

# 现场条件约束下的动态诊断策略研究

廖小燕, 陆宁云

(南京航空航天大学 自动化学院, 南京 211106)

**摘要:** 针对传统的故障诊断策略属于静态的诊断方法, 难以在复杂的现场工作环境中应用的问题, 在基于准深度优先搜索(QDFS)算法的基础上, 提出了一种基于改进的QDFS算法(考虑测试的不确定性)的现场约束条件下的诊断策略生成方法; 首先分析工作现场可能存在的条件约束, 然后给出条件约束的描述方法, 再根据改进的QDFS算法给出生成诊断策略的具体过程, 最后用案例对方法进行验证; 案例验证结果表明, 该方法不仅考虑了不可靠测试的影响, 并能在现场约束条件下应用起来, 比传统的静态诊断策略更具有适应性和实用性。

**关键词:** 诊断策略; 不可靠测试; QDFS算法; 条件约束

## Research on Dynamic Fault Diagnosis Strategy Under Condition Constraints

Liao Xiaoyan, Lu Ningyun

(College of Automation, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China)

**Abstract:** The traditional fault diagnosis strategy is a method to generate the static diagnosis, which is difficult to be applied in the complex system and operating ambient. Based on the quasi depth first search (QDFS) algorithm, this paper proposes a method of generating the diagnosis strategy under the condition constraints based on the improved QDFS algorithm (considering the uncertainty of the test). Firstly, the paper analyzes the possible condition constraints in the operating ambient, then gives the description method of the condition constraints, then according to the improved QDFS algorithm, gives the dynamic process of generating the diagnosis strategy. Finally a case is used to verify the effectiveness of the proposed method. The result shows that this method not only considers the influence of unreliable test, but also can be applied under the condition constraints of operating ambient. It is more adaptive and practical than the traditional static diagnosis strategy.

**Keywords:** diagnosis strategy; unreliable test; quasi depth first search (QDFS) algorithm; conditional constraints

## 0 引言

测试性是指产品能够及时准确地确定其状态(可工作、不可工作或性能下降), 且对内部故障进行有效隔离的一种设计特性<sup>[1-2]</sup>。在测试性设计中, 诊断策略优化设计是极其重要的内容之一<sup>[3]</sup>。诊断策略是结合目标、约束以及其他相关因素整体优化来准确对系统故障诊断的一种方法<sup>[4]</sup>。相比于传统的故障诊断方法, 其侧重以故障诊断为目的的测试内容选取和调度问题, 用以提升装备故障诊断的能力、提高诊断效率, 进而提高装备使用度、降低装备整个寿命周期的费用<sup>[5-6]</sup>。

随着高新装备的发展, 给装备的测试性设计和故障检测与诊断带来了很多问题<sup>[7-9]</sup>。故障诊断策略优化设计的主旨是以最小的测试代价生成最优的测试序列为目标的一种测试优化方法, 其对于提高设备或系统的寿命、增强维修性能等都具有重要的意义<sup>[10-11]</sup>。传统的故障诊断策略设计

方法属于静态的诊断方法, 难以在复杂的装备系统中得到应用。主要问题表现于: 维修人员只能按照固定测试步骤来开展故障诊断<sup>[12-13]</sup>, 而没有考虑在设备工作的现场条件约束, 如缺少现场测试资源导致某些测试不能使用的情况; 或者某测试因为其测试过程很简单导致其结果是直接可观测, 用户主动选择该测试; 亦或是某两个或者某几个测试之间存在相互约束的关系, 即某个测试必须在另一个测试执行之后才能执行。本文主要考虑上述的现场复杂工况的条件约束下, 提出了一种动态的故障诊断策略生成方法, 该方法是以不可靠测试下的准深度优先搜索(Quasi-Depth First Search, QDFS)算法的故障诊断策略为基础, 根据现场约束条件而确定下一步测试选择, 最终生成完整的动态故障诊断策略。

## 1 问题描述

诊断策略是具有一定顺序的测试序列。当考虑到测试不确定性问题中出现的虚警、漏检的现象时, 为系统故障诊断生成测试序列的问题包括以下部分:

- 1) 有限的系统状态集合  $S = \{s_1, s_2, \dots, s_m, s_{m+1}\}$ , 其中  $m$  为系统的状态总数,  $s_{m+1}$  表示系统的无故障状态。
- 2) 与系统状态集合  $S$  对应的状态先验概率集合  $P = \{p_1, p_2, \dots, p_m, p_{m+1}\}$ 。
- 3) 有限的系统可用测试集合  $T = \{t_1, t_2, \dots, t_n\}$ ,

收稿日期: 2020-01-16; 修回日期: 2020-03-23。

基金项目: 国家自然科学基金(61873122)。

作者简介: 廖小燕(1995-), 女, 重庆人, 硕士, 主要从事测试性设计与故障诊断方向的研究。

陆宁云(1978-), 女, 江苏连云港人, 博士生导师, 教授, 主要从事复杂工业工程建模、过程监测、故障诊断、产品质量控制等方向的研究。

其中  $n$  为可用测试的总数。

测试的代价集合  $C = \{c_1, c_2, \dots, c_n\}$ 。

4)  $P_T = [pd_{ij}, pf_{ij}]$  代表测试的不确定性, 其中,  $pd_{ij} = P\{O(t_j) = 1 | f_i = 1\}$ ,  $pf_{ij} = P\{O(t_j) = 1 | f_i = 0\}$  分别表示测试的检测率和虚警率。

5) 系统的故障与测试相关性矩阵  $D = [d_{ij}]$ , 其中  $d_{ij} = 0$  表示系统状态  $s_i$  可以被测试  $t_j$  检测到, 反之,  $d_{ij} = 1$  则表示系统状态  $s_i$  不能被测试  $t_j$  检测到。

通过特定的算法, 根据系统的相关性矩阵得到一组测试序列, 以检测并隔离系统的各个故障状态, 同时尽量减少预期的测试成本。估算测试费用的计算公式如下:

$$\min_{\{P_i\}_{i=1}^{m+1}} J = \sum_{i=1}^{m+1} \left\{ \sum_{j=1}^{|P_i|} C_{P_i[j]} \right\} P(s_i) \quad (1)$$

式中,  $P_i$  表示隔离系统状态  $s_i$  的测试序列,  $|P_i|$  表示这个序列中的测试数目。

## 2 现场约束条件分析

在实际维修过程中, 现场可能出现某些测试不可用, 用户需要主动执行某些测试, 某些测试间存在顺序相关性等问题。本文充分考虑了这些现场约束条件, 根据约束条件得到一个临时  $D$  矩阵, 然后通过该矩阵, 结合改进的准深度优先搜索算法计算得到当前被选测试, 使得最终的诊断策略具有较高的实用性。

1) 测试  $t_j$  现场不可测的情况:

在现场工作环境中, 也许会发生某些测试所对应的传感器短缺或故障等情况, 从而导致需要的测试无法执行, 使所需测试  $t_j$  变得不可测。

2) 用户主动选择测试  $t_j$  的情况:

在现场工作环境中, 维修人员也许会因为现场约束条件而主动选取测试  $t_j$ , 比如某些测试的结果直接表现出故障现象。

3) 测试  $t_i$  与  $t_j$  存在顺序相关性:

在复杂的系统或者工况下, 有些测试之间存在相互影响的情况, 即存在某测试必须在另一测试执行之后方可用。

## 3 基于改进的 QDFS 算法的动态故障诊断策略生成

### 3.1 现场约束条件的描述方法

为了描述上述存在的各种现场约束条件, 给出以下几个定义。

1) 测试状态集  $St$ :

测试状态集  $St$  用来描述测试的可用或不可用的状态:

$$St = \{St_1, St_2, \dots, St_n\} \quad (2)$$

其中:  $st_j = 0, 1$ 。当  $st_j = 0$  时表示测试  $t_j$  的状态为测试不可用,  $st_j = 1$  时表示测试  $t_j$  的状态为可用。

根据第 2 节中的约束条件一, 可以知道当测试  $t_j$  不可用时,  $St = \{1, 1, \dots, 0, \dots, 1, 1\}$ ;

2) 测试-测试相关性矩阵:

为了描述测试与测试之间的顺序相关性, 引出一种测

试-测试相关性矩阵:

$$TT = \begin{bmatrix} tt_{11} & tt_{12} & \dots & tt_{1n} \\ tt_{21} & tt_{22} & \dots & tt_{2n} \\ \vdots & \vdots & \dots & \vdots \\ tt_{n1} & tt_{n2} & \dots & tt_{nn} \end{bmatrix} \quad (3)$$

