

一种故障树结构匹配算法及其应用

岳 鑫<sup>1</sup> 杜军威<sup>1</sup> 胡 强<sup>1</sup> 王延平<sup>2</sup>  
(青岛科技大学信息科学与技术学院 山东 青岛 266100)<sup>1</sup>  
(中国石化青岛安全工程研究院 山东 青岛 266000)<sup>2</sup>

**摘 要** 面对以故障树形式汇集的大量历史事故案例,故障树结构匹配是借鉴历史经验在有限时间、人力和成本下实现对新事故准确和全面调查的有效手段。根据事件演化的时序和因果推理的结构特征,提出一种故障树结构匹配算法。通过构造故障树结构匹配的隐马尔可夫模型,利用维特比算法预测待匹配序列的最佳序列。实验数据显示,相对于基于节点的结构匹配算法,该算法在匹配的准确性、结构缺陷的检测效果等方面有显著提升。  
**关键词** 故障树,结构匹配,隐马尔可夫模型,维特比算法,事故分析  
**中图法分类号** TP399 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2018.09.033

Fault Tree Structure Matching Algorithm and Its Application

YUE Xin<sup>1</sup> DU Jun-wei<sup>1</sup> HU Qiang<sup>1</sup> WANG Yan-ping<sup>2</sup>  
(School of Information Science and Technology, Qingdao University of Science & Technology, Qingdao, Shandong 266100, China)<sup>1</sup>  
(Sinopec Research Institute of Safety Engineering, Qingdao, Shandong 266000, China)<sup>2</sup>

**Abstract** A large number of fault trees have been designed and stored with the occurrence of numerous historical accident cases. Structure matching is an effective way to achieve accurate and comprehensive investigation of new accidents by using the existing fault trees with the limited time, manpower and cost. Based on the timing of event evolution and the structural features of causal reasoning, a fault tree structure matching algorithm was proposed. The hidden Markov model of fault tree is constructed and then the Viterbi algorithm is used to predict the optimal matching sequences. Compared with the node-based structure matching algorithm, this algorithm has significant improvement in the accuracy of matching and the detection of structural defects.  
**Keywords** Fault tree, Structure matching, Hidden Markov model, Viterbi algorithm, Accident analysis

在化工安全领域,频发事故的危害性日益严重,深入剖析每个事故的直接原因、间接原因和根原因对指导安全生产及预防未来事故具有重要意义。故障树因具有直观、表达能力强、易于分析和推理等特点,已成为分析化工领域事故的首选方法。通过数十年的积累,事故监管部门已存储了大量化工事故案例的故障树,这些故障树经领域专家分类与汇总,形成了一批按事故类型、以故障树形式表达的同类事故综合因果分析的汇总,称之为标准故障树。然而,新事故发生后,往往因分析时间短以及经验和知识水平的限制,调查人员难以进行全面而深入的调查,易导致事故原因分析不完整和不准确等现象。研究案例故障树与标准故障树的结构匹配有助于提高事故调查的准确性与完整性,自动推荐事故发生的可能原因,对指导事故分析、快速处置和精准预防具有重要意义。

对于基于故障树的事故分析,文献多集中于故障树的建模方法<sup>[1]</sup>、分析方法<sup>[2]</sup>和预测方法<sup>[3]</sup>,这类研究多以待分析的事故为顶事件,通过逐层门转换为基本事件的逻辑结构<sup>[4]</sup>、代

数结构<sup>[5]</sup>或图形结构<sup>[6]</sup>,采用割集或径集的方法进行分析。此类方法难以适应于大规模、存在共性、以事故演化分析为目标的故障树模型,因其需要研究故障树间的共性、归纳、关联和映射问题,仅以故障树中的基本事件为出发点来研究顶事件的方法,已经无法满足事故的过程演化分析的需求。Doohwan<sup>[7]</sup>提出了一种基于后缀树的模式匹配算法,Liu<sup>[8]</sup>提出了一种将高斯尺度混合(GSM)模型引入到基于树结构的正交匹配追踪重建算法,这类树结构匹配方法仅考虑了树的结构特征,并不适合存在逻辑门的故障树的结构匹配。张海威等<sup>[9]</sup>提出了一种基于自适应结构概要模型的子图匹配查询算法,根据查询图顶点的标签,利用结构概要压缩图结构,实现基于图模拟方式的子图匹配查询。李瑞远等<sup>[10]</sup>提出了一种基于动态权重包含度的子图匹配方法,利用离线索引构建和处理在线查询,并进行子图匹配。基于子图匹配的方法不能准确地匹配逻辑门语义信息,也不能准确地匹配故障树事件的时序演化问题,因此也难以用于故障树的结构匹配。

