记忆和自我调节的功能.可以将抗原、抗体、抗原和抗体之间的亲和性分别对应于优化问题的目标函数、优化解、解与目标函数的匹配程度.本文选用免疫系统的克隆选择理论作为基础.该原理反映了对抗原刺激响应的这一免疫基本特征,其基本思想是:只有那些能够识别抗原的细胞才能增殖,如果不能识别则剔除.这种克隆选择方法通过不断地迭代克隆选择、高频变异和受体编辑等操作,直至满足某一停止条件,最终获得最优解[15].

两个待融合日志的匹配矩阵的求解问题映射到人工免疫算法就是一个日志的实例集(抗原)和另一个日志的实例集(抗体)的匹配,得到亲和度最高的抗体-抗原匹配的过程.与遗传算法相比,人工免疫算法在记忆单元基础上运行,确保了快速收敛于全局最优解;遗传算法则是基于父代群体,不能保证概率收敛.免疫算法的评价标准是计算抗体和抗原的亲和度,通过促进或抑制抗体的产生,体现了免疫反应的自我调节功能,保证了个体的多样性;而遗传算法只是根据适应度选择父代个体,并没有对个体多样性进行调节.

图 3 描述了人工免疫算法进行匹配矩阵求解的框架.匹配矩阵的求解开始于一个初始种群,即候选的匹配矩阵(个体)集.对种群中的个体进行亲和度计算,并按照亲和度的高低对个体进行排序.按照一定的比例选择亲和度值高的个体进行克隆,产生克隆种群(临时种群).亲和度越高的个体,越有机会被克隆.克隆种群根据变异策略,经过变异操作获得变异种群(临时种群),经过多样性保持阶段处理产生新一代种群.新一代种群再经过选择、克隆、变异等操作,直至满足进化停止条件,比如出现满足亲和度要求的个体、进化代数满足了设定值或连续若干代不能产生亲和度更高的个体等,则选择亲和度值最高的个体作为问题解.

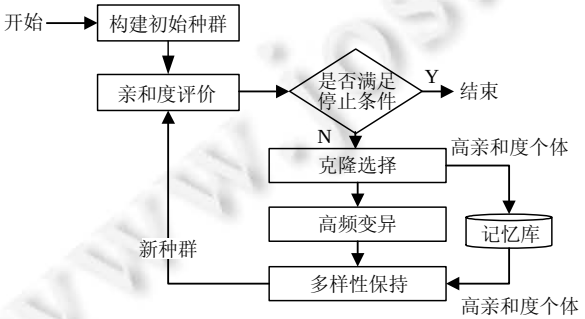


Fig.3 Framework for finding match matrix

图 3 匹配矩阵求解框架

图 3 所示的方法是一般性的框架,具体的算法还需要对种群初始化、亲和度评价、变异策略和多样性保持进行设计.

2.1 种群初始化

初始化种群可以采用完全随机的方法产生.给定两个事件日志,人工免疫算法的每一代种群中的个体,即候选匹配矩阵,都拥有相同的两个事件日志中的流程实例.利用随机方法建立匹配矩阵时,不同日志中的流程实例两两的匹配关系有 50%的概率为真或假,即$\delta(\omega_x,\omega_y)$有 50%的概率为 1 或 0.这样,建立初始种群可以包含的个体最大数量为 $2^{m\times n}$(m,n 分别是两个日志中流程实例的数量),这就是匹配矩阵问题的搜索空间.

完全随机的方法简单,产生初始种群的效率高,但盲目性强.使用完全随机方法生成只包含 0 型匹配矩阵个体的初始种群时,可根据每一行和每一列恰有 1 个 1 的特点来控制生成的个体满足 0 型匹配矩阵的要求.但对于初始种群中包含的个体是 I 型匹配矩阵、II 型匹配矩阵或 III 型匹配矩阵时,完全随机的方法会产生大量无效的"真"匹配关系,产生的矩阵也不具备匹配矩阵的稀疏特征,这样会大量增加随后的免疫进化算法的计算量.因此,本文考虑采用启发式方法来生成初始种群.启发式方法构建初始种群并不能改变最终种群进化的结果(如果种群进化足够多代数),但可以加快早期的进化进程.当将启发式方法用于人工免疫方法中进行初始种群的构建时,本文称这种方法为混合式人工免疫方法,即 HAIA.

- 启发式种群初始化

一般来说,两个有交互的业务流程执行时,它们之间会传递某些值.从流程实例角度看,这些值会从一个流程实例传递到另一个流程实例.因此可以认为,如果两个流程实例中事件属性值存在"相同",则表明这两个流程实例的匹配关系具有为"真"的可能性;"相同"的值越多,可能性就越大.例如,表 1 中 IT 事件管理实例 IN000045 的事件属性 IncidentID 值与任务管理实例 TK000012 的事件属性 RelatedIncidentID 值有相同,可以认为这两个实例的匹配关系有为"真"的可能.

基于这个观点,启发式方法构建个体的过程简单地说就是:对于匹配关系(ω_x,ω_y),统计ω_x 中事件属性值与ω_y 中事件属性值"相同"的数量,计算相同事件属性值数量占ω_x 和ω_y 所有事件属性值的比例.当这个比例达到一个设定的阈值时,则认为$\delta(\omega_x,\omega_y)=1$;否则,$\delta(\omega_x,\omega_y)=0$.两个日志中所有匹配关系都考察完毕后,就构建了一个个体.为了评价"相同事件属性值",本文定义以下度量.

定义 9(相同属性值比率). 给定两个流程实例$\omega_x\in L_1,\omega_y\in L_2$,相同属性值比例表示$\omega_x$ 和ω_y"相同"的事件属性的数量与ω_x 和ω_y 中所有事件属性值的比值,即

$$ratio_{attr}(\omega_x,\omega_y)=\frac{2\times|attr(\omega_x)\cap attr(\omega_y)|}{|attr(\omega_x)|+|attr(\omega_y)|} \tag{1}$$

这里,$attr(\omega)$表示实例ω中所有非空的事件属性值,$|attr(\omega)|$表示实例ω中非空事件属性值的数量,$|attr(\omega_x)\cap attr(\omega_y)|$表示两个实例$\omega_x,\omega_y$ 中非空属性值相同的数量.

公式(1)计算得到的是一个反映"相同"属性值在整个非空属性值的比例.一般地,如果两个实例属于同一个全局实例,则两个实例间某个或某几个事件属性会相同.通过考察两个实例间有没有"相同"属性值来判断是否为同一全局实例时,不能通过某个"相同"属性值(可能需要多个)就判断是否为同一全局实例.因此,仅仅通过计算 $ratio_{attr}(\omega_x,\omega_y)$不能判定两个实例是否为同一全局实例.这里,可以根据对业务的理解或领域知识,设定一个"相同"属性比率的阈值t或范围$[a,b]$,如果$ratio_{attr}(\omega_x,\omega_y)>t$或$ratio_{attr}(\omega_x,\omega_y)$ in $[a,b]$,则$\delta(\omega_x,\omega_y)=1$.启发式方法生成初始种群具体算法见算法 1.

算法 1. 启发式方法创建初始种群的个体.

INPUT: two event logs L_1, L_2 //待融合的日志
Threshold t //"相同"属性值比率阈值(0~1.0)
OUTPUT: MatchMatrix M //匹配矩阵(个体)
1: $T_1\leftarrow$the set of cases in L_1, $T_2\leftarrow$the set of cases in L_2 //两个实例集合
2: FOR all tuple $(\omega_x,\omega_y)\in T_1\times T_2$ DO

3: $\delta(\omega_x,\omega_y)$=0 //初始化匹配关系

4: FOR each tuple $(\omega_x,\omega_y)\in M$ DO

5: (a) select a random number k=*math.random*()+0.5

6: (b) IF $k\times t<ratio_{attr}(\omega_x,\omega_y)$ THEN

7: $\delta(\omega_x,\omega_y)$=1 //设置匹配关系为真

8: RETURN MatchMatrix M

在算法 1 中,使用了一个随机数 k(这里,k 取在[0.5,1.5)之间)来调节阈值,控制匹配关系(ω_x,ω_y)为真的概率.这样,可以使用算法 1 产生多个个体.

