文章编号:1001 - 9081(2009)05 - 1389 - 04

基于并行组合模拟退火算法的过程挖掘

铁菊红,彭 辉,阿都建华
(成都信息工程学院 软件工程学院,成都 610225)
(tiejuhong@126.com)

摘 要:首先给出了过程挖掘问题的形式化描述,然后提出了一种适合过程挖掘的并行组合模拟退火算法。该算法采用因果关系矩阵作为过程模型的编码,与同类算法相比,对适应度函数、交叉和变异算子进行了改进,并利用模拟退火算法的特性提高了算法的收敛速度。仿真实验表明该算法能较有效地处理日志噪声问题。

关键词:过程挖掘;工作流;并行组合模拟退火算法;遗传算法
中图分类号: TP311 **文献标志码**:A

Process mining based on parallel recombination simulated annealing algorithm

TIE Ju-hong, PENG Hui, ADU Jian-hua
(*College of Software Engineering, Chengdu University of Information Technology, Chengdu Sichuan* 610225, *China*)

Abstract: First, the author described formal description of process mining problem, then proposed a new approach of process mining based on parallel recombination simulated annealing algorithm. This approach adopted causal matrix as the code of process model. In comparison with other similar approaches, it improved on fitness measure function, crossover and mutation genetic operators. Experimental results show that this approach can effectively deal with noise and shorten the mining time.

Key words: process mining; workflow; parallel recombination simulated annealing algorithm; genetic algorithm

0 引言

目前大部分的信息系统如 SAP R/3,PepopleSoft,Oracle,JD Edwards 等的日志文件都记载了大量的历史事务处理信息,特别是绝大部分工作流管理系统的日志文件都记载了若干企业流程各个活动的执行过程信息[1]。如何从这些更为“客观”的海量信息中抽取知识或建立工作流模型来指导或修正这些系统的运行是企业的迫切需求。过程挖掘(又称工作流挖掘)的目标就是从这些日志数据中挖掘出最优的过程模型。

在文献[2]中,首次提出利用工作流管理系统日志挖掘工作流模型,该算法利用有向图表示模型,只能描述活动间的依赖关系,并不能描述复杂结构。文献[3]在文献[2]的基础上对算法进行了扩展,能处理活动的逻辑串行和并行关系,但不能处理逻辑或关系。文献[4]采用 Petri 网来表示工作流模型,提出了 α 算法,实现了对 AND-SPLIT 和 AND-JOINS 结构的有效挖掘,同时通过启发式的方法处理噪声数据,但此方法对噪声的抵抗不是很有效。随后出现了一些对 α 算法的扩展[5-6]。基于 Petri 网的过程挖掘算法对重复活动、非可见活动、非自由选择结构、噪声和日志不完整性等方面不能进行正确挖掘,直到文献[7]提出了基于遗传算法的过程挖掘算法。遗传算法是基于全局搜索策略,除了对上述问题中的重复活动不能处理外,对其他问题在一定程度上能进行处理。文献[7]用 Petri 网来表示染色体,但很不方便,因此文献[8]提出用因果关系矩阵(Causal Matrix)来表示染色体。文献[8]在定义适应度函数时,由于考虑因素不完整,降低了对噪声的有效抵抗,且算法收敛速度慢。文献[9]对文献[8]中的算法进行了大量的仿真和真实日志数据的测试。文献[10]提出在遗传算法中融入模拟退火算法的过程挖掘方法,同样存在适应度函数的问题,而且交叉和变异算子不能有效地处理日志噪声。本文首先给出了过程挖掘问题的形式化描述,然后提出了一种适合过程挖掘的并行组合模拟退火算法[11](Parallel Recombination Simulated Annealing, PRSA),该算法与同类算法相比,对适应度函数、交叉和变异算子进行了改进,并利用模拟退火算法的特性提高了算法的收敛速度,仿真实验表明该算法能较有效地处理日志噪声问题。

1 过程挖掘问题描述

过程日志又称事件日志,记载了若干执行实例(Case)的事件序列(event trace),是过程挖掘的输入。任何基于事务的信息系统(并不要求是工作流管理系统)都可以以某种形式提供满足下列条件的过程日志[1]:1)每一件发生的事件都对应于系统中定义好的一项任务(Task);2)每一件发生的事件都属于一个执行实例;3)事件是有序的,即使是并行发生的事件也会被顺序地记录下来。