其中:  $tt_{ij} = 0, 1$ 。当  $tt_{ij} = 0$  时表示测试  $t_i$  与  $t_j$  之间不存在测试顺序相关性, 当  $tt_{ij} = 1$  时表示测试  $t_i$  必须在测试  $t_j$  之前执行, 也就是说, 如果需要执行测试  $t_j$ , 那么必须先检查测试  $t_i$  的执行情况。

如果在某案例中, 已经给定  $t_1$  与  $t_2$  存在顺序相关性, 则  $tt_{12} = 1$ 。那么当  $t_1$  还未执行时,  $St_2 = 0$ ; 当  $t_1$  执行过后,  $St_2 = 1$ 。即

$$St_2 = (1 - St_1) \cdot tt_{12} \quad (4)$$

3) 临时  $D$  矩阵:

每需要选取下一步测试时, 都需要获得当前条件下的临时  $D$  矩阵。

对于约束条件一, 某测试  $t_j$  不可用, 即  $st_j = 0$ , 则临时  $D$  矩阵则是在上一步的  $D$  矩阵下删除不能用测试  $t_j$ ;

对于约束条件二, 主动选择某测试  $t_j$ , 那么根据该测试的结果获得临时  $D$  矩阵, 如果  $t_j$  测试结果为通过, 则选取原  $D$  矩阵中测试  $t_j$  所在列对应的元素为 0 的所有的行为临时  $D$  矩阵的行, 原  $D$  矩阵的列中删除  $t_j$  的列为临时  $D$  矩阵的列。

对于约束条件三, 测试  $t_i$  与测试  $t_j$  存在测试顺序相关性, 则根据测试状态集确定下一步的待选的所有测试可用或不可用状态, 然后取原  $D$  矩阵中对应的  $St$  中为 1 的测试和需要隔离的故障状态组成临时  $D$  矩阵。

### 3.2 改进的 QDFS 算法

准深度优先搜索算法是基于信息熵算法的单步向前回溯算法。该方法的基本思想是: 使用信息熵算法分别构造一个以每个候选测试为顶点的临时诊断树, 然后使用启发式函数  $k^*$  计算并比较这些临时诊断树以获得最佳诊断树, 而诊断树的第一个测试是当前的最佳测试。

1) 初始化:

令系统的初始状态  $x$  为系统的所以状态集合  $S$ , 初始测试集合  $t$  为系统所有的可用测试集合  $T$ 。

2) 信息量计算:

测试集合  $t$  中的每一个测试  $t_j$  将状态集合  $x$  分为  $k$  个子集, 计算每个子集的概率为:

$$p(x_{j_i}) = \sum_{s_i \in x_{j_i}} p(s_i), k = 0, 1 \quad (5)$$

更新每个故障子集的概率:

$$p^*(s_i) = \frac{p(s_i)}{\sum_{s_i \in x_{j_i}} p(s_i)}, k = 0, 1 \quad (6)$$

利用信息熵算法计算每个故障子集的最优故障诊断策略, 并且以测试  $t_j$  作为第一个测试生成临时故障诊断树  $D_j$ 。

利用下面的启发式函数  $k^*$  计算每个临时诊断树  $D_j$  的单位信息量:

$$k^* = \arg \max_j \left\{ \frac{Pr(t_j) \times IG(x; D_j) \times R(t_j)}{COST(x; D_j)} \right\} \quad (7)$$

式 (7) 中,  $IG(x; D_j)$  表示临时故障诊断树  $D_j$  的信息量,  $COST(x; D_j)$  表示临时故障诊断树的平均测试代价,  $Pr(t_j)$  表示测试  $t_j$  的故障检测能力,  $R(t_j)$  表示测试  $t_j$  的可靠度。

$$IG(x; D_j) = - \sum_{k=1}^{l_j} \left[ \frac{p(x_{jk})}{p(x)} \log_2 \frac{p(x_{jk})}{p(x)} \right] \quad (8)$$

$$COST(x; D_j) = \sum_{k=1}^{l_j} \left[ \frac{p(x_{jk})}{p(x)} \sum_{i=1}^{D_{j,k}} C_{D_{j,k}[i]} \right] \quad (9)$$

$$Pr(t_j) = \sum_{i=1}^{m+1} (1 - p(O_i(s_{m+1} | s_i)) p(s_i)) \quad (10)$$

$$R(t_j) = Pd_j^a + (1 - Pf_j)^b \quad (11)$$

3) 测试选择:

从集合  $t$  选择单位测试代价具有信息量最大的测试:  $t_q$ 。

4) 概率更新:

系统状态集合  $x$  被测试  $t_q$  划分为 2 个集合, 分别为  $x_{q0}$ ,  $x_{q1}$ , 更新每个子集的故障概率。

5) 重复:

令测试集合  $t$  为除去测试  $t_q$  的原测试集合, 令状态集合  $x$  为原状态集合的各个故障子集。重复第 2) ~ 4) 步, 直到各故障状态子集中的元素少于或等于 1, 或者测试集合  $t$  中元素少于 1。

3.3 诊断策略生成过程

在诊断策略设计过程中, 首先利用 TEAMS 软件工具建立系统的多信号模型<sup>[14-16]</sup>, 得到被诊断对象的  $D$  矩阵, 然后在实际诊断过程中随外部控制条件, 如症状、资源等变化获取所有的测试状态集合  $St$ , 并根据原  $D$  矩阵获取下一步的临时  $D$  矩阵, 在临时  $D$  矩阵基础上利用改进的准深度搜索算法产生下一步最佳的测试, 再根据测试执行情况调整测试状态集合  $St$ , 生成新的临时  $D$  矩阵, 然后利用改进的准深度搜索算法, 直至得出最后的诊断结论。具体步骤如下:

1) 根据系统的结构原理, 得到系统故障测试相关性矩阵  $D$ , 故障状态集  $S$ , 各个状态对应的故障先验率  $P$ , 可用测试集  $T$ , 测试对于费用集合  $C$ , 各个测试对应的检测率和虚警率等。

2) 初始化。令系统初始故障状态集合  $x=S$ , 初始测试集合为  $t=T$ , 初始测试状态集合  $St = \{0, 0, \dots, 0\}$ , 从现场工作条件中获得现场存在的初始条件约束。

3) 根据当前的条件约束更改测试状态集合  $St$ 。如果有某测试不可用, 则令该测试对应的  $st_j=0$ ; 如果前面执行算法的过程中有某测试不可用但当前条件使某测试可用了, 则令该测试的状态  $st_j$  由 0 变为 1; 如果某些测试存在顺序相关性, 则后面的测试对应的状态  $st_k = (1 - st_l) tt_k$ 。

4) 根据当前测试状态集合确定可用测试集  $t$ 。取测试状态集合  $St$  中的为 1 的元素对应的测试为可用测试集合  $t$ 。

5) 根据当前的故障状态集合  $x$ , 测试集合为  $t$ , 生成临时  $D$  矩阵, 用上一节的启发式搜索函数 (式 (6)) 选择下一步测试  $t_q$ 。

6) 系统状态集  $x$  被测试  $t_q$  分为两个子集, 更新系统故障率。

7) 令系统状态集  $x$  为 6) 中的子集, 将已执行的测试对应的状态更改为 0, 并从现场工况重新获得当前的条件约束, 重复 3) ~ 6) 直到  $St$  中元素全为 0 或者所有  $x$  中元素数不超过 1。

由此形成的诊断策略就是动态诊断策略生成过程, 如图 1 所示。

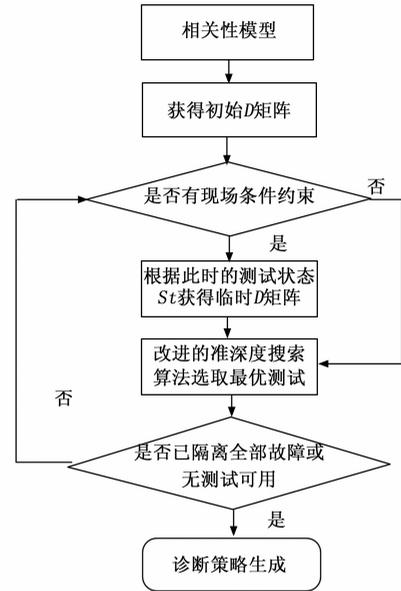


图 1 动态诊断策略生成过程

4 案例分析

为了验证在现场约束条件下所提出的方法能生成有效的测试序列问题, 使用经典的 Apollo 检测系统<sup>[17-19]</sup>进行案例研究。阿波罗探测系统是一个复杂的系统, 其物理结构和测试条件比较复杂。已知 Apollo 检测系统的故障测试相关性矩阵  $D$  和故障概率的分布如表 1 所示。每种测试的检测概率, 虚警概率和成本如表 2 所示。

