

基于贝叶斯疑似度的启发式故障定位算法*

张成^{1,2}, 廖建新^{1,2+}, 朱晓民^{1,2}

¹(北京邮电大学 网络与交换技术国家重点实验室,北京 100876)

²(东信北邮信息技术有限公司,北京 100191)

Heuristic Fault Localization Algorithm Based on Bayesian Suspected Degree

ZHANG Cheng^{1,2}, LIAO Jian-Xin^{1,2+}, ZHU Xiao-Min^{1,2}

¹(State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China)

²(EB Information Technology Co. Ltd., Beijing 100191, China)

+ Corresponding author: E-mail: liaojx@bupt.edu.cn, http://www.bupt.edu.cn

Zhang C, Liao JX, Zhu XM. Heuristic fault localization algorithm based on Bayesian suspected degree. *Journal of Software*, 2010,21(10):2610-2621. <http://www.jos.org.cn/1000-9825/3644.htm>

Abstract: Fault localization has theoretically been proven to be NP-hard. This paper takes a weighted bipartite graph, as fault propagation model, and proposes a heuristic fault localization algorithm based on Bayesian suspected degree (BSD) to reduce the computational complexity. It introduces a metric of BSD, which needs only to be calculated once, and uses incremental coverage, which makes the algorithm a low computation complexity $O(|F| \times |S|)$. Simulation results show that the algorithm has a high fault detection rate as well as low false positive rate and performs well even in the presence of unobserved and suspicious alarms. The algorithm, which has a polynomial computational complexity, can be applied to a large-scale communication network.

Key words: fault management; fault diagnosis; fault localization; fault propagation model; Bayes' formula

摘要: 故障定位问题理论上已经证明为 NP-Hard 问题。为了降低计算复杂度,以概率加权的二分图作为故障传播模型,提出了一种基于贝叶斯疑似度的启发式故障定位算法(Bayesian suspected degree fault localization algorithm,简称 BSD)。引入贝叶斯疑似度,对所有故障仅计算一遍;同时采用增量覆盖方式,使算法具有较低的计算复杂度 $O(|F| \times |S|)$ 。仿真实验结果表明,BSD 算法具有较高的故障检测率和较低的故障误检率,即使在部分告警无法观察、告警丢失和虚假等情况下,算法依然具有较高的故障检测率。BSD 算法具有多项式计算复杂度,可以满足大规模通信网故障定位的要求。

关键词: 故障管理;故障诊断;故障定位;故障传播模型;贝叶斯公式

中图法分类号: TP393 文献标识码: A

* Supported by the National Science Fund for Distinguished Young Scholars of China under Grant No.60525110 (国家杰出青年科学基金); the National Basic Research Program of China under Grant Nos.2007CB307100, 2007CB307103 (国家重点基础研究发展计划(973)); the Development Fund Project for Electronic and Information Industry of China (电子信息产业发展基金)

Received 2008-05-06; Revised 2009-02-24; Accepted 2009-03-31

通信网的复杂性、实时性和非确定性使得通过技术手段也无法完全避免故障的发生^[1,2]。为了尽量减小故障对于通信网及其业务的影响,保证通信业务的高可用性和高可靠性,需要有高效、实用的故障定位算法。通信业务故障定位的难点^[2,3]在于:(1) 通信网故障产生大量告警消息引起的告警风暴不仅降低了故障定位效率,还会淹没有用告警;(2) 故障发生的时间、位置等不确定性并具有突发性,仿真建模困难;(3) 故障具有传递性,单个故障可能会引发其他故障,加大了定位根源故障的难度;(4) 很多故障外在征兆较少且不能在发生处观测到;(5) 通信网告警可能丢失、虚假、乱序、重复、不完整等,此外,还可能存在闪断和周期性复现;(6) 通信网在同一时刻可能同时存在多个故障,其征兆信息交织在一起;(7) 异构设备和异构网络的告警格式及完备程度各不相同,全局时钟也常常出现偏移;(8) 故障定位的计算复杂度高,精确的故障定位已被证明为 NP-Hard 问题^[4,5]。

传统的故障定位技术已经在 SDH/WDM(space division multiplexing/wavelength division multiplexing)^[6,7]、网络协议诊断^[8]等方面取得了很好的应用效果,其主要特点在于:(1) 侧重于通信网络协议层的定位,特别是对链路层、网络层等网络连接的故障定位;(2) 侧重于对网络资源层的管理,主要发现和定位网络资源问题;(3) 大多采用确定性的网络故障传播模型;(4) 假定条件过于理想,如不能同时发生多个故障,不能出现虚假告警等。显然,这些技术已经不能满足当前通信业务的要求。随着 3G 及 NGN(next generation network)等网络技术的发展,故障定位技术需要考虑:(1) 要同时支持确定性故障定位和非确定性故障定位;(2) 根据具体网络和业务采取合适的故障模型及其传播模型;(3) 考虑实际网络及业务的复杂环境,告警征兆可能存在丢失和虚假的情况等;(4) 具有较高的故障检测率和较低的误检率;(5) 算法具有较低的计算复杂度,可以满足大规模通信业务的故障定位要求。

本文充分考虑了这些难点和应用需求,以概率加权的二分因果图作为故障传播模型,提出了一种基于贝叶斯疑似度的启发式故障定位算法。

本文第 1 节介绍故障定位问题的相关研究工作。第 2 节采用加权二分图作为故障传播模型,提出基于贝叶斯疑似度的启发式故障定位算法。第 3 节对模型和算法进行仿真实验,对实验结果进行分析和讨论。第 4 节总结全文并提出下一步研究工作。

1 相关工作

故障定位又称为告警关联(alarm correlation)、问题确定(problem determination)或根源故障分析(root cause analysis),是指通过分析观测到的征兆(symptom),找出产生这些征兆的真正原因。告警是最常见的征兆。因为不影响对算法的分析,本文将征兆等同于告警不加区别。一般情况下,能够对一组征兆做出解释的故障子集可能有多个,即备选故障假设(fault hypothesis)可能有多个。故障定位算法需要从所有可能的故障假设中寻找对征兆做出最优解释的故障子集。显然,最简单的方法是穷举法 ES(exhaustive search)^[9],依次产生所有 $2^{|\mathcal{A}|-1}$ 个故障子集, $|\mathcal{A}|$ 为故障数目(下同),每个子集都对征兆进行解释,计算每个子集发生的概率,从而找出最优解释的子集。这种方法简单、直观,但计算复杂度为 $O(2^{|\mathcal{A}|})$,无法满足实际故障定位的需要。