定义 1 事件序列,过程日志。令 T 是一个任务集合。$\sigma \in T^*$ 是一个事件序列。函数 $L: T^* \rightarrow N$ 是一个过程日志[9]。$L(\sigma)$ 表示事件序列 σ 在日志中出现的次数。例如,假定图 1 中的 Petri 网的一个日志 $L = [abcj, aefhgij, abcj, aehgfij]$,则 $L(abcj) = 2$, $L(aefhgij) = 1$, $L(ab) = 0$。所有过程日志的集合表示为 L'。

目前大多数挖掘算法都假定日志数据是正确的、完整的。但是实际上过程日志包含一些一些噪声,并且是不完整的。一般有两类噪声:1)日志数据是错误的,没有反应真实情况,比如由于软件的错误或者系统的临时错误配置,系统没有在日志中记载一些事件序列中的某些事件或者记错了两个事件

收稿日期:2008 - 11 - 26;**修回日期**:2009 - 02 - 17。 **基金项目**:成都信息工程学院科研基金资助项目(CRF200821)。

作者简介:铁菊红(1977 -),女,甘肃永登人,讲师,硕士,主要研究方向:工作流、数据挖掘; 彭辉(1975 -),男,甘肃玉门人,讲师,硕士,主要研究方向:工作流、数据挖掘; 阿都建华(1977 -),男,四川西昌人,讲师,硕士,主要研究方向:数据挖掘、计算机网络。

的执行顺序;另外,由于任务的并行性,相同类型的实例存在不同的事件序列。2)日志数据包含异常数据。比如日志记载了一个工作流过程的一个异常路径(exceptional path)的事件序列,一个过程的异常路径执行属于小概率事件,但它的存在影响挖掘出正确的过程模型。假定过程日志记载了所有可能的行为是不现实的,即使只包含10个任务的过程,可能存在10!个事件序列(这些任务是并行的),所以日志噪声和不完整性增加了过程挖掘的难度。

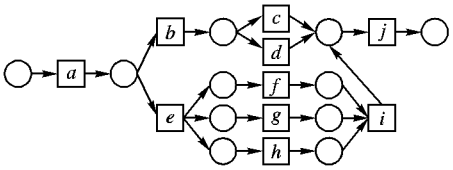

图1 一个过程模型的Petri网

对于包含n个不同任务的过程日志L,用这n个任务可以至少构造出$\sum_{k=1}^{n}\binom{n}{k}k!$个过程模型。过程挖掘就是从这么庞大的过程模型群体中寻找出一个最优模型,所谓最优模型就是用这个模型能"重新生产"出一个与过程日志L最匹配的日志。如何度量过程模型的优劣,可以用一个函数$F:\hat{W}\times L'\rightarrow[0,1]$来描述,其中$\hat{W}$是所有过程模型的集合。对于一个给定的日志$L\in L'$,一个过程模型$W\in\hat{W}$的优劣度可以用函数$F(W,L)$表示。过程挖掘问题可以形式化地描述为求解$\max F(W,L)$,s.t. $W\in\hat{W}$。显然对于无噪声和完整的日志L,$\max F(W,L)=1$,但是对于有噪声的日志L,$0\leqslant\max F(W,L)<1$。所以过程挖掘可以看作是一个最优化问题,可以用遗传算法、模拟退火(Simulated Annealing,SA)算法等优化算法求解。本文目标函数是$F(W,L)$。

2 并行组合模拟退火算法

2.1 算法步骤

遗传算法和模拟退火算法都是全局最优化方法,这些方法与文献[4-6]采用局部策略的α算法相比,能够很好地处理日志噪声、不完整性和重复活动等问题[8],并可以从所有候选过程模型中挖掘出一个最优的过程模型,这是采用局部策略的过程挖掘算法无法做到的。但是基于全局策略的遗传和模拟退化算法与α算法相比,计算速度太慢,并且适应度函数的定义并没有一个统一的标准。因此基于全局策略的过程挖掘算法的收敛速度和解的准确性都有待提高。理论上已经证明,只要模拟足够充分,模拟退火算法能收敛于全局最优解。但是对固体退火过程模拟得足够充分需要很长的时间,特别是对于自由度多且约束条件复杂的目标函数来说,其退火收敛的过程很长,寻优的效率不高。为了加速模拟退火算法收敛于全局最优解,可以借鉴遗传算法的内在并行性,从多个初始点开始并行模拟退火搜索。并行组合模拟退火算法就是借鉴遗传算法的内在并行性,在模拟退火算法中融入遗传算法的思想,即在模拟退火算法的运行过程中融入遗传算法中的群体和交叉的概念,从而实现并行组合模拟退火搜索,以提高算法的收敛速度和解的准确性,形成更有效的全局最优化方法[11-13]。

