文章编号:1671-4598(2014)11-3480-03　　中图分类号: TP206　　文献标识码:A

基于 Rollout 算法的冗余多故障诊断策略

黄以锋[1], 景　博[1], 王春晖[2], 窦　雯[3]
(1. 空军工程大学 航空航天工程学院, 西安　710038; 2. 中国人民解放军 95825 部队, 湖北 孝感　432100;
3. 空军驻成都地区军事代表局, 成都　610041)

摘要: 针对冗余系统的多故障诊断问题, 分析了冗余多故障系统的相关性矩阵模型, 通过对模型进行转换后, 用 Rollout 算法对基于信息熵的诊断策略进行优化, 提出了一种新的多故障诊断策略; 该策略不需要计算最小碰集, 而且具有先测试, 最后修复的特点, 在具体应用中更便于操作; 实例计算结果表明, 和现有算法相比, 本策略得到的期望测试费用更低。

关键词: 多故障诊断; 冗余系统; 可测性设计; Rollout 算法; 诊断策略

Multiple Fault Diagnosis Strategy for Redundant System Based on Rollout Algorithm

Huang Yifeng[1], Jing Bo[1], Wang Chunhui[2], Dou Wen[3]
(1. College of Aeronautics and Astronautics Engineering, Air Force Engineering University, Xi'an　710038, China;
2. Unit 95825 of Chinese People's Liberation Army, Xiaogan　432100, China;
3. Air Force Chendu Region Resident Military Representative Bureau, Chendu　610041, China)

Abstract: The problem of multiple fault diagnosis strategy for redundant system is considered. The fault－test relation matrix model is analyzed first. Based on the exchange of the relation matrix model, Rollout algorithm is used to improve the information entropy based diagnosis strategy. Then, a new diagnosis strategy for multiple fault redundant system is proposed. This method doesn't need Computing minimal compact set, and has the characteristic that test first and repair last. It's more useful for practical projects. The example demonstrates that the strategy can get lower expect cost than the existing method.

Keywords: multiple fault diagnosis; redundant system; design for testability; Rollout algorithm; diagnostic strategy

0　引言

故障诊断与测试过程中都会遇到序贯诊断策略问题, 它要解决的问题是在满足故障诊断、隔离要求的基础上, 选择期望测试代价尽量低一组测试序列来完成测试, 该问题已被证明是 NP 完全问题[1]。在组成部件繁多的复杂系统中, 往往会出现多个故障同时发生的情况。这时, 由于隐藏故障与掩盖故障[2]的存在, 单故障假设的诊断策略不再适用。目前, 人们已经针对序贯多故障诊断问题开展了大量研究, 并取得了不少成果[2-6]。但在部分复杂系统中, 为了提高系统的可靠性, 在一些关键部件中采用了冗余设计。在这些冗余系统中, 只有冗余模块全部发生故障, 才会导致整个系统工作异常, 我们把这类故障称为最小故障[7]。正是由于这类最小故障的存在, 使得非冗余系统的相关性模型和诊断策略不能直接应用于冗余系统。目前, 对冗余系统的多故障诊断问题的研究成果较少。Shakeri[7]、Yang 等[8-9]分别提出了基于确定算法和基于布尔逻辑推理的冗余系统多故障诊断策略。

Rollout 算法是一种用于组合优化问题的计算方法, 并被应用于解决序贯诊断策略问题[10]。本文在对冗余多故障系统相关矩阵模型进行转化后, 用 Rollout 算法对基于信息熵的多故障诊断策略进行优化, 提出了一种新的多故障诊断策略。

收稿日期:2014-04-08;　**修回日期:**2014-05-12。

作者简介:黄以锋(1982-),男,湖南耒阳人,讲师,博士,主要从事测试性设计、故障诊断方向的研究。
景　博(1965-),女,教授,博士生导师,主要从事故障预测与健康管理、测试性设计与验证方向的研究。