为了降低计算复杂度,Katzela 等人采用分而治之的思想,提出了 DCMMDA(divide and conquer maximum mutual dependency algorithm)算法^[9]。该算法分为划分和选择两个过程,将故障定位问题划分为 $|\mathcal{S}|$ 个具有单个告警的征兆集,通过引入最大相互依赖度(maximum mutual dependency)进行选择 and 聚合,从而使计算的复杂度降低为 $O(|\mathcal{A}|^3)$ 。为了进一步解决效率问题,Yemini 等人提出了基于代码簿的故障定位方法(coding-based fault localization)^[10],该算法采用确定性二分图作为故障传播模型,通过对故障征兆矩阵进行编码来描述故障征兆的关系,故障定位时只需解码即可找出相应的故障。这种方法的计算复杂度为 $O(|\mathcal{M}+1| \times \log(|\mathcal{A}+1|))$,其中, \mathcal{M} 为系统解码过程中可以纠正的错误数目, $|\mathcal{A}|$ 为故障数目。代码簿方法采用的是确定性故障传播模型,不适合故障征兆关系不确定的场合。此外,该算法没有学习能力,当网络变化时需要重新构造代码簿。为了对故障传播关系精确建模,Steinder 等人引入了贝叶斯网作为故障传播模型,并通过 3 种近似推理算法^[11,12]——BE 算法(bucket elimination algorithm)、IBU 算法(iterative belief updating algorithm)、IMPE 算法(iterative most probable

explanation algorithm)对模型进行求解,复杂度分别为 $O(|\mathcal{A}^2 \times \exp|\mathcal{A}|)$, $O(|\mathcal{A}^5|)$, $O(|\mathcal{A}^6|)$.可以看出,基于近似推理的贝叶斯网故障定位方法的计算复杂度依然很高.因此,Steinder等人又提出了基于二分图模型的增量信度更新的IHU算法(incremental hypothesis updating algorithm)^[13],算法采用事件驱动方式,根据依次分析到的征兆逐步更新故障假设.算法复杂度为 $O(|\mathcal{A}^2 \times |\mathcal{S}|)$.IHU算法限定了可能的故障假设的数目,一些故障假设在被更新为最优解之前可能被丢弃,告警信息的到达顺序影响算法的准确性和实用性.针对IHU算法存在的问题,黄晓慧等人基于贪婪思想提出了MCA/MCA+算法(max covering algorithm)^[14,15],算法首先找出征兆集所对应的故障子集,找出故障子集中的对应征兆最多的故障,加入故障假设,之后将此故障对应的征兆从征兆集中去除,循环此过程,直至征兆集为空.算法复杂度为 $O(|\mathcal{A} \times |\mathcal{S}|^2)$.MCA算法没有充分考虑率概率环境下故障征兆的非确定性关系,具有最多征兆的故障不一定是实际发生的故障,MCA+算法没有考虑故障发生的后验概率,并且在处理故障丢失情形时给出的判断条件缺乏合理性.

本文详细分析了故障与征兆的非确定性关系以及后验概率,采用概率加权的二分图故障传播模型,提出了基于贝叶斯疑似度的启发式故障定位算法BSD(Bayesian suspected degree fault localization).为了比较算法性能,采用MCA/MCA+算法作为对比算法.

2 故障定位模型与算法

2.1 故障传播模型

通信网及其业务发生故障时,由于网元间存在结构和功能的依赖关系,一个网元的故障可能会引起与其相关的网元的异常,从而发生所谓的故障传播.每个故障都可能产生大量告警.为了分析告警与故障之间的关系,有必要对故障传播进行建模.通常,建模对象有对网元的故障状态建模、对告警之间的关系建模、对故障与告警之间关系建模.通常采用的模型有Petri网模型^[16]、FSM模型^[17]、依赖图模型^[9]、因果图模型^[10]、贝叶斯网模型^[11,12]、二分图模型^[10].尽管依赖图模型、因果图模型以及贝叶斯网模型,具有良好的建模能力,但由于模型的计算复杂度较高,在实际研究中采用得相对较少.二分图可以视为这些模型的简化,因其既保持了一定的建模能力,又具有较低的计算复杂度,而被大量采用.

采用基于概率加权的二分图可以更好地表达故障与征兆之间的关系.概率加权二分图 $BG=(V(\mathcal{F}, \mathcal{S}), E(\mathcal{F}, \mathcal{S}), P_{\mathcal{F}, \mathcal{S}}, P_{\mathcal{S}, \mathcal{F}})$, \mathcal{F} 为故障集, \mathcal{S} 为征兆集, $V(\mathcal{F}, \mathcal{S})$ 为由故障和征兆组成的二分节点集, $E(\mathcal{F}, \mathcal{S})$ 为由故障指向征兆的有向边集合,边权重的集合为 $P_{E(\mathcal{F}, \mathcal{S})}$,表示当故障 f 发生时发出征兆 s 的概率,其中 $f \in \mathcal{F}, s \in \mathcal{S}$.当概率 $P_{E(\mathcal{F}, \mathcal{S})} \in (0, 1)$ 时为非确定性模型,当 $P_{\mathcal{F}, \mathcal{S}} \in \{0, 1\}$ 时为确定性模型.征兆域 $domain(s_i) \subseteq \mathcal{F}$ 代表与征兆 s_i 关联的所有故障的集合;故障域 $symptom(f_i) \subseteq \mathcal{S}$ 代表与故障 f_i 关联的所有征兆的集合.

图1为一个由3个故障节点、4个征兆组成的概率加权二分图的实例.故障集 $\mathcal{F}=\{f_1, f_2, f_3\}$,征兆集 $\mathcal{S}=\{s_1, s_2, s_3, s_4\}$, $domain(s_1)=\{f_1\}$, $domain(s_2)=\{f_1, f_2\}$, $domain(s_3)=\{f_1, f_3\}$, $domain(s_4)=\{f_2, f_3\}$, $symptom(f_1)=\{s_1, s_2, s_3\}$, $symptom(f_2)=\{s_2, s_4\}$, $symptom(f_3)=\{s_3, s_4\}$, $cluster(s_1)=\{f_1, f_2, f_3\}$, $cluster(s_2)=\{f_1, f_2, f_3\}$, $cluster(s_3)=\{f_1, f_2, f_3\}$, $cluster(s_4)=\{f_1, f_2, f_3\}$, $P_{\mathcal{F}, \mathcal{S}}=$

$$(0.005, 0.01, 0.008), P_{\mathcal{S}, \mathcal{F}} = \begin{pmatrix} 0.5 & 0.8 & 0.9 & 0.0 \\ 0.0 & 0.1 & 0.0 & 0.2 \\ 0.0 & 0.0 & 0.4 & 0.7 \end{pmatrix}.$$

本文采用概率加权二分图 BG 作为故障传播模型(fault propagation model,简称FPM), $FPM=(\mathcal{F}, \mathcal{S}, P_{\mathcal{F}, \mathcal{S}}, P_{\mathcal{S}, \mathcal{F}})$.故障集 $\mathcal{F}=\{f_1, f_2, \dots, f_n\}$ 表示系统中全部故障的集合,征兆集 $\mathcal{S}=\{s_1, s_2, \dots, s_m\}$ 表示系统中全部征兆的集合,故障 f_i 发生的概率为 $p(f_i)$, $P_{\mathcal{F}}=\{p(f_i) | f_i \in \mathcal{F}\}$.在 f_i 发生的情况下,出现 s_j 的概率为 $p(s_j | f_i)$, $P_{\mathcal{S}, \mathcal{F}}=\{p(s_j | f_i) | f_i \in \mathcal{F}, s_j \in \mathcal{S}\}$. $\mathcal{S}_0 \subseteq \mathcal{S}$ 为系统中能够观测到的征兆集, $\mathcal{S}_N \subseteq \mathcal{S}_0$ 为征兆观察窗口内的征兆, $p(f_i | s_j)$ 为出现 s_j 的情况下 f_i 发生的概率.

