

# 二元决策图 BDD 在话务故障分析中的应用<sup>①</sup>

刘林茂 边耐政 (湖南大学 软件工程系 湖南 长沙 410082)

**摘要:** 针对移动网络通话问题日益严重, 需要建立一个故障树分析(FTA)模型。利用 BDD 技术分析各个基本事件的结构重要度、概率重要度和临界值重要度。基本事件的排序对故障树生成的 BDD 节点个数有直接影响, 以及节点的结构重要度和概率重要度都有影响。采用相邻底事件优先排序法, 能够尽量减少 BDD 节点个数和对重要度的影响。利用 BDD 法比传统 FTA 法计算的重要度数值更接近, 计算效率更高。

**关键词:** 故障树分析(FTA); BDD; 结构重要度; 概率重要度; 临界值重要度

## Application of Binary Decision Diagram BDD to the Network Call Fault Analysis

LIU Lin-Mao, BIAN Nai-Zheng

(Software Engineering, Hunan University, Changsha 410082, China)

**Abstract:** With the growing problems of mobile network calls, it is necessary to create a fault tree analysis (FTA) model. Using the BDD method, this paper analyses each basic event's structural importance, probability importance and critical importance. The order of the basic event of fault tree has a direct impact on the number of BDD nodes, as well as the structural importance and probability importance degree. With the neighbor-event-first method, the number of BDD nodes and the important degree of impact can be minimized. The importance degree with BDD method value is closer to calculate than the traditional FTA method. And the calculation is more efficient.

**Keywords:** fault tree analysis (FTA); BDD; structural importance; probability importance; critical importance

二元决策图(Binary Decision Diagrams),简称 BDD,自 1993 年由 A.Rauzy 引入到故障树分析领域中且重新构建了完整的故障树分析框架<sup>[1]</sup>,已广泛应用于计算机网络、电子电路、逻辑综合和逻辑验证。BDD 技术在故障树的分析中显示出其计算效率高、计算结果精确、可方便利用计算机编程来实现等特点,近年来在系统可靠性分析中日益受到重视,得到广泛应用<sup>[2]</sup>。

## 1 引言

移动通信公司,随着移动通信用户的不断增加,网络质量问题日益严重,尤其是掉话,未接通,通话质量差这几个话务类故障。而网络规划/优化人员受制于各种原因,对该问题确实没有很好的预防、监控和

解决办法,只有花更多的力气去查找引发故障的事件及其原因,工作效率低下。迫切需要建立一个故障树分析诊断的系统,对话务故障进行预防、监控,及时提供引起故障的可能事件和原因,提高故障的解决效率。

故障树分析(FTA)是 NP 问题,其理论算法都是基于求解最小割集<sup>[3]</sup>,大型故障树分析计算量大且割集繁多,需耗费大量计算机时空资源。因此,如何在有限的计算资源内提高分析效率一直是该研究领域的一个难题。利用 BDD 和计算机技术,大型的故障树也可以被有效地自动化分析与处理。但是,在将故障树转化为 BDD 的过程中,需要先对底事件进行排序,最后产生的 BDD 的节点数很大程度上依赖于输入的底事件的排列次序。一个不合适的排序方法会导致 BDD 的规模呈指数级增长<sup>[4]</sup>,基本事件数量大,则需要耗费

<sup>①</sup> 收稿时间:2009-11-22;收到修改稿时间:2010-01-08

大量计算机时空资源。同时底事件的排序也会部件的结构重要度和概率重要度产生影响，从而影响对顶事件的发生概率。

本论文将利用 BDD 技术，对研究探索引起无线网络话务类故障建立故障树模型，分析故障发生概率和基本事件的重要度。

## 2 网络话务问题故障树建模

### 2.1 建立故障树模型

故障树的建造是 FTA 法的关键，故障树建造的完善程度将直接影响定性分析和定量计算结果的准确性。建树一般可按下列步骤进行：(1)合理选择顶事件。顶事件是指人们不希望发生的显著影响系统技术性能、经济性、可靠性和安全性的故障事件。在充分熟悉系统及其资料的基础上，做到既不遗漏又分清主次地将全部重大故障事件一一列举，必要