到稿日期:2017-08-30 返修日期:2017-11-04 本文受国家自然科学基金(61273180),山东省自然基金项目(ZR2012FL17),山东省重点研发计划项目(2018GGX101052,2016GGX101031),山东省优秀中青年科学家科研奖励基金(BS2015DX010)资助。

岳 鑫(1992—),女,硕士生,CCF 会员,主要研究方向为基于机器学习的安全分析技术;杜军威(1974—),男,博士,教授,CCF 会员,主要研究方向为智能软件工程与安全工程,E-mail:d-jw@163.com(通信作者);胡 强(1982—),男,博士,讲师,主要研究方向为服务计算;王延平(1968—),男,高级工程师,主要研究方向为安全工程。

由上述分析可知,故障树结构匹配不仅要考虑故障树间节点的相似度,还需要考虑:1)故障树的层次结构代表不同事件演化的时序关系,这种时序关系在故障树匹配中应得到保持;2)案例故障树的门结构特征应完整保持在标准故障树的映射事件中。隐马尔可夫模型(Hidden Markov Model, HMM)是一种概率图模型,可以实现序列模型的预测<sup>[11]</sup>、评估<sup>[12]</sup>、故障检测<sup>[13]</sup>等。因为故障树层次序列演化是一种典型的马尔可夫过程,可将待匹配的故障树转换为能够表达事件层次演化的序列,以此设计一种故障树的 HMM,进而可利用维特比算法预测观测序列的最佳匹配状态序列,实现故障树间的结构匹配,提高事故案例分析的时效性和准确性。

## 1 故障树结构匹配的基础理论

标准故障树是具有共性的一系列具体事故案例的汇总,如储罐爆炸标准树是汇总一系列储罐事故案例的故障树,形成包含各种储罐案例故障树的标准故障树,标准故障树中的事件是具体案例故障树事件的抽象和归纳。案例故障树则为某个具体事故的成因分析故障树。对于标准故障树中的“可燃气体”事件,在具体案例中可能对应为“甲烷”“天然气”等。本文研究在石化安全监管部门已经汇集专家经验形成的 9 类标准故障树的基础上,研究案例树映射到标准树的结构匹配问题。下面首先给出相关定义。

**定义 1** 案例事件集合定义为  $IV = \{\text{描述具体事故案例中引发事故的直接原因、间接原因和根原因的事件集合}\}$ 。

**定义 2** 抽象事件集合定义为  $AV = \{\text{同类事故案例中描述具有共性特征的直接原因、间接原因和根原因的事件的抽象集合}\}$ 。

**定义 3** 故障树结构定义为四元组  $FT = (V, G, E, v_0)$ , 其中:

- 1)  $V$  代表故障树的节点集合;
- 2)  $G$  为逻辑门集合,  $\forall g \in G$ , 定义逻辑门的类型  $Type(g) \in \{\text{And}, \text{Or}\}$ ;
- 3)  $E$  为标准故障树的边集合,  $E \subseteq V \times G \cup G \times V$ ;
- 4)  $v_0$  代表根节点,  $V_L = \{v \in V \mid (\exists g \in G, (v, g) \in E)\}$  为叶子节点,  $V_M = \{v \mid v \in V \mid (\exists g \in G, (v, g) \in E)\}$  为中间节点。

逻辑门和节点还需要满足如下条件:

- 1)  $\forall g \in G, |inGate(g)| = 1, |outGate(g)| \geq 1$ ;
- 2)  $v = v_0, |inNode(v)| = 0, |outNode(v)| = 1$ ;
- 3)  $\forall v \in V_M, |inNode(v)| = 1, |outNode(v)| = 1$ ;
- 4)  $\forall v \in V_L, |inNode(v)| = 1, |outNode(v)| = 0$ 。

上述条件中,  $inGate(g) = \{v \mid (v, g) \in E\}$ ,  $outGate(g) = \{v \mid (g, v) \in E\}$ ,  $inNode(v) = \{g \mid (v, g) \in E\}$ ,  $outNode(v) = \{g \mid (g, v) \in E\}$ , 定义  $|\cdot|$  为集合中元素的个数。