考虑到在实际系统中由于诸如预警代码嵌入不足等原因,导致故障发生时征兆已经产生但没有相关检测代码或者没有代理及上报到网管系统,从而导致征兆不能全部被观察到.由于网络丢包、协议不可靠、应用软

件出错等原因,告警还可能发生丢失现象.此外,由于管理员对告警门限设置得不准确、轻度网络拥塞等原因可能出现虚假告警的情况.如图 2 所示为概率加权二分图,征兆 s_3 发生丢失,而征兆 s_5 为虚假征兆, f_4 为其对应的虚假故障.为了刻画这些特征,我们引入如下参数^[12,13]:

- (1) 征兆可观察率 $OR, OR=|\mathcal{S}_O|/|\mathcal{S}|$;
- (2) 征兆丢失率 $LR(s), s \in \mathcal{S}_O$;
- (3) 征兆虚假率 $SSR(s), s \in \mathcal{S}_O$.

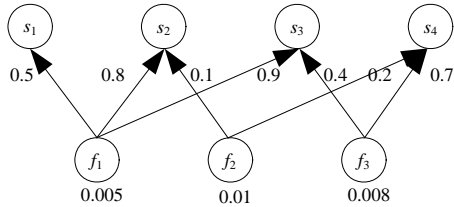


Fig.1 Probabilistic weighted bipartite graph (PWBG)

图 1 概率加权二分图

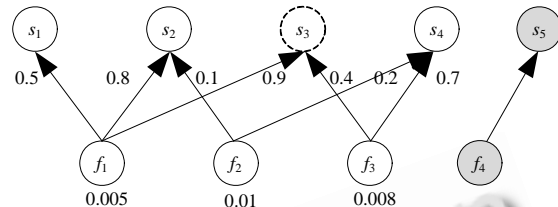


Fig.2 PWBG (considering OR, LR, SSR)

图 2 概率加权二分图(考虑 OR, LR, SSR)

2.2 故障定位算法

BSD 算法的主要思想是,通过采用启发式的探测策略,找出最有可能产生这些征兆的故障集.定义与文献[18]类似的函数来度量可能性.区别在于,后者采用归一化后的相对概率.本文采用基于贝叶斯公式的绝对概率,因此称为贝叶斯疑似度(Bayesian suspected degree),用 bsd 表示.已知在观察到征兆集 $\mathcal{S}_N (\mathcal{S}_N \subseteq \mathcal{S}_O)$ 的前提下,故障 f 发生的贝叶斯疑似度为

$$bsd(f, \mathcal{S}_N) = \frac{\sum_{s \in \mathcal{S}_N} p(f | s)}{\sum_{s_j \in \text{symptom}(f)} p(f | s_j)}$$

其中, $p(f | s) = \frac{p(f)p(s | f)}{\sum_{f_i \in F} p(f_i)p(s | f_i)}$.

对 $\text{domain}(\mathcal{S}_N)$ 中每个故障 f ,求解其贝叶斯疑似度 $bsd(f, \mathcal{S}_N)$,构成集合 \mathcal{B}_{bsd} ,并按照疑似度大小排序.当 \mathcal{B}_{bsd} 中最有可能的前 k 个故障完全覆盖了所有观测到的征兆 \mathcal{S}_N 时,就认为找到了最优的故障假设.即:满足 $\mathcal{S}_N \subseteq \mathcal{S}_{BSD}$, $\mathcal{S}_{BSD} = \sum_{f_i=1}^{i=k} \text{symptom}(f_i)$ 的最小值为 k ,最优故障假设集为 $\mathcal{H} = \{f_1, f_2, \dots, f_k\}$.

为了便于比较,BSD 算法尽量采用与 MCA/MCA+算法相同的输入.

I. 输入

- (1) 故障传播模型 $FPM = (\mathcal{F}, \mathcal{S}, P_{\mathcal{F}}, P_{\mathcal{F}, \mathcal{S}})$;
- (2) 可观察的症状集合 \mathcal{S}_O ;
- (3) 在单个时间窗口内观察到的征兆 $\mathcal{S}_N \subseteq \mathcal{S}_O$;
- (4) 征兆可观察率 OR ;
- (5) 征兆丢失率 $LR(s)$;
- (6) 征兆虚假率 $SSR(s)$.

假设:

(1) Noisy-OR 模型,即引起某个症状的多个故障相互独立,并且任何一个故障的发生都将引起该症状的发生;

(2) 故障独立假设,即不同故障之间相互独立.

II. 输出

能够对 \mathcal{S}_N 做出最优解释的故障假设集 \mathcal{H} ,具有以下性质:(1) \mathcal{S}_N 中的每个征兆可被故障假设中的至少一个

故障所解释;(2) 故障假设 \mathcal{H} 所包含的故障,最有可能产生 \mathcal{S}_N 中的征兆.

III. Algorithm BSD 算法

- (1) 设 $\mathcal{H}=\emptyset$;
- (2) 找出 \mathcal{S}_N 中每个征兆 $s_j \in \mathcal{S}_N$ 的可能故障,构成待选故障子集 $\mathcal{F}_{\mathcal{S}_N}$ 集, $\mathcal{F}_{\mathcal{S}_N}=\{f|f \in \mathcal{F}, s \in \text{symptom}(f) \cap \mathcal{S}_N\}$.
- (3) 对应 $\mathcal{F}_{\mathcal{S}_N}$ 中的每一个故障 f_i ,计算其疑似度 $\text{bsd}(f_i, \mathcal{S}_N)$,加入集合 \mathcal{B}_{bsd} 中.
- (4) 初始化征兆集 \mathcal{S}_{BSD} 为空.
- (5) 对集合 \mathcal{B}_{bsd} 中的疑似度由高向低进行排序,依次取出 $\text{bsd} \in \mathcal{B}_{\text{BSD}}$ 循环执行,直至 $|\mathcal{S}_{\text{BSD}} \cap \mathcal{S}_N|/|\mathcal{S}_N|=1$.
 - 获得 bsd 所对应的 f_i ;
 - 获得 f_i 所对应的 $\mathcal{S}_i, \mathcal{S}_i=\text{symptom}(f_i) \cap \mathcal{S}_N$;
 - 如果 \mathcal{S}_i 对 \mathcal{S}_{BSD} 增长有贡献,即 $\text{union}(\mathcal{S}_{\text{BSD}}, \mathcal{S}_i) - \mathcal{S}_{\text{BSD}} \neq \emptyset$,则 $\mathcal{S}_{\text{BSD}}=\text{union}(\mathcal{S}_{\text{BSD}}, \mathcal{S}_i)$; $\mathcal{H}=\text{union}(\mathcal{H}, f_i)$.
- (6) 输出故障假设集 \mathcal{H} .

