第 37 卷 第 1 期
2022 年 2 月

天津科技大学学报
Journal of Tianjin University of Science & Technology

Vol. 37 No. 1
Feb. 2022

DOI:10.13364/j.issn.1672-6510.20210169

基于不确定故障树的软件可靠性分析方法

刘 颖，杨芸芸
（天津科技大学人工智能学院，天津 300457）

摘 要：故障树是用来评估系统可靠性或风险的有效方法之一，其中基本事件故障发生概率通常为常数或随机变量. 在实际软件可靠性测试过程中，可得到的故障数据量非常少且存在大量的认知不确定性，因此使用传统故障树评估方法是不合理的. 为此，本文提出基于不确定故障树的软件可靠性分析方法. 故障树中基本事件发生故障的信度用不确定测度表示，以不确定环境下“与门”和“或门”的运算法则计算系统的可靠度. 最后，对某装备的软件系统进行可靠性分析及灵敏度分析，使该方法的合理性得以验证.

关键词：认知不确定性；不确定故障树；软件可靠性评估；不确定测度
中图分类号：TP311.5 **文献标志码**：A **文章编号**：1672-6510(2022)01-0052-04

Software Reliability Analysis Method Based on Uncertain Fault Tree

LIU Ying, YANG Yunyun
(College of Artificial Intelligence, Tianjin University of Science & Technology, Tianjin 300457, China)

Abstract: The fault tree is one of the effective methods used to evaluate the reliability or risk of the system, in which the probability of occurrence of each event failure is usually a constant or random variable. In the actual software reliability testing process, the amount of available fault data is very small and there is a large amount of epistemic uncertainty. Therefore, it is unreasonable to use the traditional fault tree evaluation method. For this reason, in this article we propose a software reliability analysis method based on uncertain fault trees. The belief degree of basic events failure in the fault tree is first expressed by uncertainty measure, and the reliability of the system is then calculated by the algorithm of “AND” and “OR” gates in an uncertain environment. Finally, the reliability analysis and sensitivity analysis of the software system of a certain equipment are conducted to verify the rationality of the method.

Key words: epistemic uncertainty; uncertain fault tree; software reliability evaluation; uncertainty measure

软件可靠性分析是软件可靠性工程的核心内容之一. 软件失效不仅会造成硬件系统瘫痪和经济损失，还会引起重大事故，如美国长途电话业务曾因软件故障引发系统瘫痪，火箭发动机控制系统软件故障可能会造成飞行试验失败，飞机的飞行控制软件的失效会危及人身安全. 因此，科学的软件可靠性评估是非常必要的. 软件可靠性指在一定的条件下和时间内，软件运行不会发生故障的概率[1]. 故障树是常用的软件可靠性评估方法，该方法适用于软件生命周期的各阶段. 刘文红等[2]针对软件测试用例设计不充分问题，提出一种通过故障树分析设计测试用例的新方法. 以上文献均采用传统故障树分析方法评估软件的可靠性，即假设基本事件故障概率是常数或随机变量，但概率假设只能用来评估随机不确定性.

在软件可靠性测试过程中，受到软件内部结构的复杂性、人的认知水平等因素的影响，存在许多认知不确定性因素，因而无法采用传统软件可靠性评估. 针对不精确的信息，专家学者开始引入模糊理论[3]对其进行描述. 蔡开元[4]提出了基于模糊理论的软件可靠性增长模型，该模型不仅适用于测试阶段，而且适用于确认阶段. 焦彦平等[5]针对软件开发初期信息不完全的问题，基于模糊集理论建立软件缺陷评

收稿日期：2021-08-21；**修回日期**：2021-10-04
基金项目：天津市高等学校基本科研业务费一般项目(2019KJ233)
作者简介：刘 颖（1982—），女(回)，天津人，教授，liu@tust.edu.cn

估模型. 谭海中[6]针对传统通信软件可靠性测试计算结果不精确的问题，提出了采用模糊故障树的可靠性测试方法. 但由于可能性测度不满足自对偶性，导致评估结果经常产生相互矛盾的结论.