**定义 4** 将节点的定义域中来自  $AV$  的一类故障树称为标准故障树, 记为  $AFT = (V_A, G_A, E_A, v_0)$ , 其中  $V_A = AV$ 。

**定义 5** 将节点的定义域中来自  $IV$  的一类故障树称为案例故障树, 记为  $IFT = (V_I, G_I, E_I, v_0)$ , 其中,  $V_I = IV$ 。

**定义 6**  $FT = (V, G, E, v_0)$  为故障树, 序列  $q = q_1 q_2 \cdots q_t$  称为故障树事件序列, 当且仅当  $\forall i \in [1, t], \exists g \in G, (q_i, g) \in E \wedge (g, q_{i+1}) \in E$ 。

对于故障树事件序列中的任意两个节点  $q_i$  和  $q_{i+1}$ , 利用  $TypeGate(q_i, q_{i+1})$  表示二者之间逻辑门的类型。在上述定义

的基础上, 构建如下集合。

1)  $ChainNodes(q)$ : 事件序列  $q$  的所有节点的集合;

2)  $ChainsO(FT)$ : 以根节点为起始点的故障树  $FT$  的所有事件序列集合。

## 2 基于故障树的 HMM 的构造

通常将 HMM 记为一个五元组  $\mu = (Q, O, A, B, \pi)$ 。其中,  $Q$  为初始状态空间;  $O$  为输出符号序列, 即为观测序列集合;  $\pi, A$  和  $B$  分别是初始状态的概率分布、状态转移概率分布和符号发射概率分布。HMM 模型也可表示为  $\lambda = (A, B, \pi)$ , 下面给出故障树的 HMM 的构造方案。

1) 初始状态空间

HMM 状态空间包含标准故障树的每个节点及其节点间的状态转换, 表示为标准故障树中从顶节点深度优先遍历到叶子节点的状态序列; 由标准故障树构造为初始状态空间的过程为: 根据给定的标准故障树  $AFT = \{V_A, G_A, E_A, v_0\}$ , 生成  $AFT$  的初始状态空间  $Q = \{V_M, E_M\}$ 。构造方法如下:

①  $V_M = V_A$ ;

②  $\forall v_i, v_j \in V_A$ , 存在  $g_i \in G_A$ , 满足  $(v_i, g_i) \in E_A \wedge (g_i, v_j) \in E_A$ , 则  $(v_i, v_j) \in E_M$ 。

2) 观测序列集合

将案例故障树的事件序列化后的序列集合作为输出符号的可观测状态序列集合。由于故障树具有自上而下的因果关系层次结构, 因此可以将案例故障树转化为链式结构。按照树的深度优先遍历, 以左、中、右为优先级, 从案例故障树的根节点出发依次遍历直到其叶子节点, 遍历过程中记录下案例故障树的链条化序列(假设故障树可以转换为  $n$  个序列, 序列的长度不同):  $o_1 = o_{1,1} o_{1,2} o_{1,3} \cdots o_{1,m1}, o_2 = o_{2,1} o_{2,2} o_{2,3} \cdots o_{2,m2}, \cdots, o_n = o_{n,1} o_{n,2} o_{n,3} \cdots o_{n,mm}$ 。其中,  $o_j \in ChainO(IFT) (1 \leq i \leq n)$ ,  $o_{ij} \in V_I$ 。

3) 初始状态的概率分布

由于故障树顶事件为事故结果, 为必然发生的事件, 通过自上而下的分析确定事故的相关原因, 因此 HMM 中的初始状态概率  $\pi_i$  为从顶事件开始的演化分析。构造 HMM 中的初始状态概率  $\pi_i$  为各状态为初始状态的概率分布:

$$\pi_i = \frac{P(x_1 = q_i)}{\sum_{i=1}^N P(x_1 = q_i)}, 1 \leq i \leq N$$

其中,  $P(x_1 = q_i)$  是标准故障树中初始状态  $x_1$  以  $q_i$  为起始状态的概率。

4) 状态转移概率的分布

依据历史案例故障树到标准故障树节点间迁移结构匹配统计, 计算边关联的节点  $(v_i, v_j)$  的状态转移概率。

$$P(v_i \rightarrow v_j) = \frac{C_{i,j}}{\sum_{k=1}^N C_{i,k}}, 1 \leq i, j \leq N$$

其中,  $C_{i,j}$  为在案例故障树的映射中节点  $v_i$  转移到节点  $v_j$  的频数,  $\sum_{k=1}^N C_{i,k}$  为在案例故障树的映射中从  $v_i$  转移到所有节点的总转移频数。