2.3 算法对比与分析

本文的对比算法 MCA 采用覆盖率 $\text{Coverage}(f)=\{s|s \in \text{symptom}(f) \cap \mathcal{S}_N\}$ 作为备选故障 f 的选择标准.这种定义方法存在一定问题,因为故障与征兆之间的非确定性关系,覆盖征兆最多的故障很多时候并不是真正发生的故障.其改进型算法 MCA+ 虽然考虑了故障与征兆之间的概率关系,引入收益 $G(f, \mathcal{S}_N)_1 = \sum_{s \in \text{symptom}(f) \cap \mathcal{S}_N} p(s, f)$ 作为备选故障 f 的选择标准,但由于故障 f 发生时的可能产生征兆的个数不定,产生征兆的可能性也不定,简单地概率相加很难准确表征故障 f 发生的可能性.MCA+ 算法针对征兆丢失和虚假的情况提出了扩展的收益定义 $G(f, \mathcal{S}_N)_2$:

$$G(f, \mathcal{S}_N)_2 = \frac{\sum_{s \in \text{symptom}(f) \cap \mathcal{S}_N} (1 - LR(s)) \times p(s, f)}{\sum_{s_l \in \text{symptom}(f) \setminus \mathcal{S}_N} LR(s_l) \times p(s_l, f) + \sum_{s_p \in \text{symptom}(f) \cap \mathcal{S}_N} SSR(s_p) \times p(s_p, f)}$$

这个定义考虑了征兆丢失与虚假对故障定位的影响,但仅是从影响趋势角度给出了粗略的定义.特别是 $G(f, \mathcal{S}_N)_2$ 和 $G(f, \mathcal{S}_N)_1$ 存在一定的不一致性,即 $\lim_{LR \rightarrow 0, SSR \rightarrow 0} G(f, \mathcal{S}_N)_2 \neq G(f, \mathcal{S}_N)_1$,影响了故障定位的准确率.另外, MCA 和 MCA+ 算法依次产生备选故障 f 后,采取的策略是从 \mathcal{S}_N 中删除 $\text{symptom}(f)$,形成新的 \mathcal{S}_N ,直至 $\mathcal{S}_N=\emptyset$.这种策略没有充分考虑故障与征兆之间的不确定性,影响了后续备选故障的准确选择.

考虑到根据故障和征兆的先验概率作为备选故障的选择要素,常常因为先验概率估计的不够精确和准确,使计算结果难以区分甚至造成混淆,常常造成选择错误.因此,本文根据贝叶斯后验概率的思想定义了贝叶斯疑似度作为备选故障的选择标准.贝叶斯疑似度放大了备选故障之间的区别;本文由于对所有 $\text{domain}(\mathcal{S}_N)$ 求解一次疑似度,与 MCA/MCA+ 算法相比,计算复杂度减小了 $|\mathcal{S}|$ 倍,计算复杂度为 $\mathcal{O}(|\mathcal{A}| \times |\mathcal{S}|)$;不同于 MCA/MCA+ 算法采用 $\mathcal{S}_N=\emptyset$ 作为终止条件,本文采用的是 \mathcal{S}_N 被 \mathcal{S}_{BSD} 覆盖作为标准,避免了因为删除交叉征兆引起后续备选故障选择的误差.

BSD 算法基于抽象的概率二分图模型,因而具有较宽的适用范围.二分图是因果图、依赖图等复杂定位模型的简化,很多实际中的问题都可以转换为二分图,因此,算法具有很强的实用性,可用于解决通信网络中诸如网络连通性、业务可用性、叠加网络排错等各类故障定位问题.

3 仿真结果与分析

3.1 仿真实验

本文采用与文献[12,13]类似的仿真环境,并作了适当的增强,增加了故障生成时间和征兆生成时间.根据观测窗口的时间跨度,选择包含的征兆及其对应的故障.这种增强,比之前文献中采取的先选择若干故障之后再作征兆混淆的方法更贴近于实践网络环境.

仿真环境由 4 个模块组成:

- (1) 网络生成模块 $CreateNetwork(NodeSize)$;
- (2) 故障征兆关系生成模块 $CreateFSRelation(SymptomOR)$;
- (3) 故障用例生成模块 $CreateCase(GenCaseNumber,LR,SSR)$;
- (4) 根源故障定位模块 $FindRootCause(AlgorithmName)$.

本文算法为 AlgorithmMCA,AlgorithmMCA+和 AlgorithmBSD.

首先, $CreateNetwork(NodeSize)$ 生成故障集和征兆集及其对应关系.随机产生 $NodeSize$ 个节点的任意形状网络,节点间的距离随机产生.通过最小生成树 Kruskal 算法找出 $NodeSize$ 个节点的最小生成树,通过 Dijkstra 算法寻找最小生成树中任意两点的最短距离,生成距离矩阵.引入链接(link)和路径(path)的概念.直接相连的两点称为链接,两点间通过一个或多个连接起来的通路称为路径.寻找所有链接组成故障集 \mathcal{F} ,所有路径组成征兆集 \mathcal{S} .显然,一个故障(link)发生会引起多个征兆(多条路径出现异常).系统中故障总数为 $|\mathcal{F}|=NodeSize-1$,征兆总数为 $|\mathcal{S}|=C_{NodeSize}^2$;

其次, $CreateFSRelation(SymptomOR)$ 生成故障集 \mathcal{F} 与征兆集 \mathcal{S} 之间定量的非确定性关系 $P_{\mathcal{F}}$ 和 $P_{\mathcal{F},\mathcal{S}}$. $P_{\mathcal{F}}$ 服从(0.001,0.01)的均匀分布, $P_{\mathcal{F},\mathcal{S}}$ 服从(0,1)的均匀分布,当 $P_{\mathcal{F},\mathcal{S}}$ 取 0 或 1 时,变为确定性模型.确定后,采用参数 OR ,从 \mathcal{S} 中随机选取实际能够观测到的征兆 \mathcal{S}_O ,使 $OR=|\mathcal{S}_O|/|\mathcal{S}|$;

再次, $CreateCase(GenCaseNumber,LR,SSR)$ 生成 $GenCaseNumber$ 个用例集.按照概率 $P_{\mathcal{F}}$ 生成故障,对于每个已经产生的故障 f ,按照概率 $P_{\mathcal{F},\mathcal{S}}$ 产生其征兆信息 $symptom(f)$.在 $symptom(f)$ 中,征兆信息按照参数 LR 的概率“丢失”. $\mathcal{S}_O-symptom(f)$ 中的征兆信息按照参数 SSR 的概率“生成”虚假征兆.每个故障和征兆都赋予一定随机间隔的时间戳.具体仿真运行时,针对每种网络容量及参数,生成 500 个有效随机故障案例;

最后, $FindRootCause(AlgorithmName)$ 通过 AlgorithmName 算法进行故障定位输出定位结果和性能.本文采用基于时间序列的征兆观测方法. \mathcal{S}_{Ni} 定义为第 i 个观测窗口观测到的征兆集合, \mathcal{F}_C 为产生这些征兆的实际故障.由于实际环境中窗口设置很难找到准确的故障边界,因而多个故障交织的情况非常普遍.设定仿真环境中观测窗口中实际故障个数为 1,2,3,4,6 的比例分别为 30%,25%,20%,15%,10%.