要注意到:两个实例间即使存在“相同属性值”,也不一定表明两个实例属于同一个全局流程实例.比如,两个实例中都具有“状态”属性,但不能因为这个属性值相同(比如“closed”)就判定两个实例的匹配关系为真.因此,启发式方法生成的个体并不能保证是正确解,需要采用人工免疫进化算法进一步求解最佳解.

2.2 亲和度函数

亲和度函数是人工免疫算法中最重要的设计.个体的亲和度由亲和度函数决定.对个体亲和度的评价贯穿整个进化算法,影响每一步的结果.在日志融合问题中,个体亲和度值高,表明该个体中匹配关系正确率高.理想情况下,最后得到的最佳个体,其所有的匹配关系都是正确的.因此,个体亲和度的度量就是度量匹配矩阵中匹配关系的正确程度.本文研究中考虑了两个主要因子——频次匹配度和时间匹配度,来度量个体的亲和度.公式(2)是亲和度函数:

$$f=\alpha_1\sum AOF+\alpha_2\sum OLT \tag{2}$$

公式(2)中,AOF 是指频次匹配度.不同日志中,两个实例ω_x,ω_y如果是匹配的,那么ω_x和ω_y两个实例对应的流程路径(见定义 2)在两个日志中出现的次数是相等的.对于实例之间存在一对多、多对一或多对多匹配关系,单个考察某一个实例对,其对应的流程路径出现次数可能不同,但从个体的全局视角看,它们的出现次数在统计意义上应相近.频次匹配度就是衡量个体中所有匹配关系间流程路径出现次数的相近程度.OLT 是指时间匹配度.有业务交互的两个流程执行时,在时间上是有交点.在这个时间交点处,一个流程为触发另外一个流程的执行.反映到日志中,就是可以匹配的两个流程实例的持续时间区间是有并行的部分的.因此,可通过度量所有匹配关系间流程实例的持续时间区间是否存在重叠部分来衡量个体的亲和度.公式(2)中,α_1 和α_2 是权重因子,可以针对具体问题,根据对业务的理解进行调整,以确定因子的重要性.

- 频次匹配度

定义 10(流程路径出现频次). 给定一个流程 P 和事件日志 L,Γ是流程 P 的所有流程路径的集合,π是建立在 L 上关于Γ的一个映射,$\pi:\Gamma\to$IN.对于$\forall\sigma\in\Gamma,\pi(\sigma)$是流程路径出现的次数.

比如,表 1 中流程实例 IN000001 和 IN000003 对应的图 1 中的流程路径都是$\langle a,b,c,d,e,f\rangle$,则表明$\langle a,b,c,d,e,f\rangle$在日志中出现了 2 次,这里记作$\langle a,b,c,d,e,f\rangle^2$.

对于每一个匹配关系(ω_x,ω_y),公式(3)通过计算ω_x,ω_y 对应的流程路径出现相同的频次占两个实例对应流程路径出现总频次的比例来衡量ω_x 和ω_y 的频次匹配程度.这个比例越高,表明ω_x,ω_y 属于同一全局实例的可能性越高.

$$AOF(\omega_x,\omega_y)=\frac{2\times\min\left(\pi(\sigma(\omega_x)),\pi(\sigma(\omega_y))\right)}{\pi(\sigma(\omega_x))+\pi(\sigma(\omega_y))} \tag{3}$$

这里,$\pi(\sigma(\omega))$计算每个实例ω对应的流程路径出现的频次.个体频次匹配度则可按照公式(4)来计算.

$$\sum AOF=\sum_{\delta(\omega_x,\omega_y)=1}AOF(\omega_x,\omega_y)\Big/\sum\delta(\omega_x,\omega_y) \tag{4}$$

其中,$\sum_{\delta(\omega_x,\omega_y)=1}AOF(\omega_x,\omega_y)$ 是计算所有匹配关系为真的两流程实例ω_x,ω_y 间的频次匹配度.$\sum\delta(\omega_x,\omega_y)$ 为匹配关系为真的匹配数量.

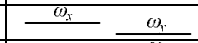
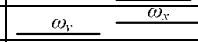
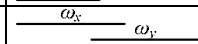
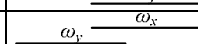
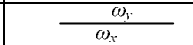
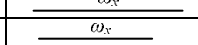
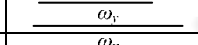
- 时间匹配度

定义 11(流程实例持续时间区间). L 是流程 P 的事件日志,$\omega\in L$ 是流程路径$\sigma=\langle a_1,\ldots,a_n\rangle$的流程实例.$\omega$持续时间区间 $T(\omega)$是一个时间区间$[t_0,t_1]$,其中,t_0 是ω的开始时间,t_1 是ω的结束时间.$t_0=e_1.timestamp$,$t_1=e_n.timestamp$,e_1,e_n 分别是ω中对应于活动 a_1,a_n 的事件.

在考察个体的时间匹配度时,本文作如下假设:待融合的事件日志涉及的两个流程有交互,一个流程总是先于另一个流程开始执行,并在本流程执行结束前触发另一个流程开始执行.要注意到,并不是所有有交互的流程的事件日志满足这一假设.比如,流程 P_2 被流程 P_1 触发,由于 P_2 的实际开始是由人工操作的,日志中可能出现 P_2 的流程实例的开始时间晚于 P_1 相对应的实例的结束时间.对于这种情况,我们认为,这两个实例在时间上是不匹配的.表 3 列出了两个实例在时间上的匹配关系,其中,ω_y 被 ω_x 触发.

Table 3 Temporal relations between two cases

表 3 两个实例间的时间关系

时间匹配关系	图例	是否匹配	说明
$\omega_x<\omega_y$		否	ω_x先于ω_y开始,$\omega_x.t_1<\omega_y.t_0$.$\omega_x$,$\omega_y$的时间区间没有重叠
$\omega_x>\omega_y$		否	ω_x不先于ω_y开始,ω_x,ω_y的时间区间没有重叠.不满足假设
$\omega_x\preceq\omega_y$		是	ω_x先于ω_y开始,$\omega_x.t_0<\omega_y.t_0$且$\omega_y.t_0<\omega_x.t_1<\omega_y.t_1$.$\omega_x$,$\omega_y$的时间区间部分重叠
$\omega_x\succeq\omega_y$		否	ω_x不先于ω_y开始,ω_x,ω_y的时间区间部分重叠.不满足假设
$\omega_x\ni\omega_y$		是	ω_x先于ω_y开始,$\omega_x.t_0<\omega_y.t_0$且$\omega_y.t_0<\omega_x.t_1$.$\omega_x$的时间区间完全包含$\omega_y$的时间区间
$\omega_x\in\omega_y$		否	ω_x不先于ω_y开始,ω_x的时间区间完全被包含于ω_y的时间区间.不满足假设
$\omega_x\|\|\omega_y$		否	ω_x不先于ω_y开始,ω_x,ω_y在时间上是并行关系.不满足假设

根据前面的假设,在亲和度计算中,只有 $\omega_x\preceq\omega_y$ 和 $\omega_x\ni\omega_y$ 这两种时间匹配关系对亲和度的提高有贡献.似乎只需统计这两种关系就可以反映个体的时间匹配度.但是存在以下可能性:考察个体中两个匹配关系(ω_x,ω_y)和(ω'_x,ω'_y),其中,ω_x 和 ω'_x 对应的流程路径相同;同样地,ω_y 和 ω'_y 对应的流程路径也相同.但日志中同时出现了 $\omega_x\preceq\omega_y$ 和 $\omega'_x\succeq\omega'_y$ 时间匹配关系.在这种情况下,$\omega_x\preceq\omega_y$ 提高了个体的亲和度,但 $\omega'_x\succeq\omega'_y$ 却降低了个体的亲和度.因此,在度量个体时间匹配度时不仅要统计提高亲和度的时间匹配关系$\omega_x\preceq\omega_y$ 和$\omega_x\ni\omega_y$,还要考虑其他时间匹配关系对个体亲和度的影响.