不确定理论[7]是基于规范性、自对偶性、次可加性和乘积公理体系的数学分支，主要用于对认知不确定性建模. 近年来，不确定理论在许多领域已经有了广泛应用，如系统可靠性建模[8]、结构可靠性评估[9]、图论[10]、不确定统计[11]、时间序列模型[12]等.

本文在认知不确定性环境下，提出基于不确定故障树的软件可靠性分析方法，并用于实际装备软件的可靠性评估.

1　基本概念

1.1　不确定理论

令 Γ 为非空集合，L 是 Γ 上的子集构成的 σ –代数，L 中的每一个元素 Λ 称为事件. 每个事件 Λ 发生的信度记作 $M\{\Lambda\}$，其中信度 M 需要满足以下 4 条公理：

公理 1（规范性）对于全集 Γ，有

$$M\{\Gamma\}=1$$

公理 2（自对偶性）对任意事件 $\Lambda\in L$，有

$$M\{\Lambda\}+M\{\Lambda^c\}=1$$

公理 3（次可加性）对任意的可列可数事件列 $\Lambda_1,\Lambda_2,\cdots$，有

$$M\left\{\bigcup_{i=1}^{\infty}\Lambda_i\right\}\leqslant\sum_{i=1}^{\infty}M\{\Lambda_i\}$$

公理 4（乘积公理）令 (Γ_k,Λ_k,M_k) 为不确定空间，$k=1,2,\cdots$，则乘积不确定测度满足

$$M\left\{\prod_{k=1}^{\infty}\Lambda_k\right\}=\mathop{\wedge}_{k=1}^{\infty}M_k\{\Lambda_k\}$$

定义 1 若集函数满足规范性、自对偶性、次可加性和乘积公理，则称其为非空集合上的不确定测度.

定义 2 令 Γ 是一个非空集合，L 是一个在 Γ 上的 σ –代数，M 是一个不确定测度，则称三元组 (Γ,L,M) 为一个不确定空间.

定义 3 不确定变量是一个从不确定空间 (Γ,L,M) 到实数集的函数 ξ，$\{\xi\in B\}$ 是一个在任意 Borel 集上的事件.

定义 4 对任意 Borel 集的实数 $B_1,B_2,\cdots,B_n$，若满足

$$M\left\{\bigcap_{i=1}^{n}(\xi_i\in B_i)\right\}=\mathop{\wedge}_{i=1}^{n}M\{\xi_i\in B_i\}$$

则不确定变量 $\xi_1,\xi_2,\cdots,\xi_n$ 是相互独立的.

定理 1 对任意 Borel 集实数 $B_1,B_2,\cdots,B_n$，若满足

$$M\left\{\bigcup_{i=1}^{n}(\xi_i\in B_i)\right\}=\mathop{\vee}_{i=1}^{n}M\{\xi_i\in B_i\}$$

则不确定变量 $\xi_1,\xi_2,\cdots,\xi_n$ 是相互独立的.

定义 5 一个不确定集合是一个从不确定空间 (Γ,L,M) 映射到实数集合的函数 ξ，$\{B\subset\xi\}$ 和 $\{\xi\subset B\}$ 是对于任意 Borel 集的事件.

定理 2 令 ξ 和 η 为不确定集合，则

$$\xi\cup\eta=\eta\cup\xi$$

$$\xi\cap\eta=\eta\cap\xi$$

定理 3 令 ξ,η,τ 为不确定集合，则

$$(\xi\cup\eta)\cup\tau=\xi\cup(\eta\cup\tau)$$

$$(\xi\cap\eta)\cap\tau=\xi\cap(\eta\cap\tau)$$

定理 4 令 ξ,η,τ 为不确定集合，则

$$\xi\cup(\eta\cap\tau)=(\xi\cup\eta)\cap(\xi\cup\tau)$$

$$\xi\cap(\eta\cup\tau)=(\xi\cap\eta)\cup(\xi\cap\tau)$$

定理 5 令 ξ,η 为不确定集合，则

$$\xi\cup(\xi\cap\eta)=\xi$$

$$\xi\cap(\xi\cup\eta)=\xi$$

1.2　不确定故障树分析法

故障树可以用来对软件风险或可靠性进行定量评估，并分析引起软件产品发生故障的原因，包括软件产品组成部分、软件使用环境、人为因素等，适用于大型复杂系统.