本文提出的算法与 AlgorithmMCA,AlgorithmMCA+算法进行比较.算法输入都为 \mathcal{S}_{Ni} ,输出为 $\mathcal{H},DR(\mathcal{S}_{Ni}),VDR(\mathcal{S}_{Ni}),FPR(\mathcal{S}_{Ni})$.其中, $DR(\mathcal{S}_{Ni})=|\mathcal{H} \cap \mathcal{F}_C|/|\mathcal{F}_C|$, $FPR(\mathcal{S}_{Ni})=|\mathcal{H} - \mathcal{F}_C|/|\mathcal{H}|$.

$FindRootCause(AlgorithmName)$ 输出为(设窗口个数为 m):

- (1) 故障检测率: $DR = \sum_{i=1}^m DR(\mathcal{S}_{Ni}) / m$;
- (2) 故障检测率方差: $VDR = \sum_{i=1}^m \{DR(\mathcal{S}_{Ni}) - DR\}^2 / m$;
- (3) 故障误检率: $FPR = \sum_{i=1}^m FPR(\mathcal{S}_{Ni}) / m$;
- (4) 故障检测时间:TimeCost.

本文采用以上 4 项指标作为算法性能的评价标准.

实验中,网络容量分别取 5,10,...,100,征兆观察率 OR 依次取 10%,50%,100%,征兆丢失率依次取 0%,10%,征兆虚假率依次取 0%,1%.

3.2 结果分析

下面给出各种情况下的仿真结果.

- (1) 确定性故障定位场景($OR=100\%,LR=0\%,SSR=0\%$)

图 3(a)比较了 3 种算法的故障检测率.BSD 算法的故障检测率最高,除了网络规模为 5 个节点时为 99.54%以外,其余网络的故障检测率均为 100%.MCA+算法的故障检测率处于 91.84%~98.80%之间,均值为 97.47%.MCA 算法的故障检测率在 93.20%~99.20%之间,均值为 97.23%.MCA+算法与 MCA 算法故障检测率基本相同,均低于 BSD 算法.

图 3(b)比较了 3 种算法故障检测能力的稳定性.BSD 算法的故障检测率方差最小,接近于 0.其次为 MCA,

故障检测率方差在 0.0026~0.0355 之间,平均为 0.0117.MCA+算法相对较差,故障检测率方差在 0.0048~0.0438 之间,平均为 0.0146.

图 3(c)比较了 3 种算法的故障误检率,BSD 算法的故障误检率最低,在 0%~3.43%之间,平均为 0.28%.MCA+算法的误检率居次,在 4.01%~8.83%之间,平均为 6.55%.MCA 算法误检率最高,在 4.26%~9.37%之间,平均为 7.01%.

图 3(d)比较了 3 种算法的故障定位时间,BSD 算法的单位故障定位时间最短,在 0.0022s~0.0332s 之间,平均为 0.016 7s.MCA 的单位故障定位时间居其次,在 0.0018s~0.0480s 之间,平均为 0.021 9s.MCA+算法的单位故障定位时间最长,在 0.0024s~0.0602s 之间,平均为 0.028 2s.

可见,3 种算法在确定性的故障定位场景下都具有较高的故障检测率、较小的故障误检率以及较少的故障定位时间.BSD 算法优于 MCA 算法和 MCA+算法.

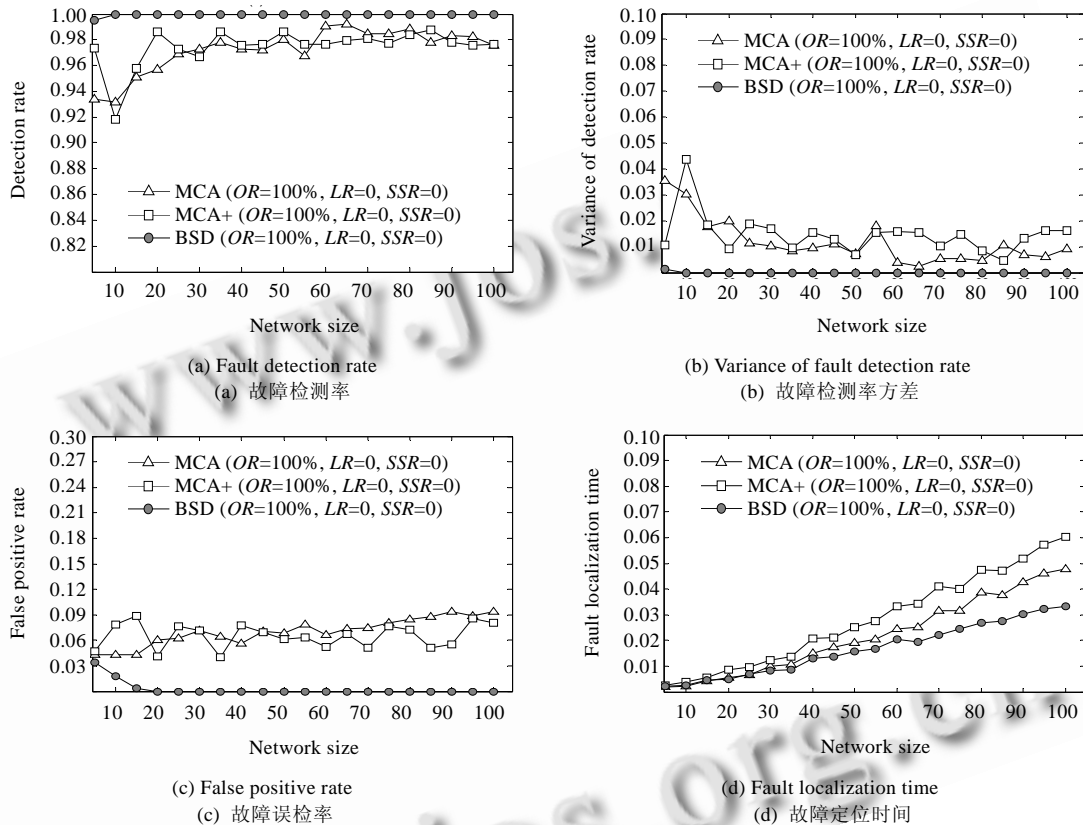


Fig.3 Determinate fault scenario ($OR=100\%$, $LR=0$, $SSR=0$)

图 3 确定型故障场景($OR=100\%$, $LR=0\%$, $SSR=0\%$)

(2) 非确定性故障定位场景 I($OR=100\%$, $LR=0\%$, $SSR=0\%$)

从图 4(a)可以看出,BSD 算法具有较高的故障检测率,除了网络规模为 5 时故障检测率较低,为 90%以外,其余网络故障检测率均达到或接近 100%,平均为 99.43%.MCA 算法的故障检测率在 87.34%~98.23%之间,平均为 96.74%.MCA+算法的故障检测率在 89.69%~96.94%之间,平均为 94.52%.相比于图 3(a),BSD 算法的故障检测率稍有下降,主要是由于故障与征兆之间不确定性引起的.

