

基于模型故障诊断中的冲突求解

代树武, 孙辉先

(中国科学院空间科学与应用研究中心, 北京 100080)

摘要: 基于模型的故障诊断是一种重要的诊断方法, 但它的计算量较大. 在 Reiter 算法的基础上, 论证了每次求解冲突集时, 每个元件的模型知识仅需调用一次. 同时指出了在某些情况下可以利用元件参数矩阵来指导冲突的求解过程, 有效减少了调用元件模型的次数.

关键词: 故障诊断; 模型; 冲突集

中图分类号: TP270.6⁺³ **文献标识码:** A

Computing conflict sets for model-based diagnosis

DAI Shu-wu, SUN Hui-xian

(Center for Space Science and Applied Research, The Chinese Academy of Sciences, Beijing 100080, China)

Abstract: Model-based diagnosis was computationally expensive because it often used component models in reasoning process. Reiter's hitting sets tree algorithm, which did not require the conflict sets to be minimal, was one of the best model-based diagnosis method. It was proved that to find a nonminimal conflict set, every component model needed to be calculated at most once. In some situations, a component-parameter matrix can be used to direct the computing of conflict sets. All these methods reduce the times of calculating component models.

Key words: diagnosis; model; conflict sets

1 引言 (Introduction)

当工程系统的实际表现和系统模型的预期值不一致时, 表明系统没有正常工作, 至少有一个元件出现了故障, 此时需要进行故障诊断. 在各种故障诊断方法中, 由于基于模型的故障诊断方式具有能够克服专家系统的知识获取瓶颈, 能够诊断未预测的故障等特点, 受到了广泛重视.

1987 年, de Kleer^[1] 和 Reiter^[2] 同时提出了较完善的基于模型的故障诊断理论. de Kleer 提出的故障诊断方法分两步完成. 首先利用广义约束传播等方法找出系统的最小冲突集, 然后利用这些冲突集产生最小候选集. 在求解最小冲突集时, de Kleer 从系统元件的空集开始向全集方向扩展, 以保证冲突集为最小. 由于求解冲突时需要使用元件的物理模型知识, 所以这种遍历方式计算量很大, 这是它的一个缺点. 而 Reiter 从严格的理论出发建立了基于模型故障诊断的完整逻辑体系, 他采用 HS 树来求解系统的最小候选集合. Russell Greiner^[3] 等人对 HS 树的修剪算法作了改进, 使它在冲突集不为最小情况下

仍然适用. HS 树算法不需要对系统元件集合进行遍历, 诊断时它首先产生一个冲突, 然后按宽度优先方式生成 HS 树, 仅在需要的时候才调用系统模型, 产生新的冲突集, 而且这些冲突集不需要最小, 这是一个很大的优点. 然而他没有叙述如何求解冲突集, 本文讨论如何求解冲突集.

2 冲突集求解 (Computing conflict sets)

按照 Reiter 的定义, 对系统 (SD, COMPONENTS, OBS), 如果集合 $\{c_1, \dots, c_k\} \subset \text{COMPONENTS}$ 使 $\text{SD} \cup \text{OBS} \cup \{\neg AB(c_1), \dots, \neg AB(c_k)\}$ 不一致, 则集合 $\{c_1, \dots, c_k\}$ 被称为系统的冲突集. 其中 SD 为系统的模型描述, COMPONENTS 为系统的元件集合, OBS 为系统的观察值. Reiter 的算法不需要冲突集最小. 下面讨论如何求解冲突集.

假设 系统中任意一个元件 c_i 有 n 个参数, 当其中的 m 个参数已知时 ($m < n$, 且 m 固定), 利用元件模型能够求得所有其它的参数. 例如加法器是这样一个元件, 它有 3 个参数, 当已知其中的 2 个时, 而且必须已知 2 个参数才能求第 3 个. 而逻辑与

门不是这样一个元件,因为当它的一个输入参数为0时,仅需要这一个参数就可以确定元件的输出,而当它的输入参数为1时,需要两个参数都已知才能确定它的输出,即它的 m 不是固定的。

命题 对一个系统,如果它的所有元件都满足上述假设,那么在每次求解冲突集时,每个元件的模型知识最多需要利用一次就可以判断系统是否存在冲突。

证 本文的目标是寻找冲突.由假设可知,系统的元件模型知识如果利用过一次,它的所有参数都应该是已知的,当试图再次调用时,只能存在两种情况:

- a) 此时的新参数和原参数一致,只不过传播路径可能不一致,此时调用不会产生新的冲突。
- b) 新参数和原参数不一致,此时不需要利用元件模型知识进行推理就可以产生冲突。

因此,在进行冲突集求解时,虽然不同的传播路径得到的冲突集可能不同,但 HS 树方法仅仅需要求出一个冲突集或判断不存在冲突,至于是否有多个传播路径并不重要.此时,不需要处理由于传播路径不同而得到的相同结论,也就是说,每个元件只需要利用一次就可以求出一个冲突集或者判断系统一致。

基于以上分析,基于模型故障诊断方法在求解时可以建立一个元件参数矩阵,矩阵的列表示系统元件,行表示系统的各个节点参数,矩阵元素表示这个元件模型是否需要这个节点参数.当节点为元件参数时,矩阵元素为1,否则为0.以图1所示的电路系统^[1]为例,它的元件参数矩阵描述为表1.每次寻找系统的冲突集时,按以下步骤进行。