故障树分析法遵循自顶向下原则. 分析系统的故障模式时，将故障模式作为输出事件（顶事件），用“中间事件”表示导致该事件发生的原因，并用逻辑门（“与门”“或门”）连接. 以此类推，直到找到顶事件发生的根本原因，即输入事件（基本事件）.

定义 6 在故障树中，若所有基本事件发生的信度均用不确定测度评估，则该故障树称为不确定故障树.

定理 6 若输入事件 $\Lambda_1,\Lambda_2,\cdots,\Lambda_n$ 相互独立且由“与门”连接，则输出事件 Λ 发生的信度为

$$M\{\Lambda\}=\mathop{\wedge}_{i=1}^{n}M\{\Lambda_i\}$$

若输入事件 $\Lambda_1,\Lambda_2,\cdots,\Lambda_n$ 相互独立且由“或门”连接，则输出事件 Λ 发生的信度为

$$M\{\Lambda\}=\mathop{\vee}_{i=1}^{n}M\{\Lambda_i\}$$

其中：∨为最大值运算符，∧为最小值运算符.

2　本文方法的流程

(1)根据软件系统控制流程图，建立故障树. 基本事件是软件各最小模块单元，各最小模块单元分别具有对应的代码块，中间事件为控制流程中间子程序模块，顶事件为软件故障. 在可靠性测试过程中，以最小模块单元的代码块的失效数据为依据，进行可靠性评估.

(2)获取基本事件发生的不确定测度. 在没有足够测试数据的情况下，邀请软件工程师给出基本事件发生的不确定测度.

(3)由树形结构给出顶事件的表达式. 将顶事件用基本事件表示，若存在不相互独立的中间事件，利用不确定集合的运算法则化为相互独立的基本事件.

(4)进行软件可靠性评估. 利用不确定理论的运算法则，结合故障树，对软件可靠性进行评估.

(5)进行敏感度分析. 分析故障树中基本事件的敏感度. 在进行测试用例设计时，考虑各模块单元对系统失效的影响程度，重点测试关键模块.

利用不确定故障树分析法的基本步骤，为软件生命周期中的各个阶段建立失效模型，计算软件失效的可能性，通过分析结果提出保障意见，从而提高软件安全性.

3　某装备软件可靠性分析

对某装备软件系统进行实例分析. 该软件的功能是实时监测某飞行控制系统的传感器，在飞行控制系统的正常工作过程中有重要作用. 由于软件的使用环境等原因，没有大量的故障测试数据，无法使用概率方法，因此利用本文的方法，为该装备软件进行可靠性评估.

(1)对某装备软件的控制流程进行分析，软件故障是顶事件，控制流程中间的子程序模块故障是中间事件，软件各最小模块单元是基本事件，各最小模块单元分别具有对应的代码块. 建立如图 1 所示的不确定故障树[12]，其中 $\Lambda_1,\Lambda_2,\cdots,\Lambda_n$ 为相互独立的基本事件，各最小模块单元所包含的代码块行数见表 1，$E_j(j=1,2,\cdots)$ 为中间事件，Λ 是顶事件.

(2)由于软件测试中的故障数据量较少，输入事件(每个代码块中发生故障的信度)是由专家评估得出(表 2).