从图 4(b)可以看出,BSD 算法的故障检测率方差最小,处于 0~0.0498 之间,平均为 0.002 9.其次是 MCA 算法,处于 0.0056~0.0702 之间,平均为 0.014 4.MCA+算法较差,处于 0.0166~0.0588 之间,平均为 0.033 7.故障检测率

的方差,相比于确定性网络定位场景有所增大。

从图 4(c)可以看出,BSD 算法的故障误检率最小,处于 0%~5.12%之间,平均为 0.72%。MCA 算法误检率在 2.44%~10.07%之间,平均为 7.32%。而 MCA+算法的误检率在 10.02%~18.92%之间,平均为 14.95%。故障误检率相比于确定性网络定位场景有所增加。

从图 4(d)可以看出,BSD 算法的单位故障定位时间处于 0.0031s~0.0429s 之间,平均为 0.020 2s。MCA 算法的单位故障定位时间处于 0.0022s~0.0616s 之间,平均为 0.026 9s。MCA+算法的单位故障定位时间处于 0.0027s~0.0789s 之间,平均为 0.035 3s。因为主要推理过程、计算步骤、分析的故障和征兆都没有发生变化,所以确定性与不确定性故障定位的时间基本一致。

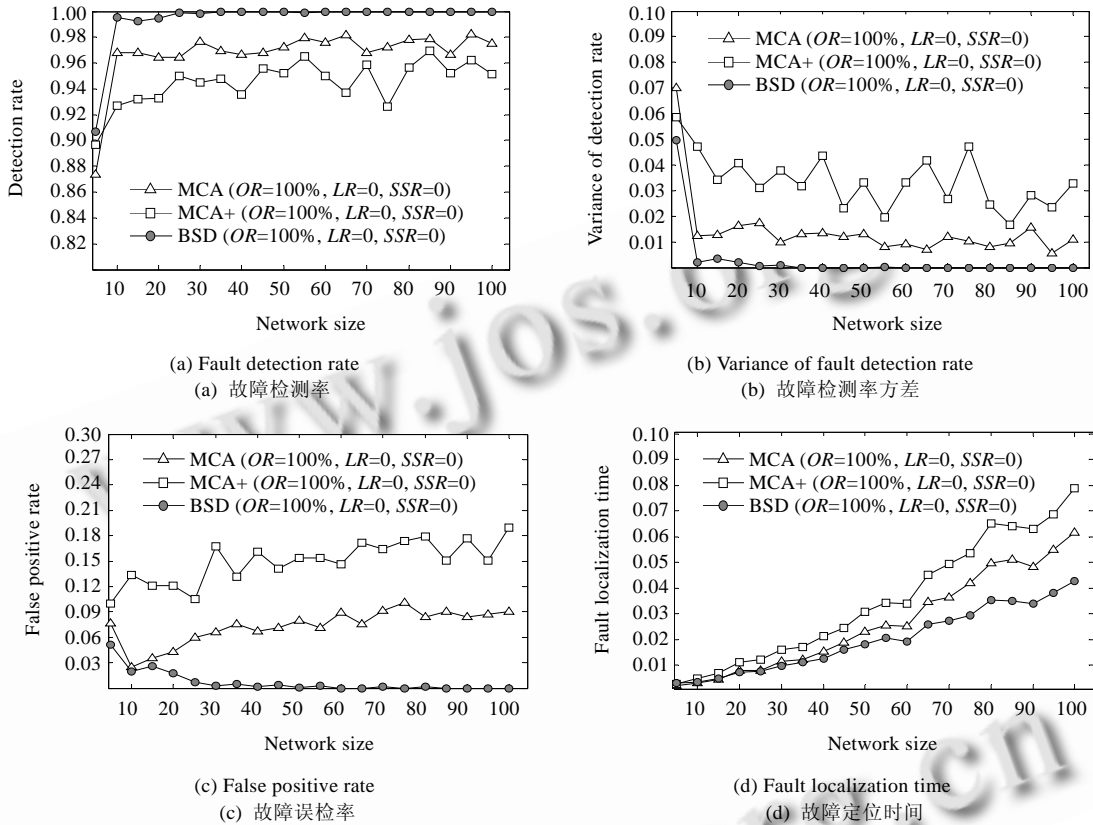


Fig.4 Non-Determinate fault scenario I (OR=100%, LR=0, SSR=0)
图 4 非确定型网络 I (OR=100%, LR=0%, SSR=0%)

(3) 非确定性故障定位场景 II (OR=50% 和 10%, LR=0%, SSR=0%)

从图 5(a)可以看出,BSD 算法的故障检测率在 OR=50% 时处于 79.83%~100% 之间,平均为 97.88%;在 OR=10% 时处于 67.79%~98.72% 之间,平均为 89.14%。MCA 算法的故障检测率在 OR=50% 时处于 58.51%~96.86% 之间,平均为 92.02%;在 OR=10% 时处于 59.05%~89.06% 之间,平均为 77.04%。MCA+算法的故障检测率在 OR=50% 时处于 80.69%~92.06% 之间,平均为 88.62%;在 OR=10% 时处于 63.02%~87.07% 之间,平均为 71.56%。在征兆存在无法观测的情况下,故障检测率均有所下降,特别是在征兆观测率较低时,故障检测率下降得比较明显。BSD 算法的故障检测率要高于 MCA 算法和 MCA+算法。

从图 5(b)可以看出,BSD 算法的故障检测率方差在 OR=50% 时处于 0~0.1019 之间,平均为 0.010 3;在 OR=10% 时处于 0.0044~0.1545 之间,平均为 0.055 3。MCA 算法的故障检测率方差在 OR=50% 时处于 0.0125~

0.1694 之间,平均为 0.035 0;在 $OR=10\%$ 时处于 0.0456~0.1803 之间,平均为 0.107 4.MCA+算法的故障检测率在 $OR=50\%$ 时处于 0.0424~0.1089 之间,平均为 0.066 5;在 $OR=10\%$ 时处于 0.0631~0.1726 之间,平均为 0.137 2.

从图 5(c)可以看出,BSD 算法的故障误检率在 $OR=50\%$ 时处于 1.28%~13.30%之间,平均为 4.30%;在 $OR=10\%$ 时处于 12.07%~37.92%之间,平均为 21.25%.MCA 算法的故障误检率在 $OR=50\%$ 时处于 6.28%~34.66% 之间,平均为 9.554%;在 $OR=10\%$ 时处于 10.82%~35.03%之间,平均为 20.82%.MCA+算法的故障误检率在 $OR=50\%$ 时处于 12.98%~23.30%之间,平均为 17.75%;在 $OR=10\%$ 时处于 8.62%~34.56%之间,平均为 29.31%.

从图 5(d)可以看出,BSD 算法的故障定位时间在 $OR=50\%$ 时处于 0.0030s~0.0329s 之间,平均为 0.015 3s;在 $OR=10\%$ 时处于 0.0018s~0.0152s 之间,平均为 0.007 7s.MCA 算法的故障定位时间在 $OR=50\%$ 时处于 0.0018s~0.0167s 之间,平均为 0.007 6s;在 $OR=10\%$ 时处于 0.0018s~0.0168s 之间,平均为 0.007 6s.MCA+算法的故障定位时间在 $OR=50\%$ 时处于 0.0027s~0.0596s 之间,平均为 0.025 6s;在 $OR=10\%$ 时处于 0.0020s~0.0233s 之间,平均为 0.010 5s.随着征兆观察率 OR 的降低,算法的性能有所下降.由于分析征兆的减少,导致故障定位时间有所减少. OR 越低,故障定位所需的时间就越少.

