

# 改进的多版本软件扩展表决系统

张浥楠<sup>1</sup> 尹乾<sup>1\*</sup> 郭平<sup>1,2</sup>

(1. 北京师范大学信息科学与技术学院, 北京 100875; 2. 北京理工大学计算机科学与技术学院, 北京 100081)

**摘要:** 本文提出了一种改进的扩展表决系统, 该系统是利用自动分析表决系统可靠性等参数的工具来构建的。在软件容错中采用 N-模块冗余 (N-Modular Redundancy-NMR) 及 N-版本编程 (N-version Programming), 应用表决技术, 可以屏蔽整个软件系统中出错的部分。在实践中直接应用理论的表决模型会发现很难得到理论分析的结果。针对这样的问题, 本文提出一种基于多数 (plurality) 的扩展表决系统: 使用构造决策树的方法和进行 DFT 遍历作为自动化的计算工具; 通过对基本的多数表决系统进行模型扩展, 加入指导性规则对决策树进行重构, 以自动计算工具的结果作为参考, 从而选择更加可靠的表决结果作为输出。最终的模拟测试表明, 所提出的扩展表决系统能够提高系统输出的正确率。

**关键词:** 多版本软件; 表决系统; 容错; 决策树

**中图分类号:** TP311.5

## 引言

表决技术被广泛应用在很多领域中, 比如: 分布式系统的人工排除, 当集群工作站在因为网络问题等断开联系时来控制升级过程; 结合分类器而广泛的应用于模式识别领域中<sup>[1]</sup>。在多版本软件系统中, 可以将每一版本看成一个专家, 通过容错配合多个专家系统<sup>[2-3]</sup>进行共同协作解决问题以提高可靠性。因此, 对表决机制进行理论上和测试上的分析是具有重要意义的。在特定的数学模型下, 从理论上分析表决系统能够帮助理解表决系统的预期行为<sup>[4-5]</sup>。在基本表决结构中所有专家都需要参与表决过程。

理论上虽然能够预计出表决系统的行为特性, 然而在实际中却很难直接应用, 这是由于数学模型过于理想化和缺乏一些灵活性。为解决这些问题, 本文提出了一种改进的扩展表决系统, 对基本表决结构进行了扩展, 加入了一些模块和规则, 以提高表决系统的输出正确率, 从而在实际应用中可以更灵活的配置表决系统。

本文包括以下内容: 首先根据文献[6]中的方法构造一种自动化参数计算工具; 之后将此计算工具应用到本文的扩展表决系统中; 最后再对本文中的扩展表决系统进行性能评价及得出结论。

## 1 度量参数及定义

**定义 1:** 专家节点。表决系统是用来处理多个专家系统的输出数据的。专家节点有时可以等价于其他领域的概念, 如模式识别中的分类器。专家系统的输出结果有三种情况: 1) 正确, 输出的结果经过验证与现实相符; 2) 错误, 输出的结果经过验证与现实不符; 3) 失效, 由于硬件问题或软件问题导致专家系统无法输出数据结果时, 此情况为专家系统的输出失效。

**定义 2:** 专家的个数 ( $N$ )。记  $N$  为表决系统中所有专家的总数。

**定义 3:** 专家节点输出的值及专家节点输出的值集合大小。因为专家系统会输出多个数据值, 所以, 记  $K$  为专家输出的这些值的集合大小, 记  $S_i$  为专家输出的第  $i$  个值 ( $i \leq K$ )。

**定义 4:** 专家节点输出的分布  $PMF$ 。专家节点输出的值可能是正态分布或者均匀分布等常见分布。

**定义 5:** 专家输出一个值是正确的概率  $R$ 。在实际中, 专家节点是可能出现错误的结果, 如果专家系统总是出错, 那么这个专家的可靠性为 0, 如果总

收稿日期: 2006-04-29

基金项目: 国家自然科学基金 (60675011)

第一作者: 男, 1982 年生, 硕士生

\*通讯联系人

E-mail: yinqian @bnu.edu.cn

是正确的,那么这个专家的可靠性是 1。

定义 6:多数(Plurality)表决。多数表决可以用如下规则表达:

IF ( $k$  个专家同意某个结果) & & ! ( $j$  个专家同意某个结果且  $j < k$ )

得到一致表决结果

ELSE

表决失败,没有实现多数表决

定义 7:表决系统的表决正确率  $R$ 。可以通过一段时间的统计得到表决系统的输出结果符合现实结果的比率。记  $R = \text{表决正确的次数} / \text{总共表决的次数}$ 。

定义 8:出现表决结果的概率  $P_c$ 。表决系统在某些情况下是无法得出表决结果的,如 5 个专家中,其中 1 个失效,出现 2 票对 2 票时就没有表决结果。 $P_c = \text{出现表决结果的次数} / \text{总共表决的次数}$ 。

定义 9:决策树。根据上面定义中  $N$ ,  $K$  及每个专家的 PMF <sub>$i$</sub>  可以生成一棵决策树。这棵树可以搜索出所有的可能表决结果。

## 2 扩展表决系统建立

相对他人提出的表决系统<sup>[7-10]</sup>,本文提出的扩展表决系统的特点是:1)通过对决策树的重构和对比,使用阻塞操作来屏蔽一些专家(即某次表决中排除特定专家),来实现较优输出;2)加入一些记录/统计模块,对于某些出错情况的再现有了较好的解决。对扩展表决系统进行性能评价主要的方面有:1)在模拟实验中,表决准确率是否有提;2)是否在一定程度上提高了表决系统的灵活性。

### 2.1 决策树的重构

假设:表决系统中前端的各个专家系统输出的都是 0/1 的二值决策,此时有 3 个专家节点与一个表决系统共同工作。若此过程是人为阻塞 3 个专家中的 1 个专家,则 3 个专家中只有 2 个能工作并输出结果参与表决,从而对系统出现表决结果的概率出现了影响:{3 专家,100%出现结果}变为{2 专家,50%出现结果};同时也对系统的可靠性出现了影响<sup>[6]</sup>。

此时决策树由图 1 变为图 2,称此过程为决策树的重构。

### 2.2 扩展表决系统的结构理论推理

前文中提到,理论化的模型具有过于理想化和不灵活的缺点,如:在实际中应用标准表决流程,如

果某些专家的连续输出结果的错误率超过某一个阈值时,会降低整个系统的可靠性,降低正确结果在决策树中的比重;某些专家大部分情况都没有输出结果可以看作失效时,却不把这些失效专家从系统中排除。

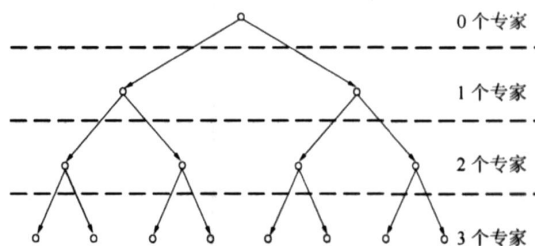


图 1 三个专家决策树

Fig. 1 Decision tree with 3 experts

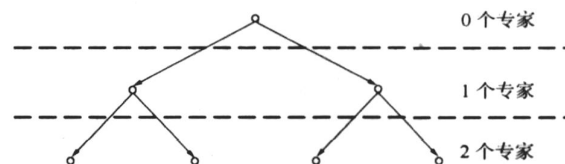


图 2 两个专家决策树

Fig. 2 Decision tree with 2 experts

而在本文提出的决策树重构过程中,专家节点数目的变化会直接影响生成表决结果的概率和表决系统的可靠性。关于这些参数间的相互联系,在文献<sup>[6]</sup>的结论中提到,如果表决系统作为输出决策的仲裁机制,那么在具有  $N$  个专家的系统中加入具有高性能的专家,成为  $N+1$  个专家,那么对于整体系统的性能是有提高的。那么反过来考虑,如果系统中加入了可靠性低的专家,则对于整个系统的可靠性是降低的。假设有  $N$  个专家和 1 个已知性能很差的专家,那么把  $N+1$  个专家中那个性能很差的专家去掉,剩下的  $N$  个专家会具有更高的可靠性。基于此推理,本文采取对应的策略加入到基本的表决系统中。为了实现这个策略,采用对专家进行评估的方法,动态的决定哪些专家参与表决。