1　问题描述

冗余多故障诊断测试模型可用五元组 (S,P,T,C,B) 表示, 其中:

1) $S=\{s_1,s_2,...s_m,...,s_z\}$ 为系统的独立故障状态集, 其中 $s_{m+1}\sim s_z$ 表示最小故障。为了便于描述, 本文定义 $\boldsymbol{S}=S\cup\{s_0\}=\{s_0,s_1,...,s_z\}$, 为包括正常状态的故障状态集, 其中 s_0 为正常状态, 符号 $\cup$ 表示并集运算;

2) $\boldsymbol{P}=[p(s_1),p(s_2),...,p(s_m)]^T$ 是系统各故障状态的先验概率矢量;

3) $T=\{t_1,t_2,...,t_n\}$ 是 n 个可用测试的集和, 每个测试可测试故障状态的某个子集, 我们假设测试的结果是绝对可靠的;

4) $\boldsymbol{C}=[c_1,c_2,...,c_n]^T$ 是以时间、人力要求、设备要求等综合起来的测试费用矢量, 矢量中, 各个测试费用与测试是相对应的;

5) $\boldsymbol{B}=[b_{ij}]$是 $m\times n$ 的故障—测试相关二进制矩阵, 它表示系统的故障状态集 S 和测试集 T 之间的关系, 如 t_j 能检测到 s_i, 则 $b_{ij}=1$; 否则 $b_{ij}=0$。

冗余多故障诊断问题可表述为设计可隔离多故障状态的多故障诊断策略, 并且要求期望测试费用尽量低。期望测试费用 J 的计算公式为:

$$J=\sum_{S_I\subseteq S}p(S_I)E(C\mid S_I)=\sum_{S_I\subseteq S}\sum_{t_j\in \mathbf{R}_I}p(S_I)c_j \qquad (1)$$

式中，$E(\cdot)$ 为期望值，$E(C\mid S_I)$ 为假设多故障状态 S_I 发生时，隔离需要的所有费用，$S_I\subseteq \boldsymbol{S}$，为最终隔离的多故障状态，也就是诊断树的叶子节点中包含的多故障状态，$p(S_I)$ 为多故障状态的发生概率，R_I 为用于隔离该故障状态所用到的测试的集合。

S_I 可用一个 m 维的向量 $\boldsymbol{X}$ 表示：$\boldsymbol{X}=[x_1,\ldots,x_m]'$，若 $s_i\in S_I$，则 $x_i=1$，否则 $x_i=0$。S_I 发生的概率可表示为：

$$p(S_I)=\prod_{i=1}^{m}p(s_i)^{x_i}(1-p(s_i))^{1-x_i} \qquad (2)$$

需要说明的是，诊断树的一个叶子节点中，可能会包含多个多故障状态，而且这些故障状态经历过的测试都是一样的，可以先计算诊断树的各个叶子节点的期望测试费用，再求和得到整个诊断树的期望测试费用 J，所以式（1）又可表示为：

$$J=\sum_{A_k\subseteq A}(\sum_{S_I\subseteq A_k}p(S_I)\times\sum_{t_j\in \mathbf{R}_I}c_j) \qquad (3)$$

其中：A 为所有叶子节点的集合，A_k 为该叶子节点包含多故障状态的集合，R_I 为用于隔离到叶子节点 A_k 所用的测试的集合。

2　冗余多故障诊断策略

首先，经过扩展、合并将冗余多故障系统矩阵模型转化为单故障系统矩阵模型。在转化的过程中要处理好最小故障。例如，若 s_5 是最小故障，当 s_3 和 s_4 都发生故障时，s_5 发生故障。这样，多故障状态 $\{s_3,s_5\}$ 是不存在的，因为若 s_5 发生故障，则 s_3 和 s_4 也必然发生故障。所以在转化过程中要根据最小故障来修改单故障状态。另外，最小故障的发生概率显然依赖于各组成单元的故障发生概率，在计算转化后的单故障状态的概率时，无需考虑最小故障的影响，但由于删除了一些单故障状态，所以需要对计算出的概率进行重新分配。然后采用 Rollout 算法对得到的单故障系统相关性矩阵模型进行计算。