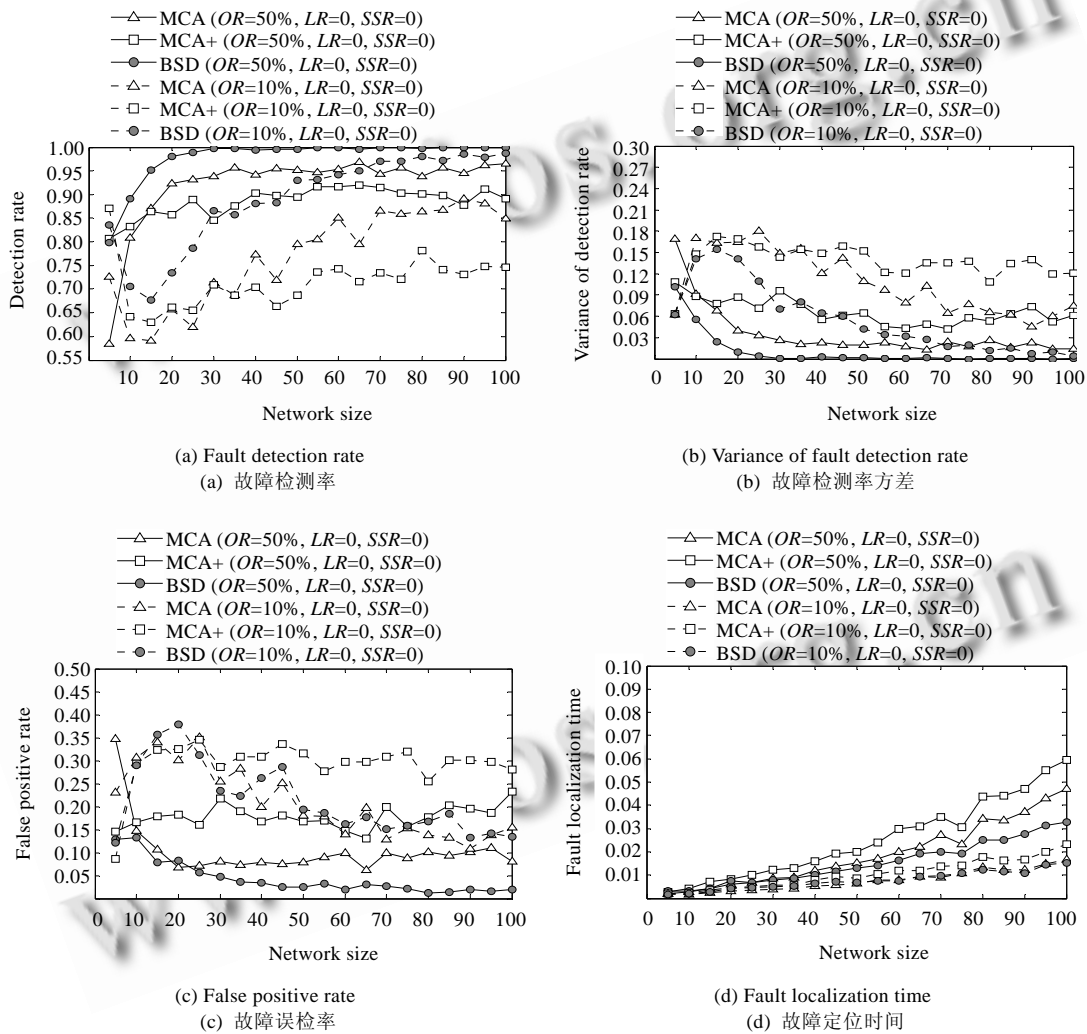


Fig.5 Non-Determinate fault scenario II ($OR=50\%$ and 10% , $LR=0$, $SSR=0$)

图 5 非确定型网络 II($OR=50\%$ 和 10% , $LR=0\%$, $SSR=0\%$)

(4) 非确定性故障定位场景 III($OR=50\%$ 和 10% , $LR=10\%$, $SSR=1\%$)

图 6 给出了在征兆观察率 OR 为 50% 和 10% 、 LR 为 10% 、 SSR 为 1% 情况下的故障场景分析曲线.可以看出,在征兆观察率 OR 下降以及存在丢失和疑似的情况下,BSD 算法的故障检测率尽管有明显下降,但依然还维持在较高水平.由于 SSR 的存在($SSR=1\%$,意味着对于容量为 100 的网络中存在 $1\% \times C_{100}^2=45$ 个虚假征兆),相比于图 5,BSD 算法存在非常高的故障误检率.故障检测率和故障误检率是一对矛盾的统一体.在存在疑似征兆的情况下如何做好更好地平衡两者,还需要作进一步的研究.