在基本表决结构中,表决的过程都是按照前面所描述的定义严格执行,所以缺乏灵活性和难以克服其内在的一些问题。为了解决这些问题,在下面引入扩展表决系统。

### 2.3 扩展表决系统

图 3 为本文提出的扩展表决结构图,灰色是表明在某些时刻,如果专家 2 参与表决使得表决结果的错误率增加,那么就阻塞专家 2,使之不进入表决

过程。当然允许同时阻塞多个错误率高的专家。

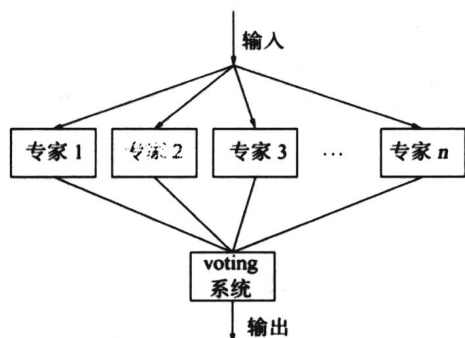


图3 扩展表决系统结构

Fig. 3 Structure of extended voting system

为了实现合理阻塞性能较差专家的目的,实现较优性能,本文对基本表决系统进行了如下扩展。

**2.3.1 自动计算度量参数的工具** 文献[6]中提出可以自动计算表决系统相关的度量参数及定义中所涉及的各项数据指标。该自动计算工具的功能是自动计算表决系统所需的各项有用参数,需要的时间复杂度和空间需求与树的深度优先遍历相同。

**2.3.2 本文的扩展表决系统结构及其规则** 本文提出的扩展表决系统框架如图4所示。

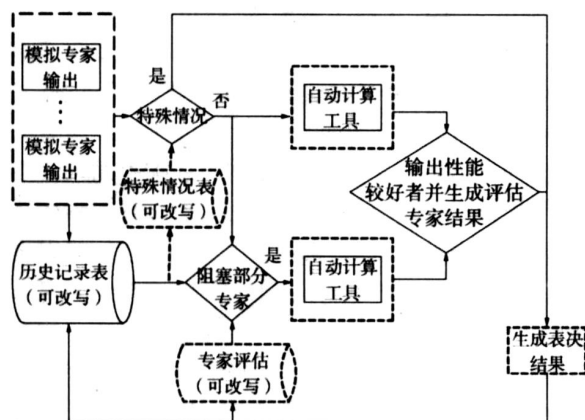


图4 扩展表决系统框架

Fig. 4 Framework of extended voting system

该扩展表决系统中包含的辅助结构为

a) 历史记录表。在表决系统中加入历史记录表,记录过去的表决情况(包括所有专家的输出结果,表决系统表决的结果,现实中真正的结果)和专家系统的输入情况。对专家系统的输入及表决系统接受的数据进行统计分析,这样能够为调节表决系统提供参考依据。

b) 特殊情况历史表。由于固定的机制是不改变的,如果在某些大多数专家的输出是错误的情况下,

生成了表决结果,那么在同样的专家系统中再次发生类似情况的话,仍然会表决出错误的结果。对上述特殊情况进行记录,构成特殊情况历史表。通过添加特殊情况表,对特殊情况进行分别计数,保留出错次数极高的情况,可以在重复出现同样情况时作为参考。

c) 专家评估表。首先对专家节点的结果进行计数,包括错误、正确、失效这三种情况(参见定义1),之后通过计算正确率得到专家的评估。

该扩展表决系统中包含的规则为

a) 特殊情况判断。待表决数据产生后,首先进入特殊情况历史表进行对比,如果在该表中存在此情况,则直接输出结果。

b) 阻塞专家判断。通过对专家评估表和历史记录表进行参照,来决定是否阻塞及如何阻塞。

c) 输出性能较好者。把两个自动化计算工具的结果进行对比,按照较优性能的表决方法进行输出。如果两个自动化计算工具的结果(决策树)相同,说明全部专家都参与了表决;反之,说明有的专家被阻塞。

d) 将所有的辅助结构设置为空,即为标准的理论模型。

### 3 实验研究

为了检验扩展的表决系统是否具有更优良的性能,我们分别使用基本表决系统和扩展表决系统进行实验,对专家系统所有可能的输入数据集合进行处理,并进行了性能对比,结果表明本文提出的扩展表决系统比基本表决系统有很大改善。