2.1　建立多故障相关性矩阵模型

1）假设要测试的系统故障状态集为 $S=\{s_0,s_1,\ldots,s_m,s_{m+1},\ldots,s_z\}$，包括一个正常状态 s_0、m 个独立的故障状态 $s_1\sim s_m$ 和最小故障 $s_{m+1}\sim s_z$。测试集为 $T=\{t_1,t_2,\ldots,t_n\}$，包括 n 个可用测试。系统各故障状态的先验概率矢量为 $\boldsymbol{P}=[p(s_1),p(s_2),\ldots,p(s_m)]^T$。与测试相对应的测试费用矢量为 $\boldsymbol{C}=[c_1,c_2,\ldots,c_n]^T$。故障—测试相关二进制矩阵为 $\boldsymbol{B}=[\,b_{ij}]$。

2）将单故障状态集 $\boldsymbol{S}$ 扩展为多故障状态集 $\boldsymbol{S}'=\{s'_0,s'_1,\ldots,s'_{2^m-1}\}$，$\boldsymbol{S}'$ 中的任意多故障状态 s'_i 可用一个 m 维的向量 $X=[x_1,\ldots,x_m]'$ 表示，其中，若 $s_j\in s'_i$，则 $x_j=1$，否则 $x_j=0$，并且满足：

$$\sum_{j=1}^{m}2^{j-1}x_j=i \qquad (4)$$

多故障状态 s'_i 的故障—测试相关二进制行向量（故障特征信息）为它所包含的单故障状态对应的二进制行向量相或得到。

多故障状态 s'_i 的概率则用式（5）进行计算。

$$p(s'_i)=\prod_{j=1}^{m}p(s_j)^{x_j}(1-p(s_j))^{1-x_j} \qquad (5)$$

3）将多故障状态 $\boldsymbol{S}'$ 扩展为 $\boldsymbol{S}''=\{s''_0,s''_1,\ldots,s''_{2^{m-1}-1}\}$，其中 $\boldsymbol{S}''$ 中的任意多故障状态 s''_i 可用一个 z 维的向量 $X'=[x_1,\ldots,x_m,x_{m+1},\ldots,x_z]'$ 表示，其中，$x_1\sim x_m$ 的取值和 s'_i 一致，对于 $x_{m+1}\sim x_z$，若 $m+1\leqslant j\leqslant z$，则 s_j 是最小故障，它包括多个故障状态，若这些故障状态都被 s'_i 包含，则 $x_j=1$，否则 $x_j=0$。

多故障状态 s''_i 的故障—测试相关二进制行向量（故障特征信息）为 s'_i 和它所包含的最小故障对应的二进制行向量相或得到。

多故障状态 s''_i 的概率和 s'_i 相同。

4）将故障—测试相关二进制行向量（故障特征信息）相同的多故障状态进行合并，合并后的多故障状态用 s'''_k 表示，k 为合并的多故障状态中，序号最小的多故障状态的序号。s'''_k 的概率为合并的所有多故障的概率之和。将合并后的多故障状态集用 $\boldsymbol{S}'''$ 表示。

2.2　用 Rollout 算法计算出测试序列

首先对故障状态集 x 和测试集 t 进行初始化，x 初始化后为合并处理后的多故障状态 $\boldsymbol{S}'''$，t 初始化为可用测试集为 T。接下来的计算过程和文献［11］中算法的第二部分一致，这里不再具体给出。