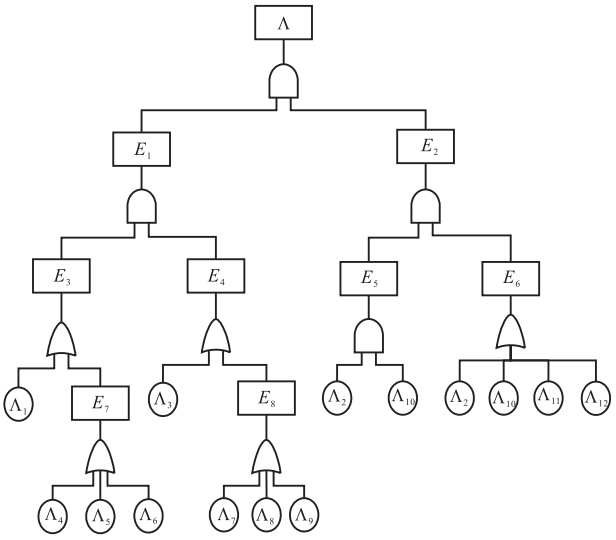

图 1　某装备软件的故障树示意图

Fig. 1　Fault tree diagram of a certain equipment software

表 1　基本事件对应的代码块行数

Tab. 1　Numbers of lines of code corresponding to the basic event

基本事件	代码块行数	基本事件	代码块行数
Λ_1	2 378	Λ_7	1 838
Λ_2	5 525	Λ_8	1 552
Λ_3	8 462	Λ_9	1 286
Λ_4	1 824	Λ_{10}	2 329
Λ_5	1 156	Λ_{11}	1 089
Λ_6	1 133	Λ_{12}	3 005

表 2　基本事件故障信度数据

Tab. 2　Failure belief degrees data of the basic events

基本事件	信度	基本事件	信度
Λ_1	0.045 1	Λ_7	0.023 3
Λ_2	0.034 3	Λ_8	0.041 7
Λ_3	0.026 8	Λ_9	0.067 7
Λ_4	0.034 6	Λ_{10}	0.042 8
Λ_5	0.047 2	Λ_{11}	0.063 5
Λ_6	0.042 5	Λ_{12}	0.051 7

(3)由图 1 中的树形结构，结合定理 2、定理 3、定理 4 和定理 5，从而得到输出事件为

$$\begin{aligned}\Lambda = E_1 \cap E_2 &= (E_3 \cap E_4) \cap (E_5 \cap E_6) = \\ &\left[(\Lambda_1 \cup E_7) \cap (\Lambda_3 \cup E_8)\right] \cap \\ &\left[(\Lambda_2 \cap \Lambda_{10}) \cap \right. \\ &\left.(\Lambda_2 \cup \Lambda_{10} \cup \Lambda_{11} \cup \Lambda_{12})\right] = \\ &\left[\left(\Lambda_1 \cup (\Lambda_4 \cup \Lambda_5 \cup \Lambda_6)\right) \cap \right. \\ &\left.\left(\Lambda_3 \cup (\Lambda_7 \cup \Lambda_8 \cup \Lambda_9)\right)\right] \cap \\ &(\Lambda_2 \cap \Lambda_{10}) = \left(\Lambda_1 \cup (\Lambda_4 \cup \Lambda_5 \cup \Lambda_6)\right) \cap\end{aligned}$$

$$(\Lambda_3 \cup (\Lambda_7 \cup \Lambda_8 \cup \Lambda_9)) \cap (\Lambda_2 \cap \Lambda_{10})$$

(4) 根据定义 4、定理 1 得出系统的故障信度为

$$\begin{aligned} M\{\Lambda\} = {} & M\{(\Lambda_1 \cup (\Lambda_4 \cup \Lambda_5 \cup \Lambda_6)) \cap \\ & (\Lambda_3 \cup (\Lambda_7 \cup \Lambda_8 \cup \Lambda_9)) \cap (\Lambda_2 \cap \Lambda_{10})\} = \\ & (M\{\Lambda_1\} \vee (M\{\Lambda_4\} \vee M\{\Lambda_5\} \vee M\{\Lambda_6\})) \wedge \\ & (M\{\Lambda_3\} \vee (M\{\Lambda_7\} \vee M\{\Lambda_8\} \vee M\{\Lambda_9\})) \wedge \\ & M\{\Lambda_2\} \wedge M\{\Lambda_{10}\} = \\ & (0.045\,1 \vee (0.034\,6 \vee 0.047\,2 \vee 0.042\,5)) \wedge \\ & (0.026\,8 \vee (0.023\,3 \vee 0.041\,7 \vee 0.067\,7)) \wedge \\ & 0.034\,3 \wedge 0.042\,8 = 0.034\,3 \end{aligned}$$