个体时间匹配度由公式(5)计算:

$$\sum OLT=\sum_{\omega_x\preceq\omega_y\cup\omega_x\ni\omega_y}\delta(\omega_x,\omega_y)\Big/\sum\delta(\omega_x,\omega_y) \tag{5}$$

其中,$\sum_{\omega_x\preceq\omega_y\cup\omega_x\ni\omega_y}\delta(\omega_x,\omega_y)$ 是指个体满足$\omega_x\preceq\omega_y$或$\omega_x\ni\omega_y$且匹配关系为真的配对数量,$\sum\delta(\omega_x,\omega_y)$ 为匹配关系为真的匹配数量.时间匹配度的值越大,表明个体的亲和度越高.

在计算个体亲和度时,本文考虑了频次匹配度和时间匹配度两个因素.这两个因素是在流程实例级别上考察个体的亲和度,是与具体业务和日志规则无关的.针对实际问题时,可以有更多因素来考虑.比如,如果在配置两个事件日志的实例标识规则时,让两个可匹配的实例标识相同,而且事件日志中记录了实例标识.这样,个体亲和度函数中就可以考虑诸如“实例标识相同”等因素.还可以从一些特定事件属性来考虑,比如两个流程的特定的活动执行人的匹配来识别两个实例是否匹配等.总之,判断一个匹配矩阵的正确性,仅考虑单一因素是难以得到正确结论的,需要综合考虑多个因素.

2.3 变异规则

变异操作是向种群中增加新的成分,在日志融合问题中,就是改变种群中两个日志现有的匹配关系.因此,变异操作针对种群中个体(匹配矩阵),随机选取一个匹配关系(ω_x,ω_y),选择如下操作之一执行.

- 增加.当$\delta(\omega_x,\omega_y)$=0 时,如果该匹配关系满足$\omega_x \preceq \omega_y$或$\omega_x \ni \omega_y$,则设置$\delta(\omega_x,\omega_y)$=1;
- 替换.当$\delta(\omega_x,\omega_y)$=1 时,如果匹配关系亲和度低于该个体的所有匹配关系亲和度的平均值,则删除该匹配关系(即设置$\delta(\omega_x,\omega_y)$=0);同时,寻找另一个匹配关系$(\omega_x,\omega_z)$,如果满足$\omega_x \preceq \omega_z$或$\omega_x \ni \omega_z$,则设置$\delta(\omega_x,\omega_z)$=1.
- 删除.当$\delta(\omega_x,\omega_y)$=1 时,如果该匹配关系不满足$\omega_x \preceq \omega_y$或$\omega_x \ni \omega_y$,则设置$\delta(\omega_x,\omega_y)$=0.

例如,表 3 描述的匹配矩阵中,考察关系(IN000001,TK000012),由于 IN000001<TK000012,不满足$\omega_x \preceq \omega_y$或$\omega_x \ni \omega_y$,则设置$\delta$(IN000001,TK000012)=0,得到包含新个体,如图 4 所示.

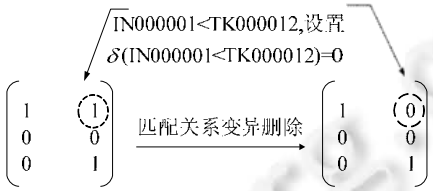


Fig.4 Mutation operator on match matrix

图 4 匹配矩阵的变异

每个个体的匹配关系变异数量与其亲和度值相关,亲和度值越高,变异的数量越少.本文采用公式(6)的计算方式来确定一个个体中匹配关系的变异数量:

$$num=\max(1,(1-e^{-\beta(best-curr)})\times m) \tag{6}$$

这里,*best* 是指种群中个体的亲和度最高值;*curr* 是指当前计算个体的亲和度值;*m* 是指当前计算个体中匹配数量,即 $m=\sum\delta(\omega_x,\omega_y)$.个体亲和度值越高,变异的匹配关系数量就越少.$\beta$是指减少的比率,是一个设定值.

2.4 多样性保持

在生物免疫系统中,如果抗体的浓度较高,那么该抗体受到抑制的概率就较高;如果抗体的浓度较低,那么该抗体受到促进的机会就较高.在免疫系统中,抗体浓度的高低直接影响到种群的多样性和质量.如果种群中基因相同或相似的抗体大量存在,就容易导致该类抗体的浓度较高,从而使得免疫算法的寻优搜索只出现在可行解区间的部分区域,这样将直接影响到算法的全局优化性能[17].

在日志融合的人工免疫进化方法中,在每一代种群中,亲和度高的个体就越有机会被克隆,而且个体亲和度越高,其匹配关系参与变异的数量也越少.经过若干代进化后,种群中基因相同或相似的解会大量存在,种群的基因相同或相似的抗体大量存在,导致人工免疫融合算法的早熟.为了保持每一代种群的多样性,HAIA 引入模拟退火机制,以便让进化过程中恶化种群的个体有机会参与到下一代种群的生成中,避免种群的基因模式呈现单一化趋势.

2.4.1 记忆库

生物免疫系统在免疫识别过程中,会将最优抗体以免疫记忆的方式保留对抗原的记忆.当免疫系统再次遇到相同或者结构相似的抗原时,在联想记忆的作用下,其应答速度将大为提高.基于这个机制,在本文免疫进化中设计了免疫记忆库,类似于免疫系统的记忆细胞,每一代群体个体亲和度评价后,将亲和度高的个体加入记忆库中.这样,记忆库中始终保持是进化过程中亲和度值最高的那些个体.当产生下一代种群时,记忆库中的个体会加入新种群中,这样会加速得到最优解.

- 初始化.记忆库的规模设置为 *R*.创建初始种群时,记忆库为空.初始种群个体进行亲和度评价后,选择亲和度值排在前 *R* 位的个体,加入到记忆库中.
- 更新.每一代种群产生后进行亲和度评价,用该代中亲和度高的个体去替换记忆库中亲和度比它低的个体.

- 复制.随机选取记忆库中 r%的个体替换新一代候选种群中个体亲和度值排位最低的 $R\times r$%个个体,参与新一代种群的产生.

2.4.2 模拟退火机制

模拟退火(simulated annealing,简称 SA)[18]源于对热力学中退火过程的模拟,在某一给定初温下,通过缓慢下降温度参数,使算法能够在多项式时间内给出一个近似最优解.SA 算法引入优化问题的求解,所得解依概率收敛到全局最优解.将模拟退火机制融入到克隆选择原理中,发挥模拟退火算法的优势,解决人工免疫克隆选择算法因基因模式的单一性带来的早熟问题.

当种群经历免疫进化(克隆、变异)后,生成新的种群.对这个新种群采用模拟退火机制来决定是否接受:对新种群 S'的个体进行亲和度评价,获得新种群中个体的最高亲和度值 $A_{\max}(S')$.比较 S'和上一代种群 S 中个体的最高亲和度值 $A_{\max}(S)$.根据模拟退火的 Metropolis 准则,如果新种群中最高亲和度值高于上一代种群的最高亲和度值,即$\Delta Z=A_{\max}(S')-A_{\max}(S)\geqslant 0$,表明新种群发生进化,则接受该新种群;如果$\Delta Z<0$,表明新种群发生了退化,则依据公式(7)计算的概率接受该新种群:

$$prob=\exp(\Delta Z/kT) \tag{7}$$

其中,k 为温度相关的常数因子,T 为退火温度.引入模拟退火机制后,下一代种群可以以一定的概率接受退化解,避免种群进化的早熟.

经过模拟退火机制选择的退火种群,经过记忆库的复制操作后,完成受体的编辑,最终形成新一代种群.算法 2 描述了模拟退火选择的过程.