构造 HMM 转移矩阵  $A$ , 令元素  $a_{ij} = P(v_i \rightarrow v_{i+j})$ , 需要满足  $\sum_{j=1}^N a_{ij} = 1$  的约束, 则按照下面的公式对转移矩阵中的每个元素进行归一化处理:

$$a_{ij} = \frac{a_{ij}}{\sum_{k=1}^N a_{ik}}$$

5)符号发射概率的分布

发射概率是标准故障树节点与案例故障树节点的映射概率。由于标准故障树的节点是案例故障树节点的一种抽象表示,其在语法和语义上都存在一定的相关性。通过网络爬虫获得 11 万篇化工事故报告,利用 word2vec 训练获得领域相关语料库,从而可以获得标准故障树和案例故障树节点的特征向量。对于标准树与案例树的符号发射概率分布矩阵  $B$ ,可以利用余弦定理计算两个节点信息的相似度  $P_{i,j}$ ,并进行归一化处理:

$$b_{ij} = \frac{P_{i,j}}{\sum_{k=1}^N P_{i,k}}, 1 \leq i \leq N, 1 \leq j \leq M$$

其中, $P_{i,j}$ 是在训练集中通过余弦定理获得状态  $q_i$  到状态  $o_j$  的两节点之间的相似度。

### 3 故障树结构相似度匹配算法

#### 3.1 序列匹配

维特比算法是一种动态规划算法<sup>[14]</sup>,可以针对 HMM 模型及一个相应的观测序列寻找生成此序列的最可能的隐藏状态序列。本文提出的故障树结构匹配是基于构造的 HMM,来求解案例故障树  $IFT=(V_I,G_I,E_I,v_0)$  中的可观测序列,以及在标准故障树  $AFT=(V_A,G_A,E_A,v_0)$  中最可能的隐藏状态序列,即已知故障树的 HMM 模型  $\lambda$ ,求观测序列  $o$  的最佳匹配序列  $q$ ,亦即求条件概率  $P((q_1,q_2,\cdots,q_T)|(o_1,o_2,\cdots,o_T),\lambda)$  达到最大值时对应的信息串  $q_1,q_2,q_3,\cdots,q_T$ 。

首先定义一个变量  $\delta_t(i)$  代表在时刻  $t$  HMM 沿着某一条路径输出观测序列  $o=o_1,o_2,o_3,\cdots,o_T$  的最大概率:

$$\delta_t(i)=\max P(q_1,q_2,\cdots,q_T,o_1,o_2,\cdots,o_T|\lambda)$$

另设置一个变量用于路径记忆,它记录该路径上在  $t-1$  时刻的状态:

$$\varphi_t(i)=\arg \max_{1 \leq j \leq N} [\delta_{t-1}(j) \cdot a_{ji}], i=1,2,\cdots,N$$

根据前文构建的 HMM,将案例故障树序列化为待匹配的观测序列,通过算法 1 求解每一条观测序列的最佳匹配序列  $l_{\text{best } o}$

**算法 1** sequenceMatching 算法

输入:模型  $\lambda=(A,B,\pi)$ ,观察序列  $o$

输出:最优状态路径  $l_{\text{best}}=(q_1,q_2,\cdots,q_T)$

initializing the probability at  $t=0$

$$\delta_1(i)=\pi_i b_i(o_1), 1 \leq i \leq N$$

$$\varphi_1(i)=0, 1 \leq i \leq N$$

For  $t=2$  to  $T$

    computing the max probability from 2 to  $T$

$$\varphi_{t+1}(i)=\max_{1 \leq j \leq N} [\delta_t(j) \cdot a_{ji}] \cdot b_i(o_{t+1}), 1 \leq i \leq N$$

$$\varphi_t(i)=\arg \max_{1 \leq j \leq N} [\delta_{t-1}(j) \cdot a_{ji}], 1 \leq i \leq N$$

End For

computing the max probability at the time of  $t=T$

$$q_T^* = \arg \max_{1 \leq i \leq N} [\delta_T(i)]$$

$$P^* = \max_{1 \leq j \leq N} [\varphi_T(i)]$$

For  $t=T-1$  to 1

    Find the best sequence from  $T-1$  to 1

$$q_t^* = \varphi_{t+1}(q_{t+1}^*)$$

End For

$l_{\text{best}}=(q_1,q_2,\cdots,q_T)$  is the best matching sequence of  $o$

#### 3.2 逻辑结构匹配

由以上模型的构造方法与模型的匹配算法,可得出以下结论。

结论 1 通过 HMM 预测的最优序列为  $l_{\text{best}}$ ,如果存在  $TypeGate(q_k,q_{k+1}) \neq TypeGate(o_k,o_{k+1})$ ,则说明观测序列  $o$  存在逻辑门错误。