为了提高PRSA算法的收敛速度,结合本文过程模型编码、交叉和变异算子的特点(见第2.2、2.6节),提出改进的PRSA算法,其基本思路为:1)采用最优保存策略[12];2)设置双阈值使得在尽量保持最优的前提下减少计算量,即在各温度下当前状态连续C_1次保持不变,则认为抽样稳定,若连续C_2次退温过程中所得到的最优解均不变,则认为算法收敛。该算法求解步骤描述如下:

步骤1 随机生成含有M个个体的初始群体$P(0)$。

步骤2 设置初始温度参数T_0,令$j=0$,初始化个体变量*best*2。

步骤3 设置循环计数器初值$r=1$,令$i=0$,初始化个体变量*best*1。

步骤4 在温度T时,对$P(t)$中的各个个体进行随机配对,并对其中的每一对个体组作如下处理:

①进行交叉和变异运算,由两个父代个体*b*1、*b*2生成两个子代个体*c*1和*c*2。

②对由父代个体和子代个体组成的两个个体组*b*1和*c*1,*b*2和*c*2,以概率ρ接受父代个体为下一代群体中的个体,以概率$1-\rho$接受子代个体为下一代群体中的个体,其中:

$$\rho=\frac{1}{1+\exp\left(\frac{fc-fp}{T}\right)} \tag{1}$$

式中*fp*和*fc*分别为父代个体和子代个体所对应的适应度函数值。

步骤5 在当前群体中,找出目标函数值最大的个体*p*1;若$F(p1,L)=1.0$,则转步骤10。

步骤6 若$F(best1,L)<F(p1,L)$,则$best1=p1$,$i=0$;否则$i=i+1$。若$i\geqslant C_1$,则转步骤8。

步骤7 在温度T时,若步数$r<$终止步数S,则$r=r+1$,转步骤4。

步骤8 若$F(best2,L)<F(best1,L)$,则$best2=best1$,$j=0$;否则$j=j+1$。若$j\geqslant C_2$,则转步骤10。

步骤9 如果未达到冷却状态,即$T>\varepsilon$,ε为非常小的正数,则按式$T_{k+l}=\alpha T_k$(α为小于1的系数)更新温度参数T,转向步骤3。

步骤10 输出当前最优解*best*2,算法结束。

2.2 过程模型编码

应用PRSA算法时,首先要确定个体(即过程模型)的编码方式。一般过程模型用Petri网表示,但Petri网的事件日志中没有存放库所信息,因此在构建初始种群、设计遗传算子和描述AND/OR-SPLIT/JOINS结构时显得异常困难。本文采用因果关系矩阵作为每个过程模型的编码方式。

定义2[8] 因果关系矩阵。一个因果关系矩阵是一个四元组$CM=(A,C,I,O)$,其中:A表示活动的有限集合;$C\subseteq A\times A$,表示活动之间的因果关系;$I:A\rightarrow\rho(\rho(A))$,表示输入条件函数,$\rho(A)$表示集合$A$的幂集;$O:A\rightarrow\rho(\rho(A))$,表示输出条件函数。

s.t.