算法 2. 模拟退火选择.

```
INPUT: G(i), P(i), B(i)   //G(i)当前种群,P(i)当前种群最佳个体,B(i)当前记忆库中亲和度值最低个体
OUTPUT: G(i+1)   //下一代种群
1:  计算 G(i)中所有个体亲和度;   //公式(2)
2:  update 记忆库 Bank[];
3:  T←affinity(P(i))−affinity(B(i));   //初始温度
4:  T_min←0.0;   //停止温度
5:  WHILE G(i) DO
6:      IF stop condition is true THEN
7:          RETURN G(i);
8:      G′(i+1);   //克隆、变异生成变异种群
9:      P′(i+1);   //变异种群 G′(i+1)的最佳个体
10:         IF (T>T_min) DO
11:             ΔZ←affinity(P′(i+1))−affinity(P(i));
12:             IF (ΔZ≥0) DO
13:                 G(i+1)←G′(i+1);   //接受新种群
14:             ELSE
15:                 IF (exp(ΔZ/kT)>random(0,1)) DO
16:                     G(i+1)←G′(i+1);   //接受新种群
17:                     T←r×T;   //退火降温
18:                 ELSE
19:                     G(i+1)←G(i);   //否则,拒绝新种群
20:         记忆库复制部分个体到 G(i+1)
21:         RETURN G(i+1);
```

2.5 算法复杂度分析

正如图 3 中的 HAIA 总体框架所示,整个算法的核心部分包括初始化种群、亲和度评价、进化计算、多样性保持.

- 时间复杂度

初始化种群就是采用启发式算法构建匹配矩阵集合的过程.假设两个待融合的事件日志 L_1 和 L_2,$|L_1|=n_1$,$|L_2|=n_2$,每一个实例中的平均事件数目为 m,每一个事件所带有的属性数目平均为 r,$m,r>0$.这里,每一个个体需要对 $n_1\times n_2$ 个匹配关系进行考察,针对每一个匹配关系,需要比较两个实例的所有事件的属性,找出“相同事件属性值”的数量.因此,建立每一个个体的时间代价是 $O(n_1n_2m^2r^2)$.如果设定种群规模 g,则启发式算法初始化种群的代价 T_1 为 $O(gn_1n_2m^2r^2)$.针对具体的融合问题,由于平均事件数量 m、事件属性数量 r、初始种群规模 g 为常数,因此,初始化种群时间代价为 $O(n_1n_2)$,初始化种群时间代价随着两个日志的流程实例规模增大而增大.

每当新种群产生时,都需要进行个体亲和度评价.亲和度评价中,每个个体亲和度计算包括频次匹配度计算和时间匹配度计算.日志中实例对应流程路径出现频次的计算,是将实例按照对应的流程路径进行分类统计,其时间代价最坏情况下为 $O(n^2)$,这里,$n^2=n_1^2+n_2^2$;最好情况下为 $O(n)$,这里,$n=\max\{n_1,n_2\}$.在整个 HAIA 算法中,流程路径频次的计算只需计算 1 次.同样地,流程实例持续时间区间的计算也是只需在免疫进化计算前计算一次,其计算代价与实例数量成正比,为 $O(n)$,这里,$n=n_1+n_2$.每个个体频次匹配度计算,是比较匹配关系为“真”的两个实例对应的流程路径出现频次.如果个体中“真”匹配关系数量为 m, $m=\sum\delta(\omega_x,\omega_y)$, $\max\{n_1,n_2\}\leqslant m\leqslant n_1\times n_2$,则个体间频次匹配度计算时间代价为 $O(m)$,即最坏情况为 $O(n^2)$,最好为 $O(n)$.实例间时间匹配关系的计算即是比较个体中所有匹配关系为“真”的两个实例对应的时间区间是否满足约束条件,这同样与个体中匹配关系为“真”的数量有关,即最坏情况为 $O(n^2)$,最好为 $O(n)$.个体亲和度的排序与具体的排序算法有关,常见的排序算法计算时间代价不超过 $O(n^2)$.综上所述,个体亲和度计算最坏情况下总代价为 $O(n^2)$.因此,在一代种群中,亲和度计算最坏情况下总开销为 $O(gn^2)$,最好可以做到 $O(gn)$.一般种群规模 g 定为常数,因此,种群亲和度计算时间开销最坏情况为 $O(n^2)$,最好情况可为 $O(n)$.

进化计算主要是对个体中的匹配关系做替换、增加、删除的变异操作以及多样性保持计算.变异操作使个体亲和度值增加,个体渐趋成熟.每一代进化中每个个体至少做 1 次变异操作,最理想情况下,每一代的变异计算代价为 $O(g)$,最坏为 $O(gm)$, $m=\sum\delta(\omega_x,\omega_y)$.多样性保持主要通过模拟退火机制来接受恶化解,记忆库更新和退火降温都是一个 n 次数的比较操作的代价 $O(n)$.

假设需要进化 s 代,因此,进化变异计算代价最好为 $O(gs)$,最坏为 $O(gsm)$,其中,种群规模 g 一般为常数.每一代变异种群均需要进行个体亲和度评价.

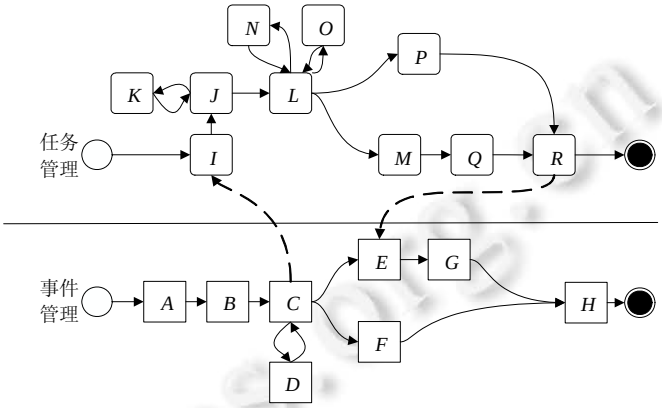
- 空间复杂度

HAIA 的输入为两个事件日志,输出为一个匹配矩阵.其中,输入的空间复杂度与事件日志的规模,即日志包含的实例数量相关,为 $O(n)$,$n=n_1+n_2$ 为两个日志的实例数量之和;输出是一个 $n_1\times n_2$ 的二值矩阵,其空间复杂度为 $O(n_1n_2)$.但正如前述分析,随着日志规模的增大,匹配矩阵逐渐具备大规模二值稀疏矩阵的特征,因此,匹配矩阵可以采用稀疏矩阵的压缩存储方法(比如三元组顺序表)减少存储空间,这样,其空间复杂度为 $O(m)$, $m=\sum\delta(\omega_x,\omega_y)$.在 HAIA 进化计算过程中,主要的空间开销为每一代种群的内存开销,每一代种群均由个体(匹配矩阵)构成,因而其空间开销为 $O(gn_1n_2)$,其中,g 为种群规模.采用稀疏矩阵的压缩存储方法,其空间开销可以降为 $O(gm)$.

3 实 验

本实验是基于某企业 IT 运维 SaaS 应用平台支撑的 IT 事件管理流程和运维任务管理流程.实验中对这两个流程进行了简化(如图 5 所示),并在受控环境下分别模拟生成这两个简化流程的事件日志.图 5 描述 IT 系统运行过程中事件处理全过程,包括两个子流程:事件管理流程和任务管理流程.这两个子流程分别由事件管理系

统和任务管理系统支撑.事件管理流程管理从 IT 事件的登记、事件分类定级、分派、处置到事件关闭的整个生命周期过程,而任务管理流程负责运维任务的分派、执行和关闭的全过程.其中,IT 事件分派(活动节点 *C*)后,触发任务管理产生一个新的运维任务.由于 IT 事件往往对客户的业务会产生影响,因此对事件处置的及时性和质量有比较高的要求,因此,运维任务执行完成后,会向事件管理流程反馈信息,这时,事件管理流程则转向事件确认阶段,客户需要对事件处置进行确认,IT 事件才能关闭.显然,事件管理流程先于任务管理流程开始,并且结束时间应晚于任务管理流程,因此,这两个流程在时间上满足 $\omega_x \ni \omega_y$ 关系.