本结论说明:观测序列与匹配序列不仅满足序列节点的最佳序列匹配,而且满足节点间链接逻辑门语义的匹配。构造算法去掉了节点间的逻辑门语义,序列匹配算法也仅对节点序列的相似度进行匹配,本结论给出了需要同时满足逻辑门序列相同和节点序列相似的最佳匹配序列需求。

结论 2 通过 HMM 预测案例故障树所有序列的最优序列集合,用于构成标准故障树的子树,如果该子树存在某个与门关联的节点集合与其对应标准树与门的关联节点集合不一致,则存在逻辑与门的结构缺失错误。

本结论说明:当待匹配的案例故障树的所有序列都计算出最佳匹配序列时,还需要考虑故障树结构的完整性,体现出序列组合的逻辑门结构的完整性。或门表示下层存在某一事件可导致上层事件的发生,与门表示下层包含事件全部发生导致上层事件的发生。因此,或门结构的完整性体现在序列中,与门结构的完整性体现在序列组合中,需要通过序列组合检测每个与门节点集合的包含。

通过案例树序列化后,匹配序列的标准故障树的最佳序列,进而通过序列组合,匹配到从案例树到标准树的映射结构,并同时能够检测出逻辑门等结构缺陷。对于给定的  $\lambda=(A,B,\pi)$ ,算法 2 用于求解案例故障树  $IFT=(V_I,G_I,E_I,v_0)$  中可观测序列在标准故障树  $AFT=(V_A,G_A,E_A,v_0)$  中对应的子结构。

**算法 2** FTMatching 算法

输入:IFT= $(V_I,G_I,E_I,v_0)$ ,AFT= $(V_A,G_A,E_A,v_0)$

输出:IFT<sub>L</sub>=(V<sub>L</sub>,G<sub>L</sub>,E<sub>L</sub>,v<sub>0</sub>),errorPatern

Construct HMM  $\lambda=(A,B,\pi)$  based on AFT

For each  $o$  in ChainO(IFT)

$q=\text{sequenceMatching}(o)$

    For each  $(o_k,o_{k+1})$  in  $o$  and its matching  $(q_k,q_{k+1})$  in  $q$

        if  $TypeGate(q_k,q_{k+1}) \neq TypeGate(o_k,o_{k+1})$

            print GateErrorPattern( $q_k,q_{k+1}$ )

    EndFor

$L_{\text{best}}(IFT)=L_{\text{best}}(IFT)+q$

End For

Construct IFT<sub>L</sub>=(V<sub>L</sub>,G<sub>L</sub>,E<sub>L</sub>,v<sub>0</sub>) of AFT subtree

print IFT<sub>L</sub>

For each  $g$  in IFT<sub>L</sub>

    If  $\exists g \in G_L \wedge T_{\text{pye}}(g) = \text{'AND'} \wedge \exists v_j \in (V_A - V_L) \wedge (g,v_j) \in E_A$

        print AndGateErrorPartten

End For

### 4 实验结果及分析

下面对油气中毒事故标准故障树的结构匹配进行实验分析,以验证文中所提方法的有效性。油气中毒事故标准故障树的结构如图 1 所示,节点的含义如表 1 所列。





$Accuracy_{Aij} = \frac{\text{匹配逻辑与门结构正确数}}{\text{逻辑与门总数}}$

第  $i$  类案例故障树的序列匹配准确率记为  $Accuracy_{Si}$ , 逻辑门数匹配准确率记为  $Accuracy_{Ci}$ , 逻辑与门结构匹配准确率记为  $Accuracy_{Ai}$ , 其定义分别如下:

$$Accuracy_{Si} = \frac{\sum_{k=1}^m Accuracy_{Sik}}{m}, 1 \leq k \leq m$$
$$Accuracy_{Ci} = \frac{\sum_{k=1}^m Accuracy_{Cik}}{m}, 1 \leq k \leq m$$
$$Accuracy_{Ai} = \frac{\sum_{k=1}^m Accuracy_{Aik}}{m}, 1 \leq k \leq m$$