$C=\{(a_1,a_2)\in A\times A\mid a_1\in\cup I(a_2)\}$,$\cup I(a_2)$表示集合$I(a_2)$的并集

$C=\{(a_1,a_2)\in A\times A\mid a_2\in\cup O(a_1)\}$,

$C\cup\{(a_0,a_i)\in A\times A\mid a_{0\bullet}=\varnothing\wedge{}_\bullet a_i=\varnothing\}$是一个强连通图,$a_{0\bullet}$表示活动$a_0$的后继活动集合,${}_\bullet a_i$表示活动$a_i$的前驱活动集合。

例如,对于图1中的Petri网,其对应的因果关系矩阵为$\boldsymbol{CM}=(A,C,I,O)$,其中$A=\{a,b,c,d,e,f,g,h,i,j\}$;$C=\{(a,b),(b,c),(b,d),(c,j),(d,j),(a,e),(e,f),(e,g),(e,h),(f,i),(g,i),(h,i),(i,j)\}$;由于篇幅有限,在此仅给出个别活动的输入和输出条件函数。$I(i)=\{\{f\},\{g\},\{h\}\}$,表示$i$的前驱活动之间的逻辑关系是$f\wedge g\wedge h$,即只有$f$、$g$、$h$这三个活动都被执行后,$i$才被执行。$O(b)=\{c,d\}$,表示$b$的后继活动之间的关系是$c\vee d$,即$b$执行完之后,$c$、$d$之

间只能有一个被选择执行。本文采用集合的形式来表达各活动之间的逻辑关系,如$\{\{a,d\},\{b,c\}\}$表示$(a \vee d) \wedge (b \vee c)$,这种描述方式有利于遗传算子操作。可见,因果关系矩阵描述了两个活动之间的因果关系,每个活动的前驱活动集合和后继活动集合,以及这些活动之间的路由逻辑关系。

2.3　初始种群的创建

T为过程日志L中任务的集合,初始种群$P(0)$的大小主要取决于$|T|$,一般$|T|$越大,$P(0)$所包含的个体越多,但不应超过一个阈值M(经验值),否则 PRSA 算法的计算时间过长。由于每个过程模型个体都用一个因果关系矩阵 ***CM*** 表示,所以在创建每个个体时遵循定义 2。由此定义知,种群中的所有个体都具有相同的活动集合A,但是每个个体的因果关系C和输入和输出条件函数I和O可能不同。如果C的建立是随机的,则很可能形成不良的初始种群,影响 PRSA 算法的效率。如何确定两个活动t_1、t_2是否有因果关系(即$(t_1,t_2) \in C$),这主要依赖于一个启发式方法:若t_1、t_2的依赖关系度量函数(见定义 3)值越大,则t_1、t_2具有因果关系的概率越高。这样可以生成一个随机数$r(0 \leqslant r \leqslant 1)$,若$r \leqslant D(t_1,t_2,L)$,则$(t_1,t_2) \in C$。

定义 3　设L为一事件日志,T为L中任务的集合,t_1和t_2为T中的两个任务,t_1、t_2之间的依赖关系函数$D: T \times T \times L' \to (-1,1)$定义如下:

$$D(t_1,t_2,L) = \begin{cases} \dfrac{lp(t_1,t_2,L)+lp(t_2,t_1,L)}{lp(t_1,t_2,L)+lp(t_2,t_1,L)+1}, & t_1 \neq t_2 \text{ 且 } lp(t_1,t_2,L) > 0 \\ \dfrac{f(t_1,t_2,L)-f(t_2,t_1,L)}{f(t_1,t_2,L)+f(t_2,t_1,L)+1}, & t_1 \neq t_2 \text{ 且 } lp(t_1,t_2,L) = 0 \\ \dfrac{lp(t_1,t_2,L)}{lp(t_1,t_2,L)+1}, & t_1 = t_2 \end{cases} \tag{2}$$

其中:$f(t_1,t_2,L)$代表活动t_1、t_2以子串"t_1t_2"序列出现的日志L中的次数;$lp(t_1,t_2,L)$代表活动t_1、t_2以子串"$t_1t_2t_1$"序列出现的日志L中的次数。如果$f(t_1,t_2) \gg f(t_2,t_1)$,那么可以认为活动$t_1$和$t_2$之间存在因果关系。

日志包含的事件序列越多,以上述方法构成的因果关系C越正确。确定了因果关系C之后,就可以通过C找到每个活动t的前驱活动集合${}_\bullet t = \{x \mid (x,t) \in C\}$和后继活动集合$t_\bullet = \{x \mid (t,x) \in C\}$,然后在$\cup I(t) = {}_\bullet t$, $\cup O(t) = t_\bullet$的约束下,通过随机插入、删除和划分集合的方式,形成输入和输出条件函数$I(t)$和$O(t)$。