A:登记事件 *B*:分派事件 *C*:事件处置 *D*:挂起事件 *E*:事件解决 *F*:取消事件
G:确认事件 *H*:关闭事件 *I*:创建任务 *J*:分配任务 *K*:挂起任务 *L*:执行任务
M:任务完成 *N*:转派任务 *O*:请求协作 *P*:取消任务 *Q*:评价任务 *R*:关闭任务

Fig.5 IT incident handling process

图 5 IT 事件处置流程

实验分别进行了对比实验和多对多(包含一对多和多对一)匹配关系的实验.实验从融合效率、融合质量两方面来衡量相关算法,其中,融合效率是指完成两个事件日志融合的算法执行时间,融合质量用融合成功率来评价.融合成功率(merging success rate,简称 MSR)是指两个日志融合后正确的实例配对数量(m_s)占实际实例配对数量(m_t)比例,如公式(8)所示.

$$MSR = \sum m_s / \sum m_t \tag{8}$$

本实验的所有测试均是在单机下运行的实验结果,单机测试所用的基本配置双核 CPU,主频为 2.20GHZ,内存为 6G.每一组实验均采用相同测试数据和参数设置,独立执行 3 次,取 3 次运行结果的平均值作为最终实验结果.实验中,所有测试数据均已结构化为事件日志,并采用 XES 格式(http://www.xes-standard.org/)进行描述.

3.1 对比实验

目前,特别针对流程挖掘的事件日志数据融合技术成果还相对比较少,主要研究成果有基于遗传算法的日志融合方法、基于人工免疫算法的日志融合方法和基于规则配置的日志融合方法.基于遗传算法的日志融合方法作为较早实现日志融合的方法,效果较差.这种方法在 ProM 中已被剔除,被基于免疫算法的日志融合方法所替代.基于规则配置的日志融合方法需要手动配置规则(需要先验性知识才可以准确配置规则),不同配置对结果影响很大,同时对生成的数据也有要求,例如,某个子流程的实例中的某个事件的某个属性固定写入另一个子流程的实例的某个事件中,类似于这种数据关系就可以配置融合规则.基于以上分析,本文只选择与基于免疫算法的日志融合方法(即 AIA)进行对比实验.

对比实验是对 HAIA 和 Claes 的算法(下面简称 AIA)进行比较,主要比较两种算法在免疫进化阶段的融合效率和融合成功率.两种算法均采用免疫克隆选择原理作为基础,除了在亲和度函数、变异策略和多样性保持方面,两种算法有较大不同以外,AIA 方法只支持一对一的匹配关系,采用的是随机初始种群.因此,对比实验做

了两点限制:(1) HAIA 和 AIA 都采用随机初始种群;(2) 测试用数据只包含一对一的实例匹配 关系.

对比实验采用 3 组测试数据,见表 4.每组数据包含的两个日志含有相同的实例数量,分别为 1 024, 2 048 和 4 096,均包含少量噪声(错误流程路径对应的实例,随机产生).

Table 4 Test data for the comparative experiment between HAIA and AIA

表 4 HAIA 与 AIA 对比实验用数据

测试组	事件日志	实例数	事件数	事件属性数
G1	log1 vs. log2	1 024 vs. 1 024	5 790 vs. 7 942	5
G2	Log3 vs. log4	2 048 vs. 2 048	11 598 vs. 15 847	5
G3	Log5 vs. log6	4 096 vs. 4 096	23 401 vs. 31 903	5

实验中,随机种群的规模为 100,AIA 算法停止条件设置为 10 000 代;而 HAIA 算法停止条件设置为连续 50 代种群中个体亲和度最大值不再提高或者免疫进化最大 10 000 代.实验结果见表 5.

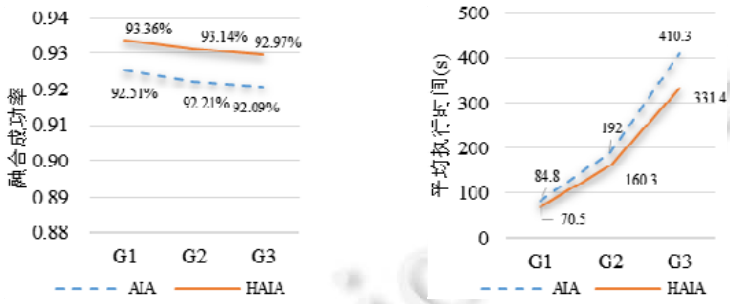
Table 5 Test results of the comparative experiment between HAIA and AIA

表 5 HAIA 与 AIA 对比实验结果

实验数据	算法	平均成功匹配数	融合成功率(%)	进化代数	平均执行时间(s)
G1	AIA	949	92.51	10 000	84.8
	HAIA	957	93.36	6 367	70.5
G2	AIA	1 883	92.21	10 000	192.0
	HAIA	1 907	93.14	7 362	160.3
G3	AIA	3 735	92.09	10 000	410.3
	HAIA	3 785	92.97	7 299	331.4

从表 5 的实验结果可以发现:两个算法在事件日志中仅存在一对一匹配关系的情况下,融合成功率均在 90%以上,HAIA 的运算结果稍好于 AIA.3 组实验中,AIA 所用的进化代数均达到了停止条件设定的 10 000 代,而 HAIA 算法停止时所用的进化代数明显小于 AIA,HAIA 算法所用的平均执行时间比 AIA 算方法降低了 21%.因此,在保证融合成功率的情况下,HAIA 算法能够比 AIA 算法更快地收敛到最优解.

这里应注意到:随着事件日志规模的不断扩大,不论是 AIA 还是 HAIA 算法,它们的融合质量和融合效率都存在下降趋势,如图 6 所示.随着日志规模的扩大,实例间的匹配关系在数量上增大,匹配关系正确性判断的复杂性随之增加,这是导致两个算法融合质量和效率下降的原因.



(a) 融合质量趋势 (b) 融合效率趋势

Fig.6 Trends of merging quality and efficiency of HAIA and AIA

图 6 HAIA 与 AIA 融合质量和融合效率趋势

3.2 多对多关系实验

多对多关系的实验是指测试用事件日志中除了包含 0 型关系外,还包含 I 型、II 型或 III 型的匹配关系,对 HAIA 的融合质量和融合效率进行评价,分析启发式方法生成初始种群对免疫进化的影响.

实验用数据见表 6.

Table 6 Test data for HAIA experiments on many to many relation between cases

表 6 HAIA 多对多关系实验数据

测试类	匹配关系	测试组	实例数	事件数	事件属性数
T1	0型、I型	G4	869 vs. 1 332	4 083 vs. 10 169	7
		G5	2 702 vs. 4 127	12 764 vs. 32 161	
		G6	5 365 vs. 8 238	24 702 vs. 61 713	
T2	0型、II型	G7	1 115 vs. 595	4 519 vs. 6 528	7
		G8	2 218 vs. 1 181	12 963 vs. 9 167	
		G9	4 479 vs. 2 349	26 492 vs. 17 890	
T3	0型、I型、II型、III型	G10	1 243 vs. 1 108	7 530 vs. 6 712	7
		G11	3 626 vs. 3 418	18 749 vs. 17 783	
		G12	7 061 vs. 6 813	43 169 vs. 38 711	

表 6 中,测试数据分为 3 大类:T1,T2,T3,分别包含了 I 型关系、II 型和 III 型关系的事件日志,每类数据包含 3 组数据,均包含少量随机产生的噪声.3 组数据的规模基本上成倍数增长.实验中,初始种群分别使用随机方法和启发式方法生成,规模均为 100,免疫进化的停止条件设置为连续 50 代种群中个体亲和度最大值不再提高.

采用随机方法生成初始种群的实验,主要目的是验证启发式方法对免疫进化的影响.按照包含 I 型、II 型和 III 型关系的各匹配矩阵的特点,个体中每个匹配关系均采用完全随机方式来确定.采用随机方法生成初始种群的实验结果见表 7.采用启发式方法生成初始种群的实验结果见表 8.