在没有现场约束条件的情况下, 根据改进的 QDFS 算法生成的不可靠测试条件下的静态故障诊断策略如图 2 所示, 该诊断策略的总代价为 132.85。

接下来考虑在现场约束条件下的诊断策略。

假设在某一次检测中发现测  $t_4$  出现故障现象, 即  $t_4$  测试结果直接可观测, 并且  $t_{12}$  对应的传感器出现故障, 即  $t_{12}$  不可测, 并且存在现场测试的顺序相关性约束,  $t_{15}$  必须在  $t_1$  的后面执行。

根据条件约束一和二, 可以得到表 3 所示的临时  $D$  矩阵。如果不考虑  $t_{15}$  和  $t_1$  的顺序相关性, 可以获得的诊断树如图 3 所示, 从图中可以看出, 因为现场约束条件的影响,

表 1 Apollo 系统的故障测试相关性矩阵

故障状态	测试															故障率
	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$	$t_6$	$t_7$	$t_8$	$t_9$	$t_{10}$	$t_{11}$	$t_{12}$	$t_{13}$	$t_{14}$	$t_{15}$	
$s_1$	0	0	0	1	0	0	0	1	0	1	1	1	1	0	0	0.03
$s_2$	0	0	1	0	1	0	0	0	0	0	1	1	0	1	0	0.07
$s_3$	0	0	0	0	1	1	1	0	1	1	1	1	0	0	1	0.02
$s_4$	0	1	0	0	0	1	1	0	0	0	0	0	0	1	1	0.01
$s_5$	0	1	0	1	0	1	1	1	1	1	0	0	1	1	0	0.03
$s_6$	0	0	0	1	1	0	0	1	1	1	0	1	1	1	1	0.04
$s_7$	1	0	0	1	1	0	0	1	0	0	1	0	1	0	1	0.02
$s_8$	1	1	1	0	0	1	1	0	1	1	1	0	0	1	0	0.05
$s_9$	1	0	0	1	1	0	0	0	0	0	0	1	0	1	1	0.03
$s_{10}$	1	1	1	1	0	0	1	0	0	0	1	0	1	1	0	0.02
$s_{11}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.68

表 2 Apollo 系统的测试参数

Parameters	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$	$t_6$	$t_7$	$t_8$	$t_9$	$t_{10}$	$t_{11}$	$t_{12}$	$t_{13}$	$t_{14}$	$t_{15}$
pdj	0.980	0.975	0.960	0.920	0.990	0.980	0.975	0.960	0.920	0.990	0.960	0.920	0.990	0.980	0.975
pfj	0.015	0.010	0.020	0.060	0.005	0.015	0.010	0.020	0.060	0.005	0.020	0.060	0.005	0.015	0.010
c	42	46	44	45	70	43	34	54	63	50	56	67	53	82	43

表 3 Apollo 系统的临时 D 矩阵

System State	Dependency Matrix														Fault Probability
	$t_1$	$t_2$	$t_3$	$t_5$	$t_6$	$t_7$	$t_8$	$t_9$	$t_{10}$	$t_{11}$	$t_{13}$	$t_{14}$	$t_{15}$		
$s_1$	0	0	0	0	0	0	1	0	1	1	1	0	0	0.03	
$s_5$	0	1	0	0	1	1	1	1	1	0	1	1	0	0.03	
$s_6$	0	0	0	1	0	0	1	1	1	0	1	1	1	0.04	
$s_7$	1	0	0	1	0	0	1	0	0	1	1	0	1	0.02	
$s_9$	1	0	0	1	0	0	0	0	0	0	0	1	1	0.03	
$s_{10}$	1	1	1	0	0	1	0	0	0	1	1	1	0	0.02	