实验中各项设定为

(1) 对专家随机设定失效概率。80%的专家失效率在1%~10%,20%的专家失效率在1%~20%。

(2) 对专家随机设定错误率。80%的专家错误率在1%~20%,20%的专家错误率在1%~40%。

(3) 特殊历史记录表的长度不超过专家系统输入域的1%。

(4) 专家累计正确率低于80%时进行阻塞操作;

(5) 相同的专家数目都是同一组专家,对同一组专家使用两种表决方法。而采取相同表决系统的不同组的多个专家表决正确率统计结果是没有对比意义的。

(6) 以上所用随机数均为均匀分布随机数。  
实验结果如表 1~3 所示。

表 1 实验结果 ( $k=2$ )  
Table 1 Simulation result ( $k=2$ )

专家节点 数目 ( $N$ )	表决正确率 ( $R$ ) / %		表决失败概率 ( $1 - P_c$ ) / %	
	标准	扩展	基本	扩展
4	84.286	90.929	6.357	7.071
5	85.214	91.827	10.285	5.857
6	90.857	94.643	5.286	3.357
7	92.214	94.786	2.101	2.714

表 2 实验结果 ( $k=3$ )  
Table 2 Simulation result ( $k=3$ )

专家节点 数目 ( $N$ )	表决正确率 ( $R$ ) / %		表决失败概率 ( $1 - P_c$ ) / %	
	标准	扩展	基本	扩展
4	90.947	92.788	8.158	1.210
5	92.474	93.158	3.684	0.842
6	89.947	91	9.017	3.158
7	91.895	93.579	4.316	0.895

表 3 实验结果 ( $k=4$ )  
Table 3 Simulation result ( $k=4$ )

专家节点 数目 ( $N$ )	表决正确率 ( $R$ ) / %		表决失败概率 ( $1 - P_c$ ) / %	
	标准	扩展	基本	扩展
4	58.043	59.681	3.152	6.240
5	60.812	62.064	2.50	7.131
6	60.247	61.683	1.401	6.157
7	62.388	63.465	7.349	3.528

从表 1~3 的统计数据可知,扩展表决系统的正确率相对于基本表决系统有较好的性能。对于同一组专家使用扩展表决系统得到的正确率更高,而且对于不同的专家数目也得到更好的结果。另外,对于专家系统自身特性的不同(不同的  $k$ ),扩展表决系统仍然输出较好的表决结果。可见通过自动分析表决系统的工具和历史记录表并配合特殊情况表,能够提高基本表决系统的输出正确率,即使专家系统自身的特性及整个容错系统中的专家节点数目变化仍然具有较高的正确率。可见,扩展表决系统有较高的正确率,并且在各种情况下都能发挥作用。

需要指出的是,本文所提出的扩展表决系统不仅可以提高多版本软件输出的正确率,而且对其他应用表决系统的领域也具有一定促进作用。比如在

模式识别中,当表决系统的性能更加优良的时候,参与识别的多个分类器不必改动就可以得到正确率更高的分类结果。

4 结论

本文通过使用一种自动分析表决系统的工具作为一个模块,构建一个具有一定学习能力的扩展表决系统,在实验中得到了优化的效果。本文的扩展表决系统还可以对辅助结构采取特定的配置,对规则进行修改,以针对某一个具体应用来进行优化,替换已有的表决系统,以得到更好的整体性能,有广泛的应用前景。

参考文献：

[1] HO T K, HULL J J, SRIHARI S N. Decision combination in multiple classifier systems[J]. IEEE Transactions on PAMI, 1994, 16(1):66 - 75.