2.4　适应度函数的确定

评价一个过程模型的适应度过程如下:1)将过程模型编码 ***CM*** 映射为一个 Petri 网 W;2)采用 Petri 网的 fire 规则解析事件日志,在解析的过程中,如果一个即将被解析活动不能被激活,解析过程并不停止,而是把这个问题记录下来,给 Petri 网添加所需的 token 并激活该活动(这是因为过程模型可能只有部分是正确的,且日志中存在噪声);3)统计已正确解析的事件序列数、已正确解析的任务数、添加的 token 数等解析数据,根据这些数据评价过程模型的适应度。

本文将第 1 节中的目标函数$F(W,L)$作为适应度函数,$F(W,L)$的具体定义并没有一个统一的形式,但应遵循三个原则:1)一个过程模型能正确解析出日志中的事件序列越多,它的适应度越高;2)一个过程模型能正确解析出的任务越多,它的适应度越高;3)一个过程模型在解析过程中,添加的 *token* 越多,它的适应度越低。依据这三个原则,本文的$F(W,L)$的定义如式(3)所示:

$$F(W,L) = k1\frac{NumPA(W,L)}{NumA(L)} + k2\frac{NumPT(W,L)}{NumT(L)} - k3\frac{NumAddTkn(W,L)}{NumA(L)} \tag{3}$$

式中,$NumA(L)$表示日志L中的任务总数,$NumT(L)$表示日志L中的事件序列总数,$NumPA(W,L)$表示已被正确解析的任务总数,$NumPT(W,L)$表示已被正确解析的事件序列总数。$NumAddTkn(W,L)$表示添加的 *token* 总数。$k1$、$k2$、$k3$分别是系数,$k1+k2=1.0$,$0<k1<1$,$0<k2<1$,$0<k3<1$是约束条件。这三个值都是经验值,一般取$k1=0.4$,$k2=0.6$,$k3=0.2$。

2.5　初温的确定

初温的确定选择$T_0 = K\delta$的形式,其中,K为充分大的数,$\delta = f_0^{max} - f_0^{min}$,式中$f_0^{max}$ f_0^{min}为初始种群中的最大适应度函数值和最小适应度函数值。

2.6　交叉算子

两个因果关系矩阵 $\boldsymbol{CM} = (A,C,I,O)$ 交叉时,由于两个 ***CM*** 的A是相同且确定的,所以不需要交叉A,只需要交叉C、I和O,根据定义 2 中C、I、O三者之间的约束关系,从I、O可以推导出C,所以本文采取的交叉办法是:在交叉概率P_c下,随机选择一个活动t,对$I(t)$和$O(t)$分别进行单点交叉,交叉完之后更新C。对两个CM的$I_1(t)$和$I_2(t)$进行单点交叉的操作步骤如下:

1)随机选择一个交叉点 $sp1$,将$I_1(t)$划分为两个集合$RI_1(t)$和$SI_1(t)$;

2)随机选择一个交叉点 $sp2$,将$I_2(t)$划分为两个集合$RI_2(t)$和$SI_2(t)$;

3)对$SI_2(t)$中的每一个子集S采取以下四种操作之一:①将S直接加入到$RI_1(t)$中;②将S并入到$RI_1(t)$中的任意一个子集中;③从$RI_1(t)$中随机选择一个子集与S进行差运算;④从$RI_1(t)$中删除与S中相同的元素,然后将S中的每一元素分别并入到$RI_1(t)$中的任意一个子集;

4)重复第3)步,但是用$SI_1(t)$和$RI_2(t)$分别代替$SI_2(t)$和$RI_1(t)$;