3　实例分析与比较

下面，通过一个实例来展示本文算法的具体计算过程，并将计算结果与现有算法进行比较。实例 1 的信号流模型如图 1 所示[9]。

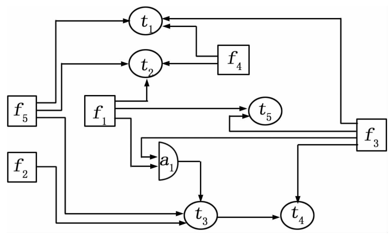

图 1　实例 1 的信息流模型

实例 1 的相关性矩阵如表 1 所示，故障状态集为 $\boldsymbol{S}=\{s_0,s_1,s_2,s_3,s_4,s_5,s_6\}$，包括一个正常状态 s_0、5 个独立的故障状态 $s_1\sim s_5$ 和最小故障 s_6，测试集为 $T=\{t_1,t_2,t_3,t_4,t_5\}$。5 个独立的故障状态 $s_1\sim s_5$ 的先验概率矢量为 $\boldsymbol{P}=[0.014,0.027,0.125,0.068,0.146]^T$，与测试相对应的测试费用矢量为 $\boldsymbol{C}=[1,1,1,1,1]^T$。首先，将故障状态集 $\boldsymbol{S}$ 扩展为多故障状态集 $\boldsymbol{S}'$，并用式（5）计算扩展后各故障状态的条件概率，得到第一次扩展后的多故障相关性矩阵模型。

表 1　实例 1 的相关性矩阵

故障状态	t_1	t_2	t_3	t_4	t_5	概率
	1	1	1	1	1	
s_0	0	0	0	0	0	——
s_1	0	1	0	0	1	0.014
s_2	0	0	1	1	0	0.027
s_3	1	0	0	1	1	0.125
s_4	1	1	0	0	0	0.068
s_5	1	1	1	1	0	0.146
$s_6=\{s_1,s_3\}$	1	1	1	1	1	——

接着，将多故障状态集 $\boldsymbol{S}'$ 扩展为多故障状态集 $\boldsymbol{S}''$，并用 $X'=[x_1,\ldots,x_6]'$ 表示各个扩展后的多故障状态，得到第二次扩展后的多故障相关性矩阵模型，如表 2 所示。这里以 s''_7 为例，说明表中数据的由来。对于 s''_7，$x_1\sim x_5$ 的取值和 s'_7 一致，而 s'_7 包括的单故障状态为 s_1、s_2 和 s_3，其中的 s_1 和 s_3

组成了最小故障 s_6，所以 $x_6=1$。多故障状态 s''_7 的故障—测试相关二进制行向量（故障特征信息）为 s'_7 和它所包含的最小故障 s_6 对应的二进制行向量相或得到。多故障状态 s''_7 的条件概率和 s'_7 相同。