1) 建立系统元件表,它包含当前的所有元件(每次求解冲突集时,由于处在 HS 树的路径上的元件已经从整个元件集合中去除,所以每次求解时元件集不相同),本例开始时的元件表为 {M1, M2, M3, A1, A2}; 建立系统的元件参数矩阵;建立已知参数表,对本文例子,开始求解时的已知参数表如表2所示,0表示参数已知,1表示参数未知.随着求解的进行,越来越多的参数被求解,已知参数表也随之变化。

2) 从系统元件表中依次选择元件(当元件表为空时,系统一致,结束本次冲突集求解),将它所在的元件参数矩阵列和已知参数表相与(首先将参数表转化为列矩阵),结果为1的项所对应的参数未知,1的个数代表元件的未知参数个数,当未知参数的个数小于或等于元件求解限制时,调用元件模型知识对元件进行计算(当存在1时,计算结果为1的项所

对应的参数,当没有1时表明所有的参数都已知,此时可以选择任意一个参数计算,以判断是否有冲突),然后转3);当1的个数大于元件限制时,表明此时元件已知信息不够,无法求解,应该查看其它元件。

3) 检查新的结论和原结论有无冲突.有冲突时,报告冲突的传播路径,即冲突集,结束计算.无冲突时,记录新结论,修改已知参数表,将新求得的参数的标志值改为0,然后从系统元件参数表中删除该元件。

4) 检查系统元件表是否被完整的查看过一次,并且没有新结论产生,如果是这样表明此时元件表中剩余的元件缺乏已知条件,不增加测量点时不可能再产生新的冲突信息,即整个系统是一致的,退出计算.否则转2)继续求解冲突。

采用这种方法计算冲突时,产生一个冲突时每个元件模型最多需要调用一次,防止了归结反驳时的多次调用,并能保证产生冲突的完备性。

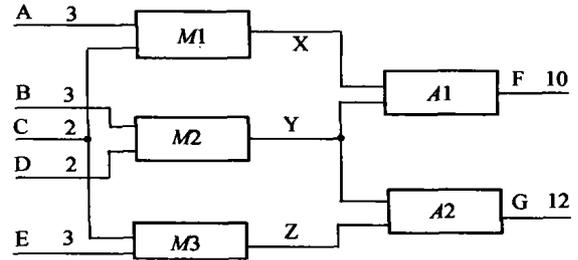


图1 系统原理图
Fig. 1 System schematic diagram

表1 元件参数矩阵

Table 1 Component-parameter matrix

	M1	M2	M3	A1	A2
A	1				
B		1			
C	1		1		
D		1			
E			1		
X	1			1	
Y		1		1	1
Z			1		1
F				1	
G					1

表2 已知参数矩阵

Table 2 Known parameter matrix

A	B	C	D	E	X	Y	Z	F	G
0	0	0	0	0	1	1	1	0	0

当求得第一个冲突集后,以这个冲突集为根节点,按照 Reiter^[2]和 Greiner^[3]的算法很容易产生 HS 树并对树进行修剪,在产生树的过程中树的每个节点上只需要找到一个冲突集即可,这个冲突集可用本文方法求解.继续这个过程可以得到完整的 HS 树,并且保证 HS 树是最小的,从 HS 树可以得到系统的诊断解.

3 讨论(Discussion)

求解系统冲突集实际是一个检验系统是否存在相互矛盾的信息的过程,如果利用不同的条件求得同一个参数存在不同的值,就可以判定系统存在冲突.采用定理证明的归结反驳方式可以求得这个系统的冲突集,然而定理证明本身是一个复杂的过程^[4].实际上采用 Reiter 算法时对冲突集的要求大大降低,它一次仅需要求解一个冲突集,而且不要求这个冲突集最小,这为冲突求解提供了方便.本文方法不需要对不同的路径进行多次计算,当所有的元件都使用过一次以后而没有发现系统有冲突,则系统一定是一致的.这有效的减少了对元件模型知识的调用.

事实上,在每次求解冲突集的过程中,不需要像前文假设那样限制系统元件的特性,不论在何种已知参数条件下,元件模型知识只需调用一次.例如对前面提到的逻辑与门,如果被调用过一次以后,下一

次试图调用时或者新参数与已知参数一致,此时不会有新信息,或者新参数与已知参数不一致,此时可直接产生冲突.但此时不能再用元件参数矩阵和已知参数矩阵来判断当前元件是否可以求解.另外,某些元件具有多个相对独立的功能,例如集成了多种功能的集成电路,此时这些器件应被分解成多个独立器件.

参考文献(References):

- [1] de KLEER J, WILLIAMS B C. Diagnosing multiple faults [J]. *Artificial Intelligence*, 1987,32(1):97-130.
- [2] REITER R. A theory of diagnosis from first principle [J]. *Artificial Intelligence*, 1987,32(1):57-95.
- [3] GREINER R, SMITH B A, WILKERSON R W. A correction to the algorithm in Reiter's theory of diagnosis [J]. *Artificial Intelligence*, 1990,41(1):79-88.
- [4] NILSSON N J. *Artificial Intelligence: A New Synthesis* [M]. Beijing: China Machine Press, 1999.

作者简介:

代树武 (1973—),男,中国科学院空间科学与应用研究中心博士后,主要从事空间飞行器自主控制研究. E-mail: daishuwu@sina.com;

孙辉先 (1946—),男,中国科学院空间科学与应用研究中心研究员,主要从事空间综合电子系统研究.