时可应用 FMEA<sup>[5]</sup>，然后再根据分析的目的和故障判据确定出本次分析的顶事件。(2)建树过程。一般建树方法可分为两大类：演绎法和计算机辅助建树的合成法或决策表法。演绎法的建树方法为：将已确定的顶事件写在顶部矩形框内，将引起顶事件的全部必要而又充分的直接原因事件置于相应原因事件符号中画出第二排，再根据实际系统中它们的逻辑关系用适当的逻辑门联结事件和这些直接原因事件。如此，遵循建树规则逐级向下发展，直到所有最低一排原因事件都是底事件为止。(3)故障树的简化。由真实系统图得到一个主要逻辑关系等效的简化系统图。

话务类问题的分析非常复杂，引起话务类问题的事件非常繁多，事件之间关系错综复杂，引起话务类问题常见的和主要的事件定义起来总计有中间事件 32 个，底事件 44 个。故障树如下图 1 所示：

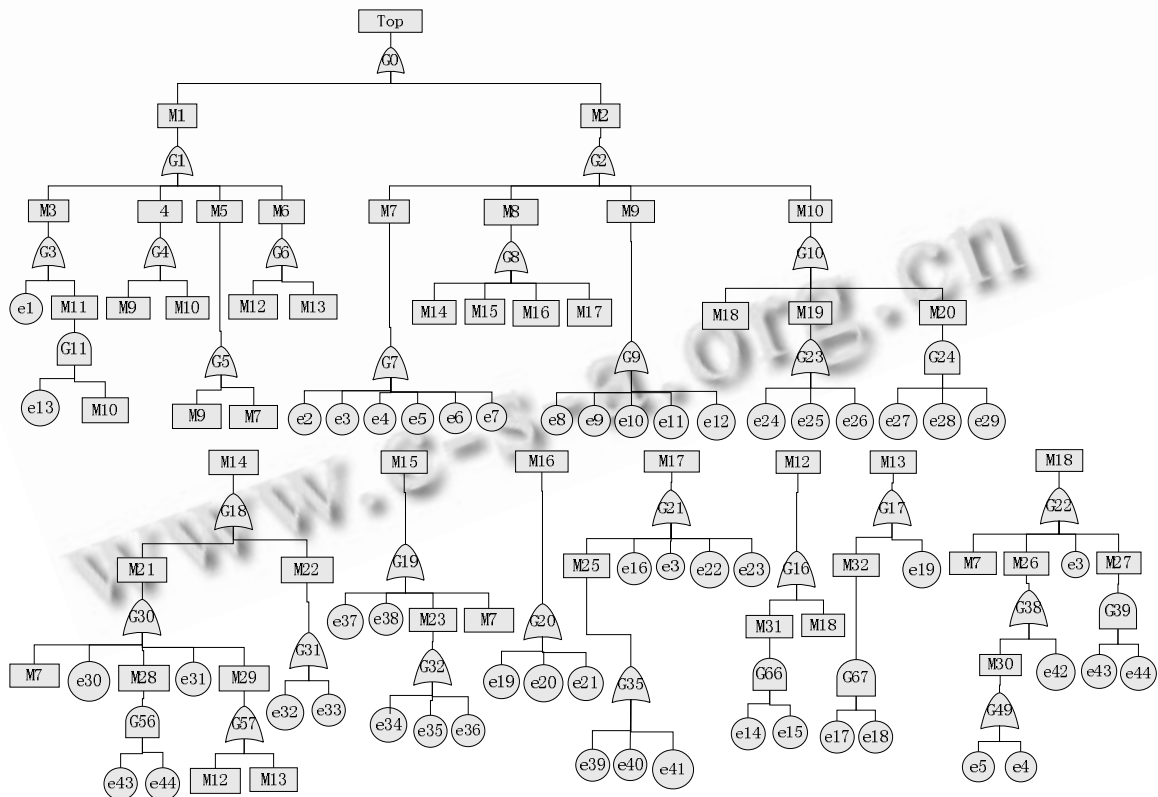


图 1 话务问题故障树模型

顶事件 TOP 是由底事件引发的，以上定义的底事件都是相互独立的。

### 2.2 故障树转化成 BDD

在引言中，基本事件的排序对生成的 BDD 节点个数有直接影响，以及节点的结构重要度和概率重要度都有影响。相邻底事件优先排序法，基于底事件之间的邻近关系来安排其优先顺序。整个 BDD 底事件排序过程是静态的，而 BDD 的构造过程则是动态的，实践这种排序方法可有效提高排序性能<sup>[6]</sup>。话务类故障事件数量大，底事件间关系复杂，采用相邻底事件优先排序法<sup>[7]</sup>是比较适合的。其排序规则为：