Table 7 Test results of HAIA experiments on many to many relation between cases (randomly generated initial population)

表 7 HAIA 多对多关系实验结果(随机生成初始种群)

测试类别	测试组	种群初始化平均执行时间	种群进化平均执行时间(s)	融合成功率(%)
T1	G4	4.8	81.5	91.98
	G5	15.5	595.6	91.74
	G6	32.0	2 223.6	91.56
T2	G7	3.9	60.6	92.04
	G8	9.7	194.6	91.88
	G9	18.0	512.4	91.73
T3	G10	5.3	116.3	91.25
	G11	18.8	625.3	91.11
	G12	35.7	2 414.3	91.07

Table 8 Test results of HAIA experiments on many to many relation between cases (heuristic to generate initial population)

表 8 HAIA 多对多关系实验结果(启发式生成初始种群)

测试类别	测试组	种群初始化平均执行时间	种群进化平均执行时间(s)	融合成功率(%)
T1	G4	68.8	8.5	94.08
	G5	400.6	73.4	93.91
	G6	1 603.6	390.0	93.80
T2	G7	22.8	4.6	94.12
	G8	92.5	22.6	93.81
	G9	346.5	61.4	93.78
T3	G10	77.3	16.1	93.88
	G11	415.8	152.3	93.91
	G12	1 654.1	491.5	93.67

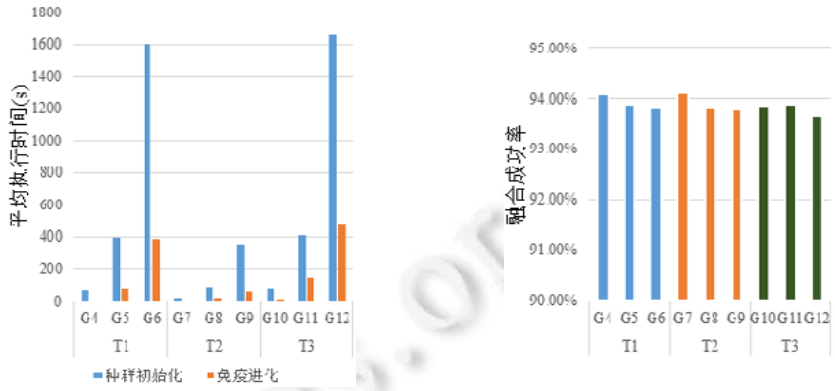
从表 7 和表 8 的实验结果对比来看,采用随机方法生成初始种群平均执行时间远小于采用启发式方法生成初始种群.这是由于实例间相同属性值比率 $ratio_{attr}(\omega_x,\omega_y)$的计算是要对$\omega_x$ 和ω_y 的所有事件的所有属性进行比较,因而 $ratio_{attr}(\omega_x,\omega_y)$的计算是非常耗时的.

然而,随机方法生成初始种群的免疫进化消耗的平均执行时间远大于采用启发式方法生成初始种群的进化平均执行时间.综合来看,采用启发式方法生成初始种群的免疫算法总耗时要小于采用随机方法生成初始种群免疫算法.这说明采用启发式方法产生初始种群,能够有效地提高免疫进化的效率,加快进化收敛.同时,从表 7 和表 8 也可以看到,采用启发式方法生成初始种群的融合成功率比采用随机方法生成初始种群的融合成功率

略高.

从表 8 中可知:在日志中包含 I 型关系、II 型和 III 型关系的 3 类测试中,HAIA 融合成功率都在 93%以上.在整个算法中,种群初始化的时间开销要远远高于进化阶段的时间开销.这个实验结果和第 2.5 节算法复杂度的分析结论一致,启发式方法生成的随机种群的复杂度比种群进化计算的复杂度要高.

同样需要注意的是:无论事件日志中包含 I 型关系、II 型还是 III 型匹配关系,随着事件日志规模的不断扩大,HAIA 的种群初始化和免疫进化的效率都呈恶化趋势;HAIA 融合成功率相对稳定,但仍呈现逐渐下降趋势,如图 7 所示.



(a) HAIA 多对多关系融合效率 (b) HAIA 多对多关系融合成功率

Fig.7 Trends of merging quality and efficiency of HAIA

图 7 HAIA 融合质量和融合效率趋势

相对于 0 型匹配关系的融合问题,I 型、II 型和 III 型匹配关系的实例间关系比较复杂.因此,当事件日志规模增大时,日志中包含的实例间关系复杂性均增加,导致融合效率呈下降趋势.图 6(a)中,在日志规模增长大体相当的情况下,包含多对多关系日志的融合效率恶化更迅速,这也表明日志规模增大,多对多关系融合问题的复杂性比其他类型的匹配关系更加严重.

上述实验表明:(1) HAIA 能够较好地实现包含多对多匹配关系的日志融合;(2) 在保证融合成率的情况下,HAIA 算法比 AIA 算法能够更快地收敛(针对一对一匹配关系);(3) 启发式方法生成初始种群,能够提高 HAIA 的免疫进化的效率;(4) 随着日志规模的增大,日志中匹配关系的复杂度升高,HAIA 融合性能趋于恶化.

4 讨 论

第 2.5 节的 HAIA 算法复杂度分析说明,HAIA 的性能随着事件日志的规模增大而趋于下降.第 3 节的实验结果进一步表明:两个待融合的日志中包含的匹配关系类型越复杂,HAIA 的性能下降得越快.如何面对大规模日志数据,有效地提高日志融合效率,是日志融合技术得到实际应用需要解决的重要问题.

如今,诸如多核计算、集群计算、网格计算、云计算等分布式计算系统被广泛应用于提高计算性能和可扩展性,分布式聚类[19,20]、分布式关联规则挖掘[21]等分布式数据挖掘技术也被用于提高数据挖掘性能.分布式流程挖掘技术的研究也见于文献[3,22,23].同样地,日志数据融合也可以采用分布式技术来提高融合性能.本节将对分布式流程日志数据融合技术进行探讨.在此,本文不关注分布式融合的实现细节,而是重点讨论分布式 HAIA 中的数据划分问题.

4.1 种群初始化

HAIA 的启发式方法构建初始种群的个体的核心是计算日志间两两实例的相同属性值比率 $ratio_{attr}(\omega_x,\omega_y)$.从算法分析和实验结果看,$ratio_{attr}(\omega_x,\omega_y)$的计算是非常耗时的.从 $ratio_{attr}(\omega_x,\omega_y)$的计算方式看,$ratio_{attr}(\omega_x,\omega_y)$的

计算仅与ω_x和ω_y两个实例中的事件相关,与其他实例无关.因而可以考虑将日志划分为多个不同的实例子集,每个子集分配给不同的计算节点进行 $ratio_{attr}(\omega_x,\omega_y)$的计算,通过这种分布式计算方式提高初始化的效率.

日志中的每个实例需要与另一个日志的所有实例进行 $ratio_{attr}$ 的计算.假设有 n 个计算节点,一种方案是考虑将规模较大的日志划分为 n 个不同子集,分别分配到 n 个计算节点上,而将规模较小的日志(实例数量较少的日志)的所有实例在每个节点上复制,计算的结果可以按照矩阵的行或列的形式进行融合.日志划分为子集时,可以将规模(实例中包含事件数量)相当的实例均匀划分到各子集,使各子集的 $ratio_{attr}$ 计算量尽可能地均衡.如图 8 所示,日志 logA 根据各实例的规模被划分为两个实例子集{case1,case4,case5},{case2,case3,case6}.这两个子集分别分配给两个计算节点,同时,将logB所有实例复制到这两个计算节点,分别在这两个节点上并行计算得到两个子匹配矩阵,最后,将这两个子矩阵合并,得到初始种群中的一个匹配矩阵.这种方案在子匹配矩阵计算过程中,节点间不需要进行数据交换,因而分布式计算中的通信开销小.但是对于"大数据"级的日志,将大规模日志的所有实例复制到每个节点并不适合.