表2 第二次扩展后的多故障相关性矩阵模型

多故障状态	X' 的取值	包含的单故障状态	t_1	t_2	t_3	t_4	t_5	条件概率
			1	1	1	1	1	
s''_0	[0,0,0,0,0,0]′	s_0	0	0	0	0	0	0.668 1
s''_1	[1,0,0,0,0,0]′	s_1	0	1	0	0	1	0.009 5
s''_2	[0,1,0,0,0,0]′	s_2	0	0	1	1	0	0.018 5
s''_3	[1,1,0,0,0,0]′	s_1s_2	0	1	1	1	1	0.000 3
s''_4	[0,0,1,0,0,0]′	s_3	1	0	0	1	1	0.095 4
s''_5	[1,0,1,0,0,1]′	$s_1s_3s_6$	1	1	1	1	1	0.001 4
s''_6	[0,1,1,0,0,0]′	s_2s_3	1	0	1	1	1	0.002 6
s''_7	[1,1,1,0,0,1]′	$s_1s_2s_3s_6$	1	1	1	1	1	0.000 0
s''_8	[0,0,0,1,0,0]′	s_4	1	1	0	0	0	0.048 7
s''_9	[1,0,0,1,0,0]′	s_1s_4	1	1	0	0	1	0.000 7
s''_{10}	[0,1,0,1,0,0]′	s_2s_4	1	1	1	1	0	0.001 4
s''_{11}	[1,1,0,1,0,0]′	$s_1s_2s_4$	1	1	1	1	1	0.000 0
s''_{12}	[0,0,1,1,0,0]′	s_3s_4	1	1	0	1	1	0.007 0
s''_{13}	[1,0,1,1,0,1]′	$s_1s_3s_4s_6$	1	1	1	1	1	0.000 1
s''_{14}	[0,1,1,1,0,0]′	$s_2s_3s_4$	1	1	1	1	1	0.000 2
s''_{15}	[1,1,1,1,0,1]′	$s_1s_2s_3s_4s_6$	1	1	1	1	1	0.000 0
s''_{16}	[0,0,0,0,1,0]′	s_5	1	1	1	1	0	0.114 2
s''_{17}	[1,0,0,0,1,0]′	s_1s_5	1	1	1	1	1	0.001 6
s''_{18}	[0,1,0,0,1,0]′	s_2s_5	1	1	1	1	0	0.003 2
s''_{19}	[1,1,0,0,1,0]′	$s_1s_2s_5$	1	1	1	1	1	0.000 0
s''_{20}	[0,0,1,0,1,0]′	s_3s_5	1	1	1	1	1	0.016 3
s''_{21}	[1,0,1,0,1,1]′	$s_1s_3s_5s_6$	1	1	1	1	1	0.000 2
s''_{22}	[0,1,1,0,1,0]′	$s_2s_3s_5$	1	1	1	1	1	0.000 5
s''_{23}	[1,1,1,0,1,1]′	$s_1s_2s_3s_5s_6$	1	1	1	1	1	0.000 0
s''_{24}	[0,0,0,1,1,0]′	s_4s_5	1	1	1	1	0	0.008 3
s''_{25}	[1,0,0,1,1,0]′	$s_1s_4s_5$	1	1	1	1	1	0.000 1
s''_{26}	[0,1,0,1,1,0]′	$s_2s_4s_5$	1	1	1	1	0	0.000 2
s''_{27}	[1,1,0,1,1,0]′	$s_1s_2s_4s_5$	1	1	1	1	1	0.000 0
s''_{28}	[0,0,1,1,1,0]′	$s_3s_4s_5$	1	1	1	1	1	0.001 2
s''_{29}	[1,0,1,1,1,1]′	$s_1s_3s_4s_5s_6$	1	1	1	1	1	0.000 0
s''_{30}	[0,1,1,1,1,0]′	$s_2s_3s_4s_5$	1	1	1	1	1	0.000 0
s''_{31}	[1,1,1,1,1,1]′	$s_1s_2s_3s_4s_5s_6$	1	1	1	1	1	0.000 0

然后，将故障—测试相关二进制行向量（故障特征信息）相同的多故障状态进行合并，合并后的多故障状态如表3所示。其中，合并后的多故障状态的条件概率为被合并的多故障状态的条件概率之和。

表3 合并后的多故障相关性矩阵模型

多故障状态	被合并的多故障状态	t_1	t_2	t_3	t_4	t_5	条件概率
		1	1	1	1	1	
s'''_0	s''_0	0	0	0	0	0	0.668 1
s'''_1	s''_1	0	1	0	0	1	0.009 5
s'''_2	s''_2	0	0	1	1	0	0.018 5
s'''_3	s''_3	0	1	1	1	1	0.000 3
s'''_4	s''_4	1	0	0	1	1	0.095 4
s'''_5	$s''_5s''_7s''_{11}s''_{13}s''_{14}s''_{15}s''_{17}s''_{19}s''_{20}$ $s''_{21}s''_{22}s''_{23}s''_{25}s''_{27}s''_{28}s''_{29}s''_{30}s''_{31}$	1	1	1	1	1	0.021 7
s'''_6	s''_6	1	0	1	1	1	0.002 6
s'''_8	s''_8	1	1	0	0	0	0.048 7
s'''_9	s''_9	1	1	0	0	1	0.000 7
s'''_{10}	$s''_{10}s''_{16}s''_{18}s''_{24}s''_{26}$	1	1	1	1	0	0.127 3
s'''_{12}	s''_{12}	1	1	0	1	1	0.007 0