(1)自顶向下分析，首先分析最上层的底事件；若最上层底事件有多个，取这些底事件中重复次数最多的一个底事件；若有多个重复次数相同的底事件，选底事件所在子树最小的那个底事件；若还相同，则从左往右选取底事件。选出前面选中底事件的兄弟事件，记为集合  $A_i (i = 1, 2, 3 \dots n)$ ，对于集合内的底事件，重复次数多的先排序；若相同，则从左往右排序。

(2)向下追踪。如果存在最近选中底事件且尚未排序的兄弟树时，则在这些子树中查找是否出现前面已排序的底事件。①当查找到存在已排序的底事件时，若出现多个，选出前面最早排序的那个底事件；若那

个底事件多次出现，则取最上层的那个底事件；若最上层又相同，则左往右选取底事件，并将它的兄弟事件记为  $A_i$ 。再根据是否存在最近选中底事件的兄弟树，重复(2)或(3)。②当查找的这些兄弟树中不存在前面已经排序的底事件时，从左往右选一棵兄弟树视为步骤(1)中的故障树，重复(1)。

(3)向上追踪。如果最近选中底事件不存在兄弟树或所有兄弟树都已排序，则在最近选中底事件的父树的父树中查找是否还存在未排序的底事件；若其父树的父树中所有底事件都已排序，再次向上追踪，直到出现未排序的底事件。当出现未排序的底事件时，在这些未排序的事件中查找是否存在前面已排序的事件。①如果存在已排序的底事件，若出现多个，选出前面最早排序的那个底事件；若那个底事件多次出现，则取最上层的那个底事件；若所在最上层又相同，则任选一个，并将它的兄弟事件记为  $A_i$ ；排序后，再根据是否存在最近选中底事件的兄弟树，重复(2)或(3)。②如果不存在前面已经排序的底事件，则将查找的那棵树视为步骤(1)中的故障树，重复(1)。

当所有的底事件都被排序后，查找结束，最后得到集合  $A_1, A_2, A_3, \dots, A_n$ 。根据上面排序规则，排序集合如下表 1 所示：

表 1 底事件排序集合

序号	排序集合	序号	排序集合	序号	排序集合
A <sub>1</sub>	e <sub>1</sub>	A <sub>8</sub>	e <sub>27</sub> <e <sub>28</sub> <e <sub>29</sub>	A <sub>15</sub>	e <sub>37</sub> <e <sub>38</sub>
A <sub>2</sub>	e <sub>13</sub>	A <sub>9</sub>	e <sub>8</sub> <e <sub>9</sub> <e <sub>10</sub> <e <sub>11</sub> <e <sub>12</sub>	A <sub>16</sub>	e <sub>34</sub> <e <sub>35</sub> <e <sub>36</sub>
A <sub>3</sub>	e <sub>3</sub>	A <sub>10</sub>	e <sub>14</sub> <e <sub>15</sub>	A <sub>17</sub>	e <sub>19</sub> <e <sub>20</sub> <e <sub>21</sub>
A <sub>4</sub>	e <sub>3</sub> <e <sub>4</sub> <e <sub>5</sub> <e <sub>2</sub> <e <sub>6</sub> <e <sub>7</sub>	A <sub>11</sub>	e <sub>19</sub>	A <sub>18</sub>	e <sub>3</sub> <e <sub>16</sub> <e <sub>22</sub> <e <sub>23</sub>
A <sub>5</sub>	e <sub>42</sub>	A <sub>12</sub>	e <sub>17</sub> <e <sub>18</sub>	A <sub>19</sub>	e <sub>39</sub> <e <sub>40</sub> <e <sub>41</sub>
A <sub>6</sub>	e <sub>43</sub> <e <sub>44</sub>	A <sub>13</sub>	e <sub>30</sub> <e <sub>31</sub>		
A <sub>7</sub>	e <sub>24</sub> <e <sub>25</sub> <e <sub>26</sub>	A <sub>14</sub>	e <sub>32</sub> <e <sub>33</sub>		