5)令$I_1(t) = RI_1(t)$; $I_2(t) = RI_2(t)$。

对两个 ***CM*** 的$O_1(t)$和$O_2(t)$也进行上述单点交叉操作。交叉完两个个体的$I(t)$和$O(t)$后,有可能破坏两个个体的因果关系C,所以需要根据定义 2 更新C,使C与$I(t)$、$O(t)$保持一致。两个相同的个体使用上述交叉方法,也可能产生不同的个体,这样可以保持群体的多样性,有利于从优秀的个体中产生出更好的新个体模式。例如:设$I_1(t) = I_2(t) = \{\{a,b\},\{c,d\},\{f,e\}\}$,随机将$I_1(t)$划分为两个集合$SI_1(t) = \{\{a,b\}\}$, $RI_1(t) = \{\{c,d\},\{f,e\}\}$,随机将$I_2(t)$划分为两个集合$SI_2(t) = \{\{a,b\},\{c,d\}\}$, $RI_2(t) = \{\{f,e\}\}$;将$SI_1(t)$中的子集$\{a,b\}$直接加入到$RI_2(t)$中,将$SI_2(t)$中的子集$\{a,b\}$并入到$RI_1(t)$中的子集$\{c,d\}$中,将$RI_1(t)$中的子集$\{f,e\}$与$SI_2(t)$中的子集$\{c,d\}$进行差运算,按上述步骤交叉完之后,得到$I_1(t) = \{\{a,b,c,d\},\{f,e\}\}$, $I_2(t) = \{\{a,b\},\{f,e\}\}$。

2.7　变异算子

对个体进行变异操作可以改善 PRSA 算法的局部搜索能力,维持群体的多样性,防止早熟现象。本文采取的变异办法是:在变异概率P_m下,对每个个体的每个活动t的$I(t)$和$O(t)$进行变异操作。对$I(t)$和$O(t)$随机选择执行以下三种变异操作之一:1)随机选择一个子集,并添加一个活动$t' \in A$到该子集中;2)随机选择一个子集,从该子集中任意删除一个活动;3)重新分配$I(t)$和$O(t)$中的所有元素,形成新的一组子集。

3 实验结果

本文使用图1的Petri网作为测试的原始过程模型,随机产生13个各自包含500个事件序列的事件日志,用文献[8]的GA算法和本文的PRSA算法对这些事件日志进行测试。除了一个事件日志不包含噪声外,其余12个事件日志分别包含5%、10%、15%的丢失活动噪声、交换活动噪声和混合噪声。丢失活动噪声是指从一个事件序列中任意删除一个活动,交换活动噪声是指任意交换一个事件序列中的两个活动的顺序,混合噪声是指日志中同时包含这两种噪声。两个算法的初始群体规模都为100。GA算法的最大进化代数为300,交叉概率为1.0,变异概率为0.01,选择概率为0.01。PRSA算法的交叉概率为0.9,变异概率为0.1,初温系数K = 25,退温系数α = 0.8,终止温度ε = 0.001,终止循环步数S = 20,$C1$ = 10,$C2$ = 10,适应度函数系数$k1$ = 0.4,$k2$ = 0.6,$k3$ = 0.2。共测试10次,测试结果如表1所示,表中分数形式的测试数据表示为算法正确挖掘出的模型次数与算法运行次数之比。

表1 PRSA算法与GA算法的测试结果

噪声百分比/%	算法正确挖掘出的模型次数/算法运行次数							
	丢失活动		交换活动		混合噪声		平均进化代数	
	GA	PRSA	GA	PRSA	GA	PRSA	GA	PRSA
0	10/10	10/10	10/10	10/10	10/10	10/10	52	56
5	1/10	3/10	9/10	9/10	4/10	6/10	198	187
10	1/10	2/10	6/10	7/10	3/10	4/10	215	199
15	0/10	1/10	4/10	4/10	2/10	3/10	227	208

从表1中测试数据可以看出,本文算法与GA算法相比,在噪声处理方面更好一些,这主要是因为本文的适应度函数考虑了对增加token的处理,变异算子扩大了因果关系的搜索空间,交叉算子更易使两个相同的个体产出不同的个体,增加了群体的多样性。对于无噪声的日志,两个算法都可以100%找到正确的模型,平均进化代数也比较小,这主要是因为在无噪声的环境下适应度函数值在等于1时终止了程序,减少了进化代数。对于有噪声的日志,本文算法的平均进化代数相比GA算法还是要小一些,这主要是因为本文算法采用最优保存策略和设置双域值减少了没必要的循环。另外本文算法在高温状态下容易接受恶化解,保证群体在进化初始阶段的多样性,有利于算法快速收敛到最优解,这是文献[8]的GA算法没有考虑到的。