故系统的可靠度为

$$R = 1 - M\{\Lambda\} = 1 - 0.034\,3 = 0.965\,7$$

(5) 在对软件系统进行可靠性评价时，由于基本事件 $\Lambda_1, \Lambda_2, \cdots, \Lambda_{10}$ 对顶事件的影响程度不同，因此采用 Sobol 灵敏度分析方法对基本事件 $\Lambda_1, \Lambda_2, \cdots, \Lambda_{10}$ 进行敏感度分析并进行评价，见表 3.

表 3　基本事件灵敏度对比表

Tab. 3　Comparison table of basic event sensitivity

基本事件	基本事件灵敏度	
	传统故障树	不确定故障树
Λ_1	3.15×10^{-21}	1.27×10^{-7}
Λ_2	1.49×10^{-20}	3.26×10^{-6}
Λ_3	4.01×10^{-21}	2.03×10^{-7}
Λ_4	3.51×10^{-21}	1.29×10^{-7}
Λ_5	3.38×10^{-21}	5.46×10^{-8}
Λ_6	3.65×10^{-21}	1.26×10^{-7}
Λ_7	4.19×10^{-21}	1.67×10^{-7}
Λ_8	3.67×10^{-21}	1.53×10^{-7}
Λ_9	3.85×10^{-21}	1.42×10^{-7}
Λ_{10}	1.24×10^{-20}	2.67×10^{-6}

从表 3 可以看出，基于本文提出的方法得到的基本事件灵敏度值均高于基于传统故障树的基本事件灵敏度值，代码块行数分别为 5 525 和 2 329 的基本事件 Λ_2 和 Λ_{10} 的灵敏度值最大. 因此，基本事件的高灵敏度证明了基于不确定故障树的软件可靠性评估方法可以对软件可靠性进行精准测试. 下面具体分析敏感参数对软件系统可靠性的影响情况.

图 2 为基本事件 Λ_2 和 Λ_{10} 的变化对软件系统可靠度的影响情况示意图. 在 (0，0.043) 内，软件产品可靠度随着基本事件 Λ_2 发生失效的不确定测度的增大而减小，之后产品失效的不确定测度不再变化. 在 (0，0.034 3) 内，系统可靠度随 Λ_{10} 的增大而减小；在 0.034 3 以后，系统可靠度则不再变化.

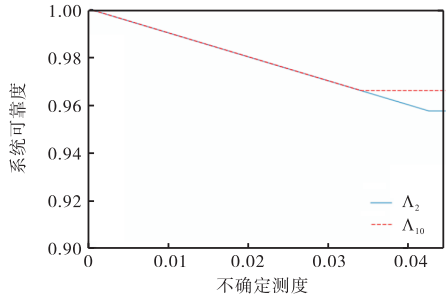

图 2　基本事件 Λ_2 和 Λ_{10} 对产品影响情况示意图

Fig. 2　Schematic diagram of impact of basic events Λ_2 and Λ_{10} on products

综上所述，应该重点关注代码块行数为 5 525 和 2 329 的代码块所对应的最小模块单元，因为这两个最小模块单元对产品可靠度影响较大，是测试的重点.

4　结　语

在软件可靠性分析过程中，由于没有充足的故障数据且存在大量的认知不确定性，基于传统概率论的故障树分析方法不再适用，故本文针对传统故障树对软件可靠性分析不足的问题，提出不确定故障树软件可靠性分析方法，主要创新内容：

(1) 最小模块单元的代码块中发生故障的信度用不确定测度评估.

(2) 在不确定故障树中，采用不确定集合运算法则，可将不相互独立的中间事件转化为相互独立的基本事件.

(3) 对基本事件进行敏感度分析，找出影响软件可靠性的关键因素.

参考文献：

[1] MICHAEL R L. Handbook of software reliability engineering[M]. New York：Mcgraw-Hill，1996.