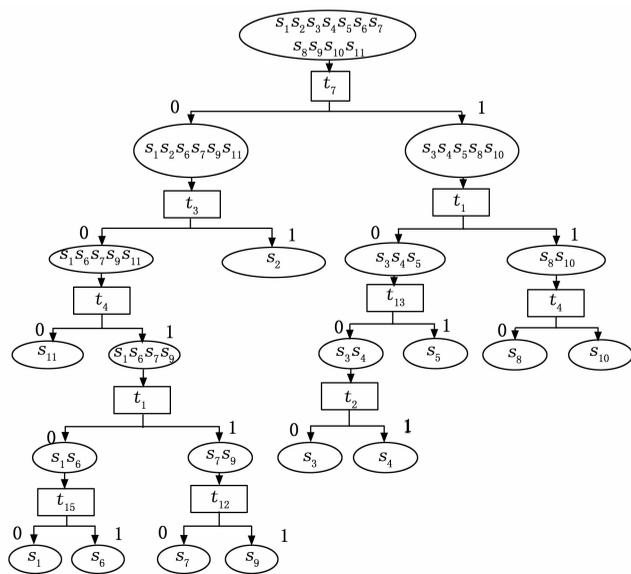


图 2 Apollo 系统的静态诊断策略

测试可用的选择数变少，需要隔离的故障数目更是大幅减少，所以故障诊断树变得很简单，且只需要三步就能隔离出全部故障。由于测试  $t_4$  的结果直接观测得到，为此省略了一些故障隔离的步骤，所以得到的该故障诊断策略的代价也比正常的静态的故障诊断树的代价要低，为 117.82。

上述分析并没有考虑约束条件三的影响，从上面结果可以看出，测试  $t_1$  和测试  $t_{15}$  都有被选择，但是  $t_1$  和  $t_{15}$  约束条件三相违背。所以在进行第二步隔离时，希望选择  $t_1$  作为该步骤的测试，但是  $t_{15}$  还没执行，所以  $t_1$  也不能执行。所以此时的临时  $D$  矩阵是没有  $t_1$  作为备用测试的。根据第三章的所提的改进的诊断策略生成算法，这一步对于隔离  $s_1, s_6, s_7, s_9$  所选的测试为  $t_1$ ，而隔离  $s_5$  和  $s_{10}$  所选的测试为  $t_6$ ，对于  $s_1, s_6, s_7, s_9$  执行完  $t_{15}$  之后，被分为  $s_1$  和  $s_6, s_7, s_9$  两个子集，接下来对  $s_6, s_7, s_9$  进行诊断策略生成，而这时候因为  $t_{15}$  已结执行完毕，测试  $t_1$  变为可用测试，接下来直接将改进的准深度搜索算法应用到此时的临时  $D$  矩阵即可生成最终的故障诊断策略，如图 4 所示。

由图 4 所示的结果可以看出，诊断策略由于约束条件

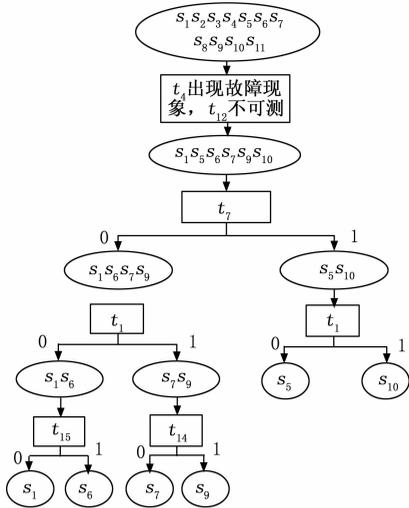


图3 案例四的动态诊断树(1)

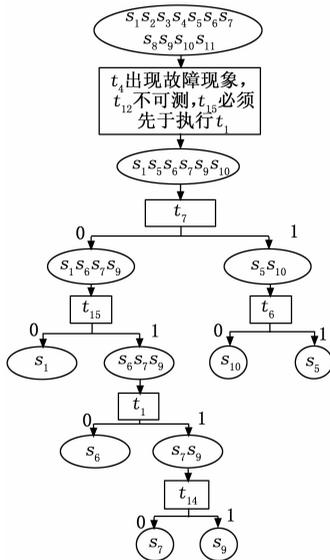


图4 案例四的动态诊断树(2)