4 结语

针对过程挖掘的特点,将遗传算法与模拟退火算法相结合,提出了一种适合过程挖掘的并行组合模拟退火算法,此算法采用因果关系矩阵作为个体编码,适应度函数采用了罚函数的形式对解析过程中添加的token进行了处理,使适应度函数能更正确地评价一个过程模型的优劣,交叉和变异算子能够有效地从含有噪声的事件日志中正确地挖掘出过程模型,克服了日志噪声的影响。利用模拟退火算法在高温下易于增加群体多样性的特性,并采用最优保存策略和双阈值减少了没必要的遗传进化代数,提高了算法的收敛速度。

参考文献:

[1] WEIJTERS A J M M, van der AALST W M P. Workflow mining: Discovering workflow models from event-based data [C]// Proceedings of the ECAI Workshop on Knowledge Discovery and Spatial Data. [S.l.]: ECAI, 2002: 78-84.

[2] AGRAWAL R, GUNOPULOS D, LEYMANN F. Mining process models from workflow logs [C]// Proceedings of the 6th International Conference on Extending Database Technology: Advances in Database Technology. Berlin: Springer-Verlag, 1998: 469-483.

[3] PINTER S S, GOLANI M. Discovering workflow models from activities' life spans [J]. Computer in Industry, 2004, 53(3): 283-296.

[4] van der AALST W M P, WEIJTERS A J M M, MARUSTER L. Workflow mining: Discovering process models from event logs[J]. IEEE Transactions on Knowledge and Data Engineering, 2004, 16(9): 1128-1142

[5] de MEDEIROS A K A, van DONGEN B F, van der AALST W M P, *et al*. Process mining: Extending the α-algorithm to mine short loops, WP 113 [R]. Eindhoven: Eindhoven University of Technology, 2004.

[6] 李嘉菲,刘大有,杨博. 过程挖掘中一种能发现重复任务的扩展α算法[J]. 计算机学报,2007,30(8):1436-1445.

[7] de MEDEIROS A K A, WEIJTERS A J M M, van der AALST W M P. Using genetic algorithms to mine process models: Representation, operators and results, WP 124 [R]. Eindhoven: Eindhoven University of Technology, 2004.

[8] van der AALST W M P, de MEDEIROS A K A, WEIJTERS A J M M. Genetic process mining [C]// Proceedings of the 26th International Conference on Applications and Theory of Petri Nets, LNCS 3536: ICATPN 2005. Berlin: Springer-Verlag, 2005: 48-69.

[9] 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.

[10] 田珂,朱清新,向培素. 基于混合遗传算法的工作流重构研究[J]. 计算机科学,2007, 34(1):103-105

[11] MAHFOUD S W, GOLDGERG D E. Parallel simulated annealing: A genetic algorithm[J]. Parallel Computing, 1995, 21(1): 1-28.

[12] 周明, 孙树栋. 遗传算法原理及应用[M]. 北京: 国防工业出版社, 1999:80-85.

[13] 吴余生,陈胜宏. 并行组合模拟退火算法在边坡稳定分析中的应用[J]. 岩土力学,2006,27(9): 1554-1558.

(上接第1388页)

[2] BINDER R V. 面向对象系统的测试[M]. 北京: 人民邮电出版社, 2005:75-90.

[3] 黄陇,陈致明,于洪敏,等. 基于UML的软件测试用例自动生成技术研究[J]. 计算机应用与软件,2004,21(11):16-18.

[4] MONALISA S, KUNDU D, MAL R. Automatic test case generation from UML sequence diagrams[C]// Proceedings of the 15th International Conference on Advanced Computing and Communications: ADCOM. Washington, DC: IEEE Computer Society, 2007: 60-67.

[5] LI BAO-LIN, LI ZHI-SHU, LI QING. Test case automate generation from UML sequence diagram and OCL expression [C]// Proceedings of the 2007 International Conference on Computational Intelligence and Security: CIS 2007. Washington, DC: IEEE Computer Society, 2007: 1048-1052.

[6] ALI S, BRIAND L C, JAFFAR-UR M, *et al*. A state-based approach to integration testing based on UML models[J]. Information and Software Technology, 2007, 49(1/2): 1087-1106.

[7] 黄陇、于洪敏、陈致明,等. 基于UML的软件测试自动化研究[J]. 计算机应用,2004,24(7):135-137.