BDD 的构造过程就是按照底事件集合进行选择的，选择一个底事件，就对其进行逻辑运算。当  $e_i$  的父节点的逻辑门为与门时， $e_i$  取 1，则左分支为其兄弟子树序列集合； $e_i$  取 0，则右分支兄弟子树集合就

可以消除；当  $e_i$  的父节点的逻辑门为或门时， $e_i$  取 1，则左分支兄弟子树集合就可以消除； $e_i$  取 0，则右分支为其兄弟子树序列集合；话务类问题故障树转化成 BDD 如下图 2 所示：

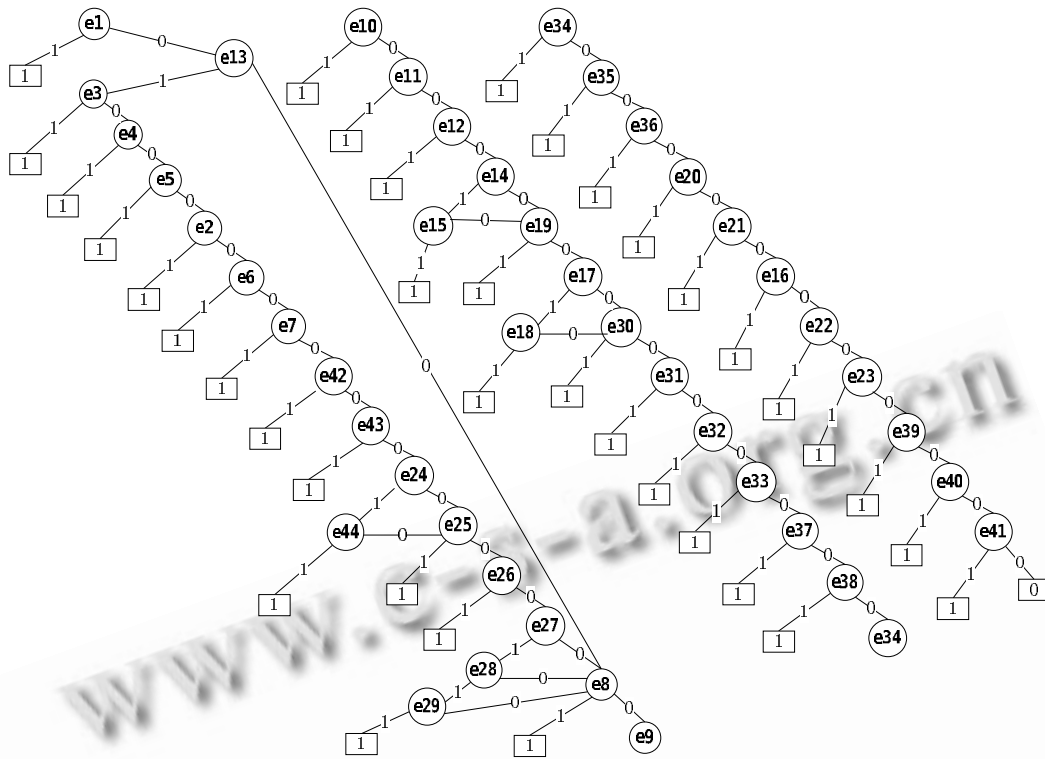


图 2 话务类问题 BDD 图

这种相邻低事件排序方式产生的节点数，比起一般的底事件排序方式有显著减少<sup>[8]</sup>。

### 3 BDD中事件重要度分析

事件重要度是指一个事件发生故障时，其对顶事件发生概率的影响程度，是一个系统中各部分或者子系统重要程度的一种数量衡量方式<sup>[9]</sup>。基本事件重要度分为三种：结构重要度、概率重要度和临界值重要度。

#### 3.1 概率重要度

概率重要度指第  $i$  个事件发生故障的变化引起顶事件故障变化的程度。假设系统包含  $n$  个独立事件， $q_i(t)$  表示部件  $i$  在  $t$  时刻的不可靠度，其中  $i=1,2,3,\dots,n$ ； $q(t)=(q_1(t), q_2(t), \dots, q_n(t))$  表示不可靠度向量；系统在  $t$  时刻的不可靠度为  $\phi(q(t))$ 。事件  $i$  在  $t$  时刻的概率重要度计算公式为：

$$I(i|t) = \frac{\partial \phi(q(t))}{\partial q_i(t)}, \text{ 且 } i=1,2,3,\dots,n$$