的影响, 变得稍微复杂了一些, 但正是由于约束条件的影响, 得到的诊断策略更符合现场工况, 所以当前方法更适用于现场条件约束下的动态故障诊断策略生成。

### 5 结束语

复杂的系统或现场工作环境使得设备具有许多测试点和复杂的故障模式, 除了虚警和漏检等测试的不确定性问题, 现场条件约束更是故障诊断策略生成方法中不能忽略的问题。目的是在现场复杂工况的约束条件下, 设计一种不可靠测试下的诊断策略生成方法, 重点考虑了测试不可用, 主动选择某测试和测试之间存在顺序相关性的问题。结果在 Apollo 案例中进行了验证, 结果表明, 在允许的时间范围内, 当前算法都更加符合设备现场工作环境, 增加了故障诊断策略的有效性和实用性。这对于复杂系统和现场条件约束下的故障诊断策略设计具有重要意义。

### 参考文献:

[1] Department of Defense. MIL-STD-2165A, Military Standard Testability Program for Systems and Equipments [S]. 1993.

[2] GJB2547-95. 装备测试性大纲 [S]. 北京: 国防科工委军标出版社, 1995.

[3] 田仲, 石君友. 系统测试性设计分析与验证 [M]. 北京: 北京航空航天大学出版社, 2003.

[4] Liang Y J, Xiao M Q, Wang X. F, et al. Test Sequencing Optimization Method based on MDP under Multi-value Tests [J]. Journal of Physics: Conference Series. 2019.

[5] 张士刚. 非完美测试条件下的测试设计理论与方法研究 [D]. 长沙: 国防科学技术大学, 2013.

[6] Deng S, Bo J. Summary of integrated diagnostics and prognostics method based on testability for electronic system [J]. Control & Decision, 2013, 28 (5): 641-649.

[7] Rosales M J, Rojas A L, Chavarria S L, et al. Forecasting a stores credit plan using the Markov chains [J]. International Journal of Innovative Computing Information and Control. 2017, 13 (5): 1537-1547.

[8] Verbert K, Babuska R, Schutter B D. Bayesian and Dempster-Shafer reasoning for knowledge-based fault diagnosis—a comparative study [J]. Engineering Applications of Artificial Intelligence. 2017, 60: 136-150.

[9] Joshi P, Imadabathuni M, He D, et al. Application of the Condition Based Maintenance checking system for aircrafts [J]. Journal of Intelligent Manufacturing, 2013, 23 (2): 277-288.

[10] Pattipati K R, Deb S, Dontamsetty M. Start: System Testability Analysis and Research Tool [J]. Aerospace and Electronic Systems Magazine, 1991, 6 (1): 13-20.

[11] Pattipati K R, Raghavan V, Shakeri M. Teams: Testability Engineering and Maintenance System [A]. Proceedings of the American Control Conference [C]. Baltimore, Maryland, USA, 1994: 1989-1995.

[12] 黄以锋, 景博, 茹常剑. 基于信息熵的多值属性系统诊断策略优化方法 [J]. 仪器仪表学报, 2011, 32 (5): 1003-1008.

[13] 刘松风, 朱明初, 叶晓慧. 基于多信号模型的故障诊断算法研究 [J]. 计算机测量与控制, 2009, 17 (6): 1042-1044.

[14] Deb S, Pattipati K R, Raghavan V, et al. Multi-signal flow graph: a novel approach for system testability and fault diagnosis [J]. IEEE AES Magazine, 1995, 10 (5): 14-25.

[15] 徐辉, 梁力. 基于多信号模型的测试性分析方法研究 [J]. 计算机测量与控制, 2012, 20 (4): 914-916.

[16] 宋东, 马飞, 王传清. 一种电子系统测试性模型的研究与应用 [J]. 电子测量与仪器学报, 2010, 24 (9): 852-859.

[17] W. R. Simpson, J. W. Sheppard. System Test and Diagnosis [M]. Boston: Kluwer Academic Publishers, 1994.

[18] 王波, 姜新军, 孟上. 一种求解D-S诊断识别框架方法 [J]. 海军航空工程学院学报, 2008 (4): 89-92.

[19] Tu F, Pattipati K R. Rollout strategies for sequential fault diagnosis [J]. Systems Man & Cybernetics Part A Systems & Humans IEEE Transactions on, 2002, 33 (1): 86-99.