[2] 刘文红，王占武，吴欣. 故障树分析技术在软件测试中的应用[J]. 系统工程与电子技术，2004，26(7)：985–987.

[3] ZADEH L A. Fuzzy sets[J]. Information & control，1965，8(3)：338–353.

[4] 蔡开元. 一个模糊软件可靠性确认模型[J]. 航空学报，1993(11)：653–656.

[5] 焦彦平，李建华，李唱. 一种基于模糊集理论的软件缺陷评估方法[J]. 装备学院学报，2012，23(4)：87–91.

[6] 谭海中. 基于模糊故障树的通信软件可靠性测试方法

（下转第 63 页）

[10] MANUEL L, BELEN C, ANTONIO S, et al. Conditional variational autoencoder for prediction and feature recovery applied to intrusion detection in IoT[J]. Sensors, 2017, 17(9): 1967.
[11] LOUIZOS C, SWERSKY K, LI Y, et al. The variational fair autoencoder[EB/OL]. [2021-03-30]. http://arxiv.org/abs/1511.00830V1.
[12] VINCENT P, LAROCHELLE H, BENGIO Y, et al. Extracting and composing robust features with denoising autoencoders[C]// Machine Learning, Proceedings of the Twenty-Fifth International Conference(ICML 2008). Amsterdam: Elsevier, 2008.
[13] KINGMA D P, CHEN X, SALIMANS T, et al. Variational lossy autoencoder[EB/OL]. [2021-03-30]. http://arxiv.org/pdf/1611.02731.PDF.
[14] VAN DEN OORD A, NAL K, KORAY K. Pixel recurrent neural networks[EB/OL]. [2021-03-30]. https://arxiv.org/abs/1601.06759.
[15] MIRZA M, OSINDERO S. Conditional generative adversarial nets[EB/OL]. [2021-03-30]. https://arxiv.org/abs/1411.1784.
[16] LARSEN A B L, SØNDERBY S K, WINTHER O. Autoencoding beyond pixels using a learned similarity metric[EB/OL]. [2021-03-30]. https://arxiv.org/abs/1512.09300.
[17] DUMOULIN V, BELGHAZI I, POOLE B. Adversarially learned inference[EB/OL]. [2021-03-30]. http://arxiv.org/abs/1606.00704.
[18] HUANG H, LI Z, HE R, et al. IntroVAE: introspective variational autoencoders for photographic image synthesis[EB/OL]. [2021-03-30]. http://arxiv.org/pdf/1807.06358.pdf.
[19] REZENDE D J, MOHAMED S, WIERSTRA D. Stochastic backpropagation and approximate inference in deep generative models[EB/OL]. [2021-03-30]. http://www.arxiv.org/pdf/1401.4082.pdf.

责任编辑：郎婧

（上接第 55 页）

研究[J]. 信息通信, 2020(12): 152-153.
[7] LIU B D. Uncertainty theory[M]. Berlin: Springer-Verlag, 2007.
[8] LIU Y, QU Z G, LI X Z, et al. Reliability modelling for repairable systems with stochastic lifetimes and uncertain repair times[J]. IEEE Transactions on fuzzy systems, 2019, 27(12): 2396-2405.
[9] LIU Y, ZHAO J Y, QU Z G, et al. Structural reliability assessment based on subjective uncertainty[EB/OL]. [2021-08-20]. https://doi.org/10.1142/S0219876221500468.
[10] GAO Y, YANG L, LI S, et al. On distribution function of the diameter in uncertain graph[J]. Information sciences: an international journal, 2015, 296: 61-74.
[11] YAO K, LIU B D. Uncertain regression analysis: an approach for imprecise observations[J]. Soft computing, 2018, 22: 5579-5582.
[12] YANG X F, LIU B D. Uncertain time series analysis with imprecise observations[J]. Fuzzy optimization & decision making, 2018, 18: 263-278.
[13] 刘博宁, 张鹏, 张建业, 等. 软件可靠度的模糊故障树评定方法[J]. 计算机应用研究, 2012, 29(10): 3783-3786.

责任编辑：郎婧