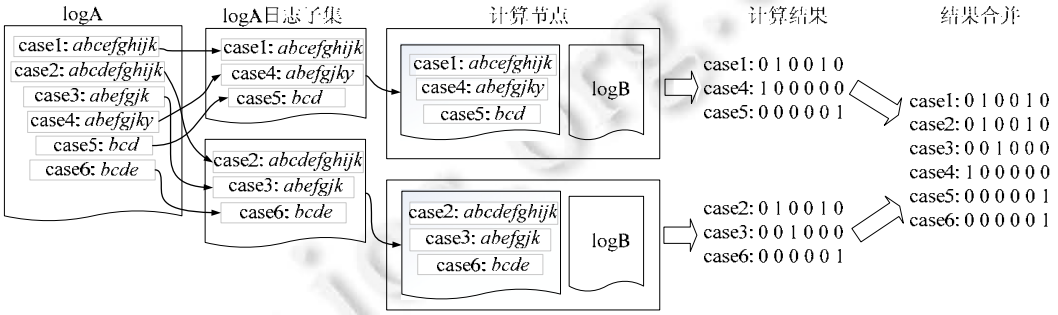


Fig.8 Partitioning the log according to the cases size

图 8 根据实例规模的日志划分

针对"大数据"级的日志,可以将两个待融合日志均按照实例规模划分为n个子实例集合,分别分配给n个计算节点.在分布式计算过程中,规模较小的日志的子实例集合在各个节点间进行交换,直至两个日志所有实例间均进行了计算.如图 9 所示,日志 logA 根据各实例的规模被划分为两个实例子集{case1,case4,case5},{case2,case3,case6},logB 则划分为两个实例子集{case1′,case2′,case4′}和{case3′,case5′,case6′}.{case1,case4,case5}与{case1′,case2′,case4′}完成匹配关系计算后,节点2将{case3′,case5′,case6′}数据交换到节点1,{case1,case4,case5}再与{case3′,case5′,case6′}进行匹配关系计算.

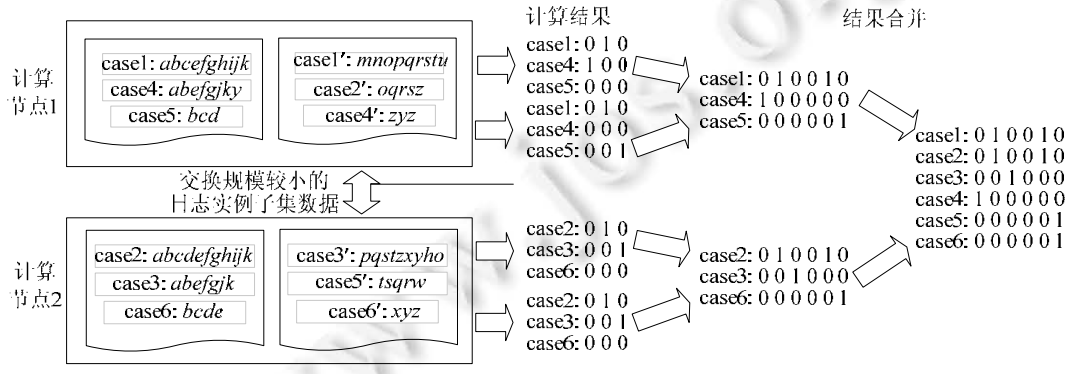


Fig.9 Partitioning the log as "big data"

图 9 针对"大数据"级日志的划分

同样地,完成{case2,case3,case6}与{case3′,case5′,case6′},{case1′,case2′,case4′}的计算.计算结果分两步进行合并,logA 的实例子集{case1,case4,case5}与 logB 的{case1′,case2′,case4′},{case3′,case5′,case6′}计算得到两个匹配关系子矩阵,这两个匹配关系子矩阵首先在节点 1 合并.同样地,计算{case2,case3,case6}与 logB 在节点 2 上合并后的匹配关系子矩阵.最后,将节点 1 和节点 2 上的匹配关系子矩阵合并,得到初始种群中的一个匹配矩阵.这种方案节点间需要进行数据交换,网络通信开销大.

4.2 分布式免疫进化

HAIA 免疫进化的核心是每一代种群中个体亲和度的计算和排序.个体亲和度的计算与种群中其他个体没有关系,因此,种群中个体亲和度计算适于采用分布式方式实现并行.可以将种群划分为 n 个子种群,分别在 n 个节点独立进行免疫进化,节点间通过各子种群免疫进化结果再进行亲和度比较,选出亲和度最高的个体作为最终解.需要进一步考虑的问题是:1) 如何划分子种群;2) 子种群间如何交换其“最佳”个体.

- 种群划分

子种群的划分需要考虑的一个重要因素是如何避免子种群中个体的亲和度均偏“高”或偏“低”,导致子种群在进化中容易陷入早熟.因此,需要亲和度相近的个体均衡分布在各子种群中.在两个匹配矩阵间,相同的匹配关系数量越多,表明这两个匹配矩阵的亲和度值越接近,这里称作“亲近度”.假设计算节点为 n,根据个体间亲近度将种群中的个体划分为若干个聚类,再将各个聚类中的个体均匀分配为 n 个子种群.这样,可以保证子种群的个体亲和度是均衡的.

匹配矩阵是 0-1 二值矩阵,可以考虑两个矩阵对应项之间进行异或计算,即 $a_{ij}\otimes b_{ij}$,异或计算得到的矩阵中值为 1 的项越少,表明这两个矩阵的相同的匹配关系数量越多,它们的近亲度越高近.如图 10 所示,匹配矩阵 A 分别与 B,C 进行对应项异或计算,得到矩阵 D 和 E.其中,D 中为 1 的项数量为 1,E 中为 4,则认为个体 A 和个体 B 的亲近度高于 A 和 C 的亲近度.可以发现,异或计算得到的矩阵所有项值的和,即 $\sum_M a_{ij}$, 就是该矩阵中为 1 的项的数量,因此,可以通过比较 D 和 E 的所有项值的和来判定 A 与 B 的亲近度和 A 和 C 的亲近度的高低.

$M\otimes N$

$$B=\begin{bmatrix}0&1&1\\0&0&0\\1&0&0\end{bmatrix}\quad C=\begin{bmatrix}1&0&1\\0&1&0\\1&0&0\end{bmatrix}$$

$$A=\begin{bmatrix}1&0&1\\0&1&0\\1&0&1\end{bmatrix}\quad D=\begin{bmatrix}0&0&0\\0&0&0\\0&0&1\end{bmatrix}\quad E=\begin{bmatrix}1&1&0\\0&1&0\\0&0&1\end{bmatrix}$$

$\sum_D a_{ij}=1<\sum_E a_{ij}=4$,因此,$A$的亲和度与$B$更接近

Fig.10 XOR between match matrixces

图 10 匹配矩阵间的异或计算

- 个体迁移

子种群间的“最佳”个体的交换方式,即分布式 HAIA 的个体迁移,需要考虑迁移个体的类型、迁移个体的规模、接受迁移个体方式、迁移间隔.

分布式 HAIA 中,子种群间迁移的个体既可以是各子种群亲和度排名高的个体,也可以考虑选取随机的个体做迁移.免疫系统的克隆选择理论的基本思想是,只有那些能够识别抗原的细胞才能增殖.因此,选择亲和度“最佳”的个体进行迁移是较好的策略.

在分布式进化算法中,子种群的个体迁移往往采用子种群间的广播方式,即按照设定的迁移规模向所有其他子种群交换个体.而在 HAIA 中设置了免疫记忆库机制,用于存储的是免疫进化过程中亲和度排在前列的个体,这种保证多样性的目的与个体迁移的目的相同.因而在分布式 HAIA 中,除了子种群间的广播方式以外,可以考虑采用以记忆库为中介的个体迁移方式,所有子种群将本种群中排名前 n 位的个体提交给记忆库,按照记忆

库的“更新”策略,子种群间进行迁移个体“竞争”,最后,将亲和度在全局排名前列的个体存储在记忆库中.各子种群则按照记忆库“复制”策略从记忆库获取“最佳”个体来接受迁移的个体,用于产生下一代子种群.关于迁移间隔,在 HAIA 的进化中,记忆库的更新及复制在每一代都发生,考虑个体的迁移在子种群每一代都进行.有研究表明:在分布式进化计算中,小的迁移间隔可能发生某些子种群主宰其他子种群的情况而导致全局多样性的降低,大的迁移间隔会降低进化收敛的速度[24].在分布式 HAIA 中,记忆库最终存储的是各子群通过“竞争”而保留下来的亲和度排名“最佳”的个体集合,加上初始种群均衡划分策略和模拟退火机制的采用,因而可以极大地降低出现某些子种群主宰其他子种群的机会.