其中,  $m$  为第  $i$  类案例故障树的数量。

综合 3 种指标定义第  $i$  类故障树的匹配率:

$$Match_{FT} = \frac{Accuracy_{Si} + Accuracy_{Ci} + Accuracy_{Ai}}{3}$$

实验收集化工事故领域的 9 类标准故障树、75 个案例故障树进行分析。案例树序列总计有 1528 条, 利用本文的匹配算法, 针对每个案例故障树序列预测的结果进行人工确认, 得到的统计结果如表 4 所列。

表 4 案例故障树匹配的基本信息

Table 4 Basic information of case fault tree matching

| 类别号        | 序号    | 案例树类型         | 案例树数( $m$ ) | 案例树序列总数 | 标准树节点数 |
|------------|-------|---------------|-------------|---------|--------|
| 1          | 1—7   | 中毒事故类案例树      | 7           | 78      | 24     |
| 2          | 8—28  | 着火破坏型爆炸类案例树   | 21          | 392     | 40     |
| 3          | 29—31 | 反应失控型爆炸类案例树   | 3           | 7       | 7      |
| 4          | 32—34 | 传热型蒸汽爆炸类案例树   | 3           | 12      | 14     |
| 5          | 35—38 | 平衡破坏型蒸汽爆炸类案例树 | 4           | 8       | 8      |
| 6          | 39—40 | 自燃着火型爆炸类案例树   | 2           | 9       | 14     |
| 7          | 41    | 超压引起的物理爆炸类案例树 | 1           | 10      | 29     |
| 8          | 42—74 | 泄露着火型爆炸类案例树   | 33          | 987     | 68     |
| 9          | 75    | 泄露着火型火灾类案例树   | 1           | 25      | 59     |
| 数量总计/ 指标平均 |       |               | 75          | 1528    | 263    |

表 5 案例故障树的匹配结果

Table 5 Matching results of case fault tree

| 类别号 | 序号    | 序列准确率/% | 逻辑门准确率/% | 与门结构准确率/% | 树匹配率/% |
|-----|-------|---------|----------|-----------|--------|
| 1   | 1—7   | 89      | 67       | 73        | 76     |
| 2   | 8—28  | 87      | 79       | 69        | 78     |
| 3   | 29—31 | 72      | 90       | 82        | 81     |
| 4   | 32—34 | 82      | 93       | 89        | 88     |
| 5   | 35—38 | 71      | 91       | 90        | 84     |
| 6   | 39—40 | 85      | 75       | 74        | 78     |
| 7   | 41    | 87      | 90       | 88        | 88     |
| 8   | 42—74 | 84      | 80       | 76        | 80     |
| 9   | 75    | 77      | 83       | 80        | 80     |
| 平均  |       | 82      | 83       | 80        | 82     |

通过实验分析, 得出了 75 个案例故障树序列化匹配、逻辑门匹配、与门结构匹配的实验结果, 通过本文算法能够预测出案例故障树序列匹配的结果, 从指标数据分析构建的案例树, 如果出现逻辑门错误、与门结构错误和构建案例序列匹配错误, 其预测的准确率平均在 80% 以上。由表 5 可得第  $i$  类案例故障树匹配的统计结果如图 3 所示。

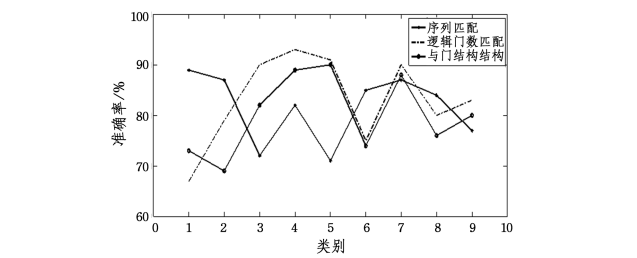


图 3 案例故障树匹配的评价指标

Fig. 3 Evaluation index of case fault tree matching

本文将图匹配常用的基于节点语义的相似度匹配方法<sup>[15]</sup>与基于 HMM 的故障树结构相似度匹配方法进行对比。前者通过网络爬虫获得 11 万篇化工事故报告, 利用 word2vec 训练得到标准故障树和案例故障树节点的特征向量, 使用余弦定理得到节点间的相似度匹配值。以第  $i$  类案例故障树  $j$  为例: 将案例故障树的节点(假设共  $S$  个节点)与其对应标准故障树的节点进行相似度匹配, 选出节点匹配值最高的  $S$  个标准故障树节点, 然后专家进行确认, 若有  $K$  个节点匹配正确, 则定义第  $i$  类案例故障树  $j$  的匹配准确率为  $P_{Nij} = \frac{K}{S}$ , 以此统计每个案例故障树在其所属类型的标准故障树中的匹配准确率值; 若第  $i$  类共有  $r$  个案例故障树, 则定