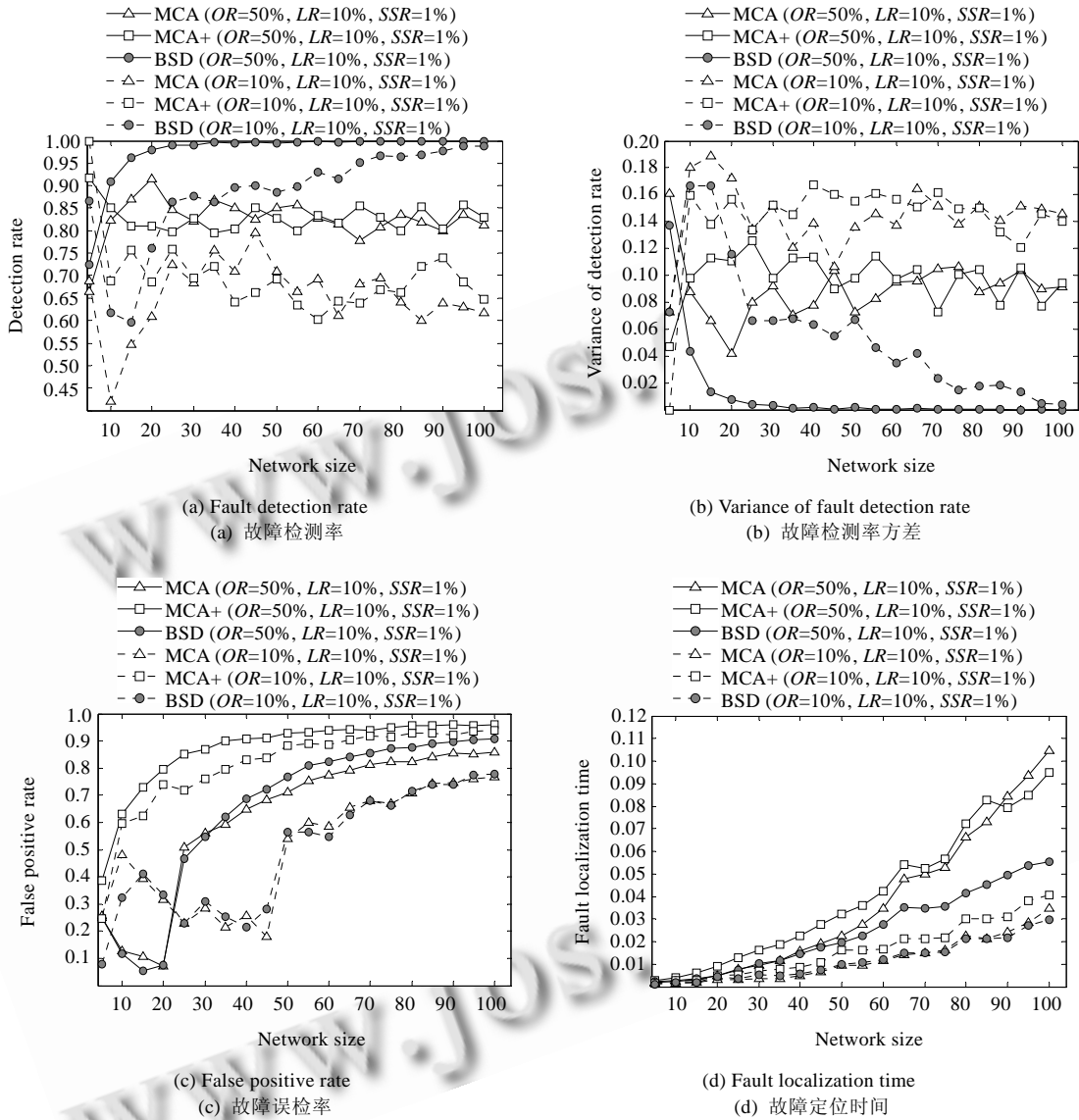


Fig.6 Non-Determinate fault scenario III ($OR=50\%$ and 10% , $LR=10\%$, $SSR=1\%$)

图 6 非确定型网络 III($OR=50\%$ 和 10% , $LR=10\%$, $SSR=1\%$)

4 结束语

本文提出了一种基于贝叶斯疑似度的启发式故障定位算法——BSD 算法.相比于 MCA 和 MCA+算法,该

算法具有更高的故障检测率、更低的故障误检率以及更少的故障定位时间.由于对所有故障仅计算一遍贝叶斯疑似度,因而 BSD 算法的计算复杂度相比 MCA 和 MCA+算法有显著降低,为 $O(|\mathcal{P} \times \mathcal{S}|)$.可以满足大规模业务网络故障定位的需要.BSD 算法还没有考虑如何利用 *OR,LR,SSR* 等相关信息,一旦利用上这些信息,则可以进一步提高噪音环境下故障定位的准确率,特别是可以有效降低误检率.此外,BSD 算法假定理想情况下故障征兆间的概率关系,而实际系统中这种概率关系不可能估计得非常准确.采用何种学习算法来提高概率关系的准确性,以及在估计不准的情况下对故障定位准确率有何影响,这些都将成为进一步的研究方向.

References:

- [1] Lü XR, Liao JX, Zhu XM, Wu JC. Research on application continuity in IMS. *Journal of Electronics & Information Technology*, 2007,29(9):2217–2221 (in Chinese with English abstract).
- [2] Steinder M, Sethi AS. A survey of fault localization techniques in computer networks. *Science of Computer Programming*, 2004,53(22):165–194.
- [3] Steinder M, Sethi AS. A survey of fault localization techniques in computer networks. *Science of Computer Programming*, 2004, 53(2):165–194. [doi:10.1016/j.scico.2004.01.010]
- [4] Vomlelova M, Vomlel J. Troubleshooting: NP-Hardness and solution methods. *Soft Computing*, 2003,7(5):357–368.
- [5] Rao NSV. Computational complexity issues in operative diagnosis of graph-based systems. *IEEE Trans. on Computers*, 1993,42(4): 447–457. [doi:10.1109/12.214691]
- [6] Mas C, Thiran P. A review on fault location methods and their application to optical networks. *Optical Networks Magazine*, 2001, 2(4):73–87.
- [7] Mas C, Thiran P. An efficient algorithm for locating soft and hard failures in WDM networks. *Journal of Selected Area Communications, Special Issue on Protocols and Architectures for Next Generation WDM Optical Networks*, 2000,18(10): 1900–1911.
- [8] Zhao Y, Chen Y, Bindel D. Towards unbiased end-to-end network diagnosis. *ACM SIGCOMM Computer Communication Review*, 2006,36(4):219–230. [doi: 10.1145/1151659.1159939]
- [9] Katzela I, Schwartz M. Schemes for fault identification in communication networks. *IEEE/ACM Trans. on Networking*, 1995,3(6): 733–764.
- [10] Yemini S, Klinger A, Mozes S, Yemini E, Ohsie YD. High speed and robust event correlation. *Communications Magazine*, 1996, 34(5):82–90. [doi: 10.1109/35.492975]
- [11] Steinder M, Sethi AS. End-to-End service failure diagnosis using belief networks. In: *Proc. of the Network Operations and Management Symp. (NOMS)*. Florence, 2002. 375–390. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1015595
- [12] Steinder M, Sethi AS. Probabilistic fault localization in communication systems using belief networks. *IEEE/ACM Trans. on Networking*, 2004,12(5):809–822. [doi: 10.1109/TNET.2004.836121]
- [13] Steinder M, Sethi AS. Probabilistic event-driven fault diagnosis through incremental Hypothesis Updating. In: *Proc. of the IFIP/IEEE Int'l Symp. on Integrated Network Management (IM)*. Colorado Springs, 2003. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1194216
- [14] Huang XH, Zou SH, Wang WD, Cheng SD. Fault management for Internet service: Modeling and algorithms. In: *Proc. of the IEEE Int'l Conf. on Communications (ICC)*. 2006. 854–859. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4024235
- [15] Huang XH, Zou SH, Wang WD, Cheng SD. Internet services fault management: Layering model and algorithm. *Journal of Software*, 2007,18(10):2584–2594 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/18/2584.htm> [doi: 10.1360/jos182584]
- [16] Benveniste A, Fabre E, Haar S, Jard C. Diagnosis of asynchronous discrete-event systems: A net unfolding approach. *IEEE Trans. on Automatic Control*, 2003,48(5):714–727. [doi: 10.1109/TAC.2003.811249]
- [17] Rouvellou I, Hart GW. Automatic alarm correlation for fault identification. In: *Proc. of the 14th Annual Joint Conf. of the IEEE Computer and Communications Societies (INFOCOM'95)*. Bringing Information to People, 1995. 553–561. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=515921

- [18] Tang YN, Al-Shaer ES, Boutaba R. Active integrated fault localization in communication networks. In: Proc. of the 9th IFIP/IEEE Int'l Symp. on Integrated Network Management (IM 2005). Nice, 2005. 543-556. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1440826

附中文参考文献:

- [1] 吕新荣,廖建新,朱晓民,武家春. IMS 应用层连续性研究. 电子与信息学报, 2007, 29(9): 2217-2221.
- [15] 黄晓慧, 邹仕洪, 王文东, 程时端. Internet 服务故障管理分层模型和算法. 软件学报, 2007, 18(10): 2584-2594. <http://www.jos.org.cn/1000-9825/18/2584.htm> [doi:10.1360/jos182584]



张成(1976-),男,辽宁盘锦人,博士,主要研究领域为业务网络智能化.



朱晓民(1974-),男,博士,副教授,主要研究领域为业务网络智能化.



廖建新(1965-),男,博士,教授,博士生导师,CCF 会员,主要研究领域为业务网络智能化.