然后，按照2.2节的计算步骤，用Rollout算法计算出优化测试序列，计算过程如图2所示。

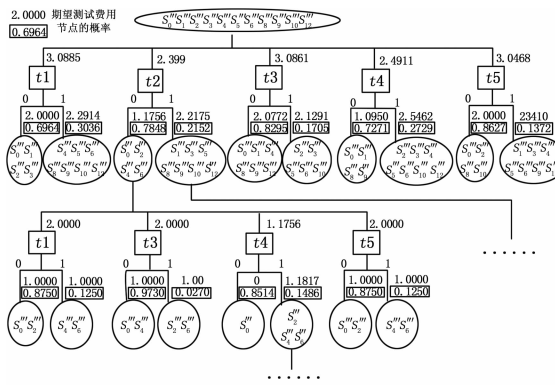

图2 实例1的计算过程

从图中可以看到，开始时，5个测试分别将故障状态集分为两个子集，接着对每个子集进行计算。例如测试 t_2 将故障状态集分为 $\{s'''_0, s'''_2, s'''_4, s'''_6\}$ 和 $\{s'''_1, s'''_3, s'''_5, s'''_8, s'''_9, s'''_{10}, s'''_{12}\}$ 两个子集。可求得子集 $\{s'''_0, s'''_2, s'''_4, s'''_6\}$ 的相关性矩阵模型，用信息熵算法求得故障树，并计算期望测试费用为1.175 6。同理，可求得子集 $\{s'''_1, s'''_3, s'''_5, s'''_8, s'''_9, s'''_{10}, s'''_{12}\}$ 的期望测试费用为2.217 5。若选用 t_2，则可计算出期望测试费用为2.399 9。同理，可求得选用其他测试的期望测试费用。经过比较，测试 t_2 的期望测试费用最小，所以选择测试 t_2 放入优化测试序列。接着用同样的方法对 t_2 分出的子集，以及子集的子集进行处理，依次将选择的测试放入优化测试序列。

最后，根据优化测试序列画出故障诊断树，并计算期望测试费用。实例1的故障诊断树如图3所示。用式（3）可求出期望测试费用为：

$$J = P(A_4)*2+P(A_8)*3+P(A_{10})*3+P(A_{11})*3+P(A_{14})*4+P(A_{15})*4+P(A_{16})*4+P(A_{17})*4+P(A_{19})*4+P(A_{20})*5+P(A_{21})*5=2.399\ 9$$

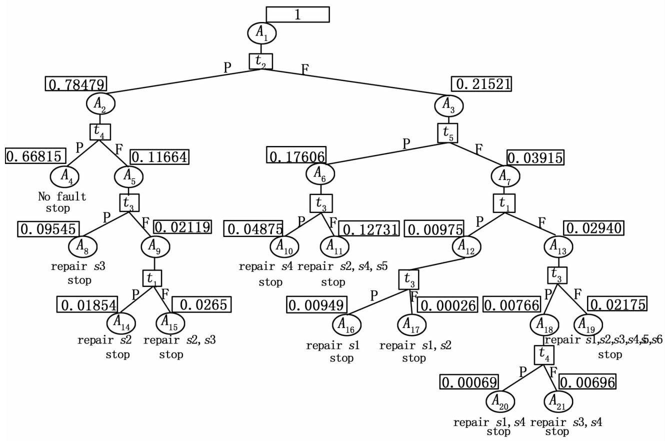

图3 实例1的故障诊断树（本文算法）

图4是文献[9]给出的实例1的故障诊断树，用式（3）进行计算可知，该故障诊断树的期望测试费用为2.784 5，比本文算法的期望测试费用高出了16%。另外，故障诊断树中有两个节点需要在诊断过程中需要故障进行修复，增加了操作

（下转第3486页）