义第  $i$  类案例故障树的匹配度值为  $P_{Ni} = \frac{\sum_{k=1}^r P_{Nik}}{r}$ ; 基于节点最大相似度的匹配仅考虑了节点语义信息, 没有考虑故障树的时序演化结构, 也没有考虑故障树的逻辑门结构; 基于 HMM 的结构匹配不仅考虑了节点间的相似度, 还考虑了节点间序列结构、序列组合的门语义信息, 同时在利用故障树的树形结构特征时, 在匹配过程中逐层剪枝以降低算法的复杂度。图 4 对比了两种算法的匹配准确率。

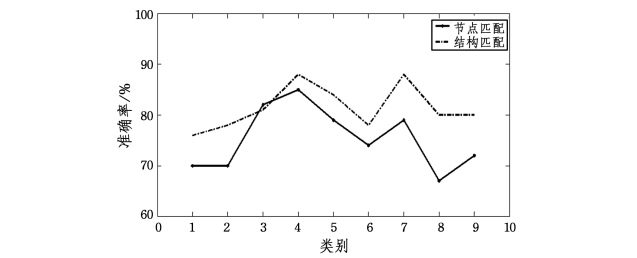


图 4 两种算法匹配率的对比

Fig. 4 Comparison of matching rates for two algorithms

如图 4 所示, 9 类案例故障树通过基于节点相似度匹配方法的匹配率并不是很高, 普遍低于基于隐马尔可夫的故障树结构相似度匹配率, 但是由于第 3 类案例故障树的节点、序列和逻辑门数较少, 其表达的故障树结构信息较少, 通过基于节点的相似度匹配也能较好地匹配其对应的标准故障树结构。通过以上实验可得, 基于隐马尔可夫的结构相似度匹配方法在针对具备较多节点、结构信息和时序演化关系的故障树时具有较好效果, 比仅仅依靠语义节点的相似度匹配方法的准确率高, 且在结构匹配的准确性和结构缺陷的检测等方面有显著提升。

[J]. Journal of Software,2011,22(2):323-338. (in Chinese)

王雅哲,冯登国,张立武,等. 基于多层次优化技术的 XACML 策略评估引擎[J]. 软件学报,2011,22(2):323-338.

[4] QI Y,CHEN J,LI Q M. XACML policy evaluation optimization method based on reordering [J]. Journal of Nanjing University of Science and Technology,2015,39(2):187-193. (in Chinese)

戚湧,陈俊,李千目. 一种基于重排序的 XACML 策略评估优化方法[J]. 南京理工大学学报,2015,39(2):187-193.

[5] CHEN J. The research on XACML strategy optimization method [D]. Nanjing: Nanjing University of Science and Technology, 2015. (in Chinese)

陈俊. XACML 策略优化方法研究[D]. 南京:南京理工大学, 2015.

[6] CHEN W H,WANG N N. Research on XACML policy evaluation optimization technology [J]. Application Research of Computer,2013,30(3):900-905. (in Chinese)

陈伟鹤,王娜娜. 基于 XACML 的策略评估优化技术的研究[J].

计算机应用研究,2013,30(3):900-905.

[7] QI Y,CHEN J,LI Q M,et al. XACML strategy optimization method based on redundancy elimination and attribute numericalization [J]. Journal of Computer Science,2016,43(2):163-168. (in Chinese)

戚湧,陈俊,李千目. 基于冗余消除和属性数值化的 XACML 策略优化方法[J]. 计算机科学,2016,43(2):163-168.

[8] eXtensible Access Control Markup Language(XACML) Version 3.0 [EB/OL]. <http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.doc>.

[9] XACML 2.0 conformance tests [EB/OL]. <http://www.oasis-open.org/committees/download.php/14846/xacml2.0-ct-v.0.4.zip>.

[10] Sun's XACML Implementation [EB/OL]. <http://sunxacml.sourceforge.net>.

[11] Enterprise XACML Implementation [EB/OL]. <http://sourceforge.net/projects/java-xacml>.