由全概率公式可推导得，系统不可靠度为：

$$\phi(q(t)) = (1 - q_i(t)) \cdot \phi(0_i, q(t)) + q_i(t) \cdot \phi(1_i, q(t))$$

事件  $i$  在  $t$  时刻的概率重要度计算公式为：

$$I(i|t) = \frac{\partial \phi}{\partial q_i(t)} = \phi(1_i, q(t)) - \phi(0_i, q(t))$$

对于 BDD 重要度分析，假设部件  $i$  对应 BDD 中基本事件  $e_i$ ，当  $e_i$  发生时，系统发生故障概率为：

$$G_i(q(t)) = \phi(1_i, q(t)) - \phi(0_i, q(t))$$

基本事件  $e_i$  对系统发生概率计算公式为：

$$\phi(0_i, q(t)) = \sum_n \{ p_{rxi}(q(t)) \cdot p_{oxi}^0(q(t)) \} + Z(q(t))$$

$$\phi(1_i, q(t)) = \sum_n \{ p_{rxi}(q(t)) \cdot p_{oxi}^1(q(t)) \} + Z(q(t))$$

其中  $n$  为基本事件  $e_i$  在 BDD 中的节点个数； $p_{rxi}(q(t))$  为从根节点开始到节点  $e_i$  路径概率； $p_{oxi}^1(q(t))$  为从节点  $e_i$  开始，且经  $e_i$  的分支为 1 到终点为 1 的部分路径概率； $p_{oxi}^0(q(t))$  为从节点  $e_i$  开始，且经  $e_i$  的分支为 0 到终点节点为 1 的部分路径概率； $Z(q(t))$  为从根节点开始到终点节点为 1 的路径概率，且该路径不通过节点  $e_i$ 。可推导出基本事件  $e_i$  的关键函数  $G_i(q(t))$  的计算公式变换为：

$$G_i(q(t)) = \phi(1_i, q(t)) - \phi(0_i, q(t)) = \sum_n \{p_{rx_i}(q(t))(p_{ox_i}^1(q(t)) - p_{ox_i}^0(q(t)))\}$$

3.2 结构重要度

结构重要度指事件在系统中所处位置的重要程度，结构重要度与事件本身发生的故障概率无关，只与该事件在系统中的位置有关。根据 Lambert 提出的结构重要度方法，对事件 i 可设：

$$q_i = \begin{cases} 1 & j = i \\ 1/2 & j \neq i \end{cases}$$

代入概率重要度计算公式可得：

$$I_i^\phi = G_i(q) = \phi(1_i, 1/2) - \phi(0_i, 1/2) = \sum_n \{p_{rx_i}(1/2)(p_{ox_i}^1(1/2) - p_{ox_i}^0(1/2))\}$$

3.3 临界值重要度

一般情况，减少概率大的基本事件的概率要比减

少概率小的容易，而概率重要度系数并未反映这一事实，因此，它不是从本质上反映各基本事件在事故树中的重要程度。而临界重要度系数  $C_i$  则是从敏感度和概率双重角度衡量各基本事件的重要度标准，它与概率重要度系数的关系是：

$$C_i = \frac{q_i}{F_s} I_{p(i)}^{\phi}$$

其中  $I_{p(i)}^{\phi}$  表示事件 i 概率重要度， $q_i$  表示事件 i 概率， $F_s$  表示顶事件发生概率。

4 实验结果

根据上述建立的 BDD 图和重要度计算公式，通过计算机仿真实验，取概率重要度的截断取为  $1.0 \times 10^{-9}$ ，可以求出各个底事件在某个时刻发生概率下，其结构重要度、概率重要度和临界值重要度的结果。并与利用传统求解最小割集的 FTA 方法计算机结果进行比较。结果如下表 2 所示：