[2] LI Shaoming, YIN Qian, GUO Ping, et al. A hierarchical mixture of software reliability model for prediction [J]. Applied Mathematics and Computation, 2007:1120 - 1130.

[3] 尹乾, 李邵明, 郭平. 软件可靠性分层模型的案例研究[M] 梅宏,刘超. 软件技术进展,2004:242 - 246.

[4] LAM L. SUEN C. Application of majority voting to pattern recognition: An analysis of its behavior and performance[J]. IEEE Transactions on Systems, Man and Cybernetics,1997,27(5).

[5] LIN X, YACOUN S, BURNS J, et al. Performance analysis of pattern classifier combination by plurality voting[J]. Pattern Recognition Letters, 2003:959 - 1969.

[6] YACOUN Sherif, LIN XiaoFan, SIMSKE S, et al. Automating the analysis of voting systems[C] Proceedings of the 14th IEEE International Symposium on Software Reliability Engineering, 2003.

[7] LEVITIN G, LISNIANSKI A. Reliability optimization for weighted voting system [J]. Reliability Engineering and System Safety, 2001: 131 - 138.

[8] NORDMANN L, PHAM H. Weighted voting systems [J]. IEEE Transactions on Reliability, 1999(1).

[9] PARHAM I B. Voting algorithms[J]. IEEE Transaction on Reliability, 1994,43: 617 - 629.

[10] PARIS J. Voting with witnesses: A consistency scheme for replicated files[C] Proceedings of the 6th International Conference on Distributed Computing Systems, 1986,5:606 - 616.

**参考文献:**

- [1] 宫云战. 软件测试[M]. 北京: 国防工业出版社, 2006: 112 - 130.
- [2] 宫云战. 一种面向故障的软件测试新方法[J]. 装甲兵工程学院学报, 2004(1): 21 - 25.
- [3] FindBugs/Java[CP/OL]. [http: findbugs. sourceforge. net/](http://findbugs.sourceforge.net/).
- [4] 张青. Java 程序的优化[J]. 青海师范大学学报, 2004(4): 42 - 44.
- [5] 巩晶, 郭庆平. Java 性能优化[J]. 微电脑应用, 2002(3): 16 - 17.
- [6] 贺钧. 通过静态分析提高程序测试的效率[J]. 现代电子技术, 2002(11): 7 - 8.

## Analysis of performance fault models in Java program

HOU Bin ZHANG GuanNan, GONG YunZhan

(School of Computer Science and Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China)

**Abstract:** Fault-oriented software testing method is rapidly developing and one of the keys to this method is to define all fault models. This paper mainly introduces fault model related to software performance in Java language. The general pattern of these fault models and their causes are discussed, and their importance to software quality control is analyzed. Finally, measures to improve this kind of fault model are suggested.

**Key words:** fault models; Java; fault detection

(上接第 109 页)

## Improved extended voting system for multi-version software

ZHANG YiNan<sup>1</sup> YIN Qian<sup>1</sup> GUO Ping<sup>1,2</sup>

(1. College of Information Science and Technology, Beijing Normal University, Beijing 100875;

2. College of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China)

**Abstract:** An extended plurality voting system which combines automatic analysis tool of voting system was proposed to improve performance of voting system. As a well-known technology, voting is used in fault tolerant applications of software systems to mask errors using N-Modular Redundancy (NMR) and N-version Programming. Although theoretical analysis of theoretical model of voting systems provides rigorous insight into understanding their expected behavior, theoretical model are often hard to use in practice. Therefore, we propose this extended voting system: using the automatic analysis tool which includes building decision tree and depth first traversal (DFT) algorithm, then to extending the basic plurality voting system with special rules which rebuild decision tree, finally, reference results of that tool analyzes each decision tree to select a more reliable voted result as output. Simulation experiments showed correctness rate of this extended plurality voting system's output was higher than basic plurality voting system.

**Key words:** multi-version software; voting system; fault-tolerance; decision tree