(上接第 206 页)

**结束语** 为提高事故分析中多事件演化的准确性和事件因果分析的全面性,充分利用历史专家经验,本文基于 HMM 设计了一种面向故障树结构匹配的算法。构造的 HMM 模型充分考虑了故障树结构特征,利用维特比算法检测待匹配案例故障树,并完成算法的原型系统开发。实验结果表明,算法匹配具有较高的准确率。该方法不仅有助于分析人员在事故分析中充分利用已有经验的支持,而且可以逐步建立以故障树模型为中心的同类事故因果的汇总,能够更加全面、准确地分析事故形成的机理,从而有针对性地开展安全生产与事故预防工作。后续的主要研究工作是改进历史统计数据的完备性和节点中文表达的规范性,降低状态转移概率分布与观测概率分布值之间存在的偏差,以扩大事故案例样本,进一步提升文中算法的效率和准确性。

参 考 文 献

[1] XU B F,HUANG Z Q,HU J,et al. Time Property Analysis Method for State/Event Fault Tree[J]. Journal of Software, 2015,26(2):427-446. (in Chinese)

徐丙凤,黄志球,胡军,等. 一种状态事件故障树的时间特性分析方法[J]. 软件学报,2015,26(2):427-446.

[2] TANAKA H,FAN L T,LAI F S,et al. Fault-Tree Analysis by Fuzzy Probability[J]. IEEE Transactions on Reliability, 2009, R-32(5):453-457.

[3] CUI T J,MA Y D. Establishment of DSFT and determination of spatial distribution of probability of failure [J]. Systems Engineering-Theory & Practice, 2016, 36(4):1081-1088. (in Chinese)

崔铁军,马云东. DSFT 的建立及故障概率空间分布的确定[J]. 系统工程理论与实践,2016,36(4):1081-1088.

[4] ALIEE H,ZARANDI H R. A Fast and Accurate Fault Tree Analysis Based on Stochastic Logic Implemented on Field-Programmable Gate Arrays[J]. IEEE Transactions on Reliability, 2013,62(1):13-22.

[5] NI J,TANG W,XING Y. A Simple Algebra for Fault Tree

Analysis of Static and Dynamic Systems[J]. IEEE Transactions on Reliability,2013,62(4):846-861.

[6] LI Z F,REN Y,LIU L L,et al. Parallel algorithm for finding modules of large-scale coherent fault trees[J]. Microelectronics Reliability,2015,55(9/10):1400-1403.

[7] DOOHWAN O H,WON WOO R O. High Performance Pattern Matching algorithm with Suffix Tree Structure for Network Security[J]. Pattern matching,2014,51(6):110-116.

[8] LIU P. Tree structure matching pursuit based on Gaussian scale mixtures model[J]. Proc Spie,2011,8135(1):813520-813520-8.

[9] ZHANG H W,JIE X F,DUAN Y Y,et al. A directed graph based subgraph matching query algorithm based on adaptive structure outline[J]. Chinese Journal of Computers,2017, 40(1):52-71. (in Chinese)

张海威,解晓芳,段媛媛,等. 一种基于自适应结构概要的有向标签图子图匹配查询算法[J]. 计算机学报,2017,40(1):52-71.

[10] LI R Y,HONG L. A subgraph matching method based on inclusion degree[J]. Journal of Software,2018,29(6):1792-1812. (in Chinese)

李瑞远,洪亮. 一种基于包含度的子图匹配方法[J]. 软件学报, 2018,29(6):1792-1812.

[11] LV Q,QIAO Y,ANSARI N,et al. Big Data Driven Hidden Markov Model Based Individual Mobility Prediction at Points of Interest[J]. IEEE Transactions on Vehicular Technology,2017, PP(99):1.

[12] CHOI K W,HOSSAIN E. Estimation of Primary User Parameters in Cognitive Radio Systems via Hidden Markov Model[J]. IEEE Transactions on Signal Processing,2013,61(3):782-795.

[13] SOUALHI A,CLERC G,RAZIK H,et al. Hidden Markov Models for the Prediction of Impending Faults[J]. IEEE Transactions on Industrial Electronics,2016,63(5):3271-3281.

[14] HAGENAUER J. Source-controlled channel decoding[J]. IEEE Transactions on Communications,2002,43(9):2449-2457.

[15] NING A,LI X,WANG C. ASE Approach for Large Graphs Matching[J]. Information Technology Journal, 2013,12(14): 2969-2974.