表 2 重要度结果及其比较

事件	概率	结构重要度		概率重要度		临界重要度	
		BDD 法	FTA 法	BDD 法	FTA 法	BDD 法	FTA 法
e <sub>1</sub>	6.7	0.025641026	0.025641026	0.336767478	0.336767487	0.034357474	0.034357475
e <sub>2</sub>	3.6	0.025641026	0.025641026	0.321994095	0.321994179	0.017650893	0.017650898
e <sub>3</sub>	1.3	0.025641026	0.025641026	0.311105798	0.311105822	0.006158398	0.006158398
e <sub>4</sub>	0.91	0.025641026	0.025641026	0.309259478	0.309259584	0.004285295	0.004285296
e <sub>5</sub>	0.69	0.025641026	0.025641026	0.308217688	0.308217732	0.003238344	0.003238344
e <sub>6</sub>	1.6	0.025641026	0.025641026	0.312525689	0.312525601	0.00761416	0.007614158
e <sub>7</sub>	8.1	0.025641026	0.025641026	0.343518801	0.343518911	0.042369351	0.042369364
e <sub>8</sub>	5.8	0.025641026	0.025641026	0.332457975	0.332457996	0.029361689	0.029361691
e <sub>9</sub>	4.5	0.025641026	0.025641026	0.326265289	0.326265312	0.022356286	0.022356288
e <sub>10</sub>	4.4	0.025641026	0.025641026	0.325790211	0.325790203	0.02182765	0.02182765
e <sub>11</sub>	7.4	0.025641026	0.025641026	0.340135165	0.340135145	0.038326533	0.038326531
e <sub>12</sub>	5.7	0.025641026	0.025641026	0.331980447	0.331980408	0.028814006	0.028814003
e <sub>13</sub>	1.5	0.025641026	0.025641026	0.312052324	0.312052368	0.007127463	0.007127464
e <sub>14</sub>	1.1	0.012820512	0.012820512	0.024734960	0.024734973	0.000414305	0.000414305
e <sub>15</sub>	8.1	0.012820512	0.012820512	0.003359087	0.003359071	0.000414307	0.000414305
e <sub>16</sub>	0.8	0.025641026	0.025641026	0.308738629	0.308738698	0.003760947	0.003760948
e <sub>17</sub>	0.5	0.012820512	0.012820512	0.002897254	0.002897215	0.0000220583	0.000022058
e <sub>18</sub>	0.95	0.012820512	0.012820512	0.001524845	0.001524850	0.000022058	0.000014486
.....							

(下转第 163 页)

通过比较,发现利用 BDD 法,建立起来的 BDD 图,所求结果比起 FTA 法所求结果几乎相同。

## 5 结论与展望

本文主要讨论了,利用 FTA 方法建立话务类故障模型,通过分析各基本事件发生概率,及其重要度,求出顶事件发生概率。相反,顶事件发生,可以通过基本事件的重要度,结合基本事件发生概率,判断故障的原因。所以采用 BDD 法,其得到的基本事件重要度分析和顶事件概率结果与 FTA 法比较,几乎接近,其 BDD 图法,编程实现容易,计算效率高。但是在故障树转化成 BDD 图时,事件的排序对其重要度有影响,如何采用排序策略,使基本事件重要度更加精确仍然是个热点。实践证明相邻低事件优先排序法,在话务类故障中,同样能有效减少 BDD 节点数量。

### 参考文献

- 1 Rauzy A. New Algorithm for Fault Tree Analysis. Reliability Engineering and System Safety 40, 1993:203-211.

- 2 朱继洲.故障树原理和应用.西安:西安交通大学出版社,1989,12-25.
- 3 Dutuit Y. A Linear-time Algorithm to Find Modules of FaultTree. IEEE Transactionson on Reliability, 1996, 45(3):423-425.
- 4 Bryant R. Graph-based Algorithm for Boolean Function Manipulation. IEEE Transactions On Computers, 1996, 35(8):677-691.
- 5 戴云微,韩之俊,朱海荣.故障模式及影响分析(FMEA)研究进展.质量安全管理,2007,10(9):23-25.
- 6 Sinnamon RM, Andrews JD. New approaches to evaluating fault trees. Quality and Reliability Engineering International,1997,58:89-96.
- 7 周经伦,孙权.一种故障树分析的新算法.模糊系统与数学,1997,11(3):74-78.
- 8 孙燕,杜素果.一种二元决策图底事件排序的新方法.系统管理学报,2008,17(2):210-220.
- 9 徐享成,张建国.基于 BDD 技术下的故障树重要度分析.电子机械工程.2003,19(6):16-20.