5 结 论

实际业务中的流程灵活性、融合所需信息的缺失以及日志本身的“噪声”,给流程挖掘日志融合带来了挑战.本文对事件日志融合问题进行了形式化定义,指出该问题是一个搜索优化问题,并提出了一种基于混合人工免疫算法的日志融合方法 HAIA.这种方法以人工免疫系统的克隆选择理论为基础,通过免疫进化获得“最佳”解.在免疫进化的每一代,使用两个实例级别的因素,流程执行路径出现的频次和流程实例间的时间匹配关系,分别从“量”匹配和“时间”匹配两个维度对进化过程中的个体(匹配矩阵)进行评价,通过克隆、变异操作选择保留亲和度高的个体,直至获得“最佳”个体.实验结果表明:(1) HAIA 支持包含复杂流程实例间匹配关系的日志融合;(2) 启发式方法生成初始种群,能够加快免疫进化的搜索性能;(3) 免疫记忆库和模拟退火机制的引入能够保持种群的多样性,减少陷入早熟陷进的机会.

针对大规模流程日志的融合性能趋于恶化的问题,本文还讨论了分布式日志融合中的数据划分问题:针对种群初始化,提出了以实例规模进行数据划分的方法;针对免疫进化,提出了以匹配矩阵间的亲近度为基础的聚类方法的子种群划分策略以及以免疫记忆库为媒介的子种群间个体迁移方法.但是对不断增大的日志规模来说,目前“离线”方式的日志融合方法的性能会受到日志数据的存储方案的影响,而且在时效性方面比较差.流式的日志融合方法是未来进一步研究的方向之一.

References:

[1] Beheshti SMR, Benatallah B, Sakr S, Grigori D, Motahari-Nezhad HR, Barukh MC, Gater A, Ryu SH. Process Analytics: Concepts and Techniques for Querying and Analyzing Process Data. Springer Int'l Publishing, 2016. [doi: 10.1007/978-3-319-25037-3]

[2] van der Aalst WMP. Process Mining: Discovery, Conformance and Enhancement of Business Processes. Berlin, Heidelberg: Springer-Verlag, 2011. [doi: 10.1007/978-3-642-19345-3]

[3] van der Aalst WMP. Process Mining: Data Science in Action. Berlin, Heidelberg: Springer-Verlag, 2016. [doi: 10.1007/978-3-662-49851-4]

[4] Zikopoulos P, Eaton C. Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data. McGraw-Hill Osborne Media, 2011.

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

[6] Medeiros AK, Weijters AJ, 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]

[7] van der Aalst WMP. Distributed process discovery and conformance checking. In: Proc. of the Int'l Conf. on Fundamental Approaches to Software Engineering. Springer-Verlag, 2012. 1–25. [doi: 10.1007/978-3-642-28872-2_1]

[8] van der Aalst WMP, Adriansyah A, Van Dongen B. Replaying history on process models for conformance checking and performance analysis. Wiley Interdisciplinary Reviews Data Mining & Knowledge Discovery, 2012,2(2):182–192. [doi: 10.1002/widm.1045]

[9] Claes J, Poels G. Merging computer log files for process mining: An artificial immune system technique. In: Proc. of the Business Process Management Workshops. Berlin, Heidelberg: Springer-Verlag, 2011. 99–110. [doi: 10.1007/978-3-642-28108-2_9]

[10] Pérez-Castillo R, Weber B, de Guzmán IG, Piattini M, Pinggera J. Assessing event correlation in non-process-aware information systems. Software & Systems Modeling, 2014,13(3):1117–1139. [doi: 10.1007/s10270-012-0285-5]

[11] Motahari-Nezhad HR, Saint-Paul R, Casati F, Benatallah B. Event correlation for process discovery from Web service interaction logs. The VLDB Journal, 2011,20(3):417−444. [doi: 10.1007/s00778-010-0203-9]

[12] Claes J, Poels G. Integrating computer log files for process mining: A genetic algorithm inspired technique. In: Proc. of the Advanced Information Systems Engineering Workshops. Berlin, Heidelberg: Springer-Verlag, 2011. 282−293. [doi: 10.1007/978-3-642-22056-2_30]

[13] Claes J, Poels G. Merging event logs for process mining: A rule based merging method and rule suggestion algorithm. Expert Systems with Applications, 2014,41(16):7291−7306. [doi: 10.1016/j.eswa.2014.06.012]

[14] Murata T. Petri nets: Properties, analysis and applications. Proc. of the IEEE, 1989,77(4):541−580. [doi: 10.1109/5.24143]

[15] Eiben AE, Smith JE. Introduction to Evolutionary Computing. Heidelberg: Springer-Verlag, 2003. [doi: 10.1007/978-3-662-44874-8]

[16] Burke EK, Kendall G. Search Methodologies-Introductory Tutorials in Optimization and Decision Support Techniques. 2nd ed., New York: Springer-Verlag, 2014. [doi: 10.1007/978-1-4614-6940-7]

[17] Shu WN. Artificial immune algorithm optimization and its key problems research [Ph.D. Thesis]. Wuhan: Wuhan University, 2013 (in Chinese with English abstract).

[18] Fu WY, Ling CD. Brownian motion based simulated annealing algorithm. Journal of Computer, 2014,6(37):1301−1308 (in Chinese with English abstract). [doi: 10.3724/SP.J.1016.2014.01301]

[19] Visalakshi NK, Thangavel K. Distributed data clustering: A comparative analysis. In: Proc. of the Foundations of Computational, Intelligence, Vol.6. Berlin, Heidelberg: Springer-Verlag, 2009. 371−397. [doi: 10.1007/978-3-642-01091-0_16]

[20] Singh D, Gosain A. A comparative analysis of distributed clustering algorithms: A survey. In: Proc. of the Int'l Symp. on Computational and Business Intelligence. IEEE, 2013. 165−169. [doi: 10.1109/ISCBI.2013.40]

[21] Sawant V, Shah K. A survey of distributed association rule mining algorithms. Journal of Emerging Trends in Computing and Information Sciences, 2014,5(5):391−398.

[22] Alhajj R, Rokne J. Distributed Process Mining, Encyclopedia of Social Network Analysis and Mining. New York: Springer-Verlag, 2014. 400−403. [doi: 10.1007/978-1-4614-6170-8_100682]

[23] Bratosin CC. Grid architecture for distributed process mining. Technische Universiteitndhoven, 2011. [doi: 10.6100/IR699500]

[24] Skolicki Z, De Jong K. The influence of migration sizes and intervals on island models. In: Proc. of the 7th Annual Conf. on Genetic and Evolutionary Computation. ACM Press, 2005. 1295−1302. [doi: 10.1145/1068009.1068219]

附中文参考文献:

[17] 舒万能.人工免疫算法的优化和关键问题研究[博士学位论文].武汉:武汉大学,2013.

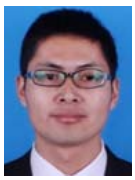
[18] 傅文渊,凌朝东.布朗运动模拟退火算法.计算机学报,2014,6(37):1301−1308. [doi: 10.3724/SP.J.1016.2014.01301]

徐杨(1970－),男,湖北武汉人,博士,讲师,主要研究领域为分布式计算,流程建模,流程分析.

汤德佑(1976－),男,博士,副教授,CCF 专业会员,主要研究领域为数据起源,数据库,高性能计算.

袁峰(1977－),男,博士,副研究员,主要研究领域为物联网,云计算,大数据.

李东(1970－),男,博士,教授,博士生导师,CCF 专业会员,主要研究领域为大数据与云计算,业务流程管理.



林琪(1991－),男,硕士,主要研究领域为流程建模,流程分析.