

值传递诊断过程的抽象和重用

张学农^{1),2)} 姜云飞²⁾ 陈蔼祥²⁾ 张立成¹⁾

¹⁾(广东药学院网络中心 广州 510006)

²⁾(中山大学软件研究所 广州 510275)

摘 要 提出了一种基于值传递的系统模型,重新定义了诊断,并给出了快速找到一个极小诊断的算法.在此基础上,提出了独立于系统输入输出的值传递路径的概念,实现了值传递诊断过程的抽象,并根据等价路径对系统值传递路径空间进行约简.以约简的值传递路径空间为基础,给出了一种故障诊断算法,在一定条件下多项式时间结束.

关键词 基于模型的诊断;值传递;过程抽象;过程重用

中图法分类号 TP18

DOI号: 10.3724/SP.J.1016.2009.01264

Abstraction and Reuse of Diagnosing with Value Propagation

ZHANG Xue-Nong^{1),2)} JIANG Yun-Fei²⁾ CHEN Ai-Xiang²⁾ ZHANG Li-Cheng¹⁾

¹⁾(Network Center, Guangdong Pharmaceutical University, Guangzhou 510006)

²⁾(Software Research Institute, Sun Yat-Sen University, Guangzhou 510275)

Abstract This paper presents a system model based on value propagation and redefines the diagnosis. On the basis of above works, an algorithm for finding one minimal diagnosis is brought out. Furthermore, this paper defines the concept of value propagation route for abstracting the process of diagnosis, and reduces the route space by deleting equivalence routes. Then, an algorithm to diagnosing system based on value propagation route space is introduced, which terminates in polynomial time for some special systems.

Keywords model-based diagnosis; value propagation; process abstraction; process reuse

1 引 言

基于模型诊断^[1]就是根据系统组成元件与元件之间的连接建立待诊断系统模型,利用逻辑推理理论,能够确定引发故障的元件集合.基于模型的诊断主要有两种研究方法,一个是基于一致性的诊断^[1-4],另一个是溯因诊断^[5-7].基于一致性的诊断对诊断空间要求较弱,诊断空间里会包含无用解.溯因诊断对诊断空间的限制强,但如果模型不完备,可能丢失真正的解.Console和Torasso^[8]把诊断问题看成是带有一致性约束的溯因问题,融合了两种方法.

事实上,溯因诊断空间是包含在一致性诊断空间里的.

经典模型^[1]采用一阶逻辑表示系统,描述系统的组成和行为.Chittaro^[9]等扩充了经典模型,提出了多错误分层诊断模型,经典模型中的正常和异常两种状态只是其中的特例.Baroni^[10]等提出了一种描述动态系统的模型.模型的结构部分包括元件和元件之间的连接,模型的行为部分则为对应于各部件的有限状态自动机,各有限状态自动机可以通过部件连接的事件序列进行通信.由于模型不同,其诊断的定义也与经典的诊断不同.Console^[11]等采用进程代数刻画诊断问题,给出了不同类型的物理系

统的统一描述。

为提高诊断效率, Chittaro^[9]等采用结构抽象的方法实现分层的基于模型的诊断, 取得了很好的结果。李占山^[12]等研究了基于模型的诊断问题分解及其算法。Pencole^[13]等提出了大规模离散事件系统的分解诊断方法, 并把它应用于通信网络的在线诊断。欧阳丹彤^[14]等通过引入定义信息和表示伴随关系的信息, 有效地提高了诊断的效率, 减少了候选诊断的数量。陈荣^[15]等提出了含约束的基于模型的诊断系统, 通过增加约束控制诊断空间, 并给出了选取理想约束的理论依据。Fattah 和 Dechter^[16]与 Stumptner 和 Wotawa^[17]给出了诊断树型结构系统的方法, 对于特定系统算法在多项式时间内结束。栾尚敏^[18]等给出了利用结构信息的故障诊断方法, 对于一些特殊结构的系统算法可在多项式时间内结束。Mozetic^[19]与 Childress 和 Valtorta^[20]介绍了求第 k 个诊断的方法。Marquis^[21]和 Haenni^[22]给出了计算结论求诊断的方法。Portinale^[23]等结合了基于案例的推理和基于模型的推理方法, 有效地提高了诊断的效率。Ying^[24]对诊断的知识转换和融合进行了理论研究, 为基于规则的诊断和基于模型的诊断的结合提供了理论依据和方法指导。

对于系统的实时诊断, 构造判定树非常有效。Console^[25]等提出的时态判定树在基于模型诊断的应用中得到较好的结果。Milde^[26]等也采用了判定树的方法应用于实际系统, 并提出了诊断重用的概念。Lamperti^[27]等采用基于相似性的推理技术实现诊断的重用, 并应用于大规模离散事件系统的诊断。

诊断问题是 NP 完全问题, 采用近似方法求解^[19-20, 22]是一种可行的方案, 只要能高效地得到可靠解, 并不要求其完备性。人们在实际的故障诊断过程中会直接考虑系统的值传递, 直接得到系统诊断, 往往能获得很高的效率。本文的研究尝试采用近似方法求解诊断问题, 显式地考虑系统的值传递, 以过程化的方法实现极小诊断的高效计算, 并对值传递诊断过程进行抽象以便于重用。

本文第 2 节通过两个实例说明值传递诊断方法的主要思想; 在第 3 节中给出值传递诊断方法的形式描述; 第 4 节给出快速查找一个极小诊断的算法; 第 5 节讨论值传递过程的抽象和值传递路径空间的约简; 第 6 节给出一个重用值传递路径空间的故障诊断算法; 第 7 节讨论算法的复杂性, 同时给出相应的实验结果; 第 8 节将叙述与相关研究工作的比较。

2 基于值传递故障诊断的基本思想

Reiter 提出的诊断的定义是基于逻辑一致性的^[1]。诊断过程分两步: (1) 产生所有的极小冲突集; (2) 求极小冲突集簇的极小碰集得到极小诊断。然而通过定理证明器或归结方法计算冲突集是 NP 难的, 并且冲突集的数量很大。因此, 本文采用值传递的过程化方法求极小诊断。

本节通过两个在诊断领域具有代表性的实例介绍基于值传递故障诊断方法的基本思想。在其它章节中, 仍然采用这两个实例进行说明。例 1 中的 Polybox 系统的变量取值范围是无限的, 元件的功能可以用简单代数方程来表示; 例 2 中的全加器的变量取值范围是有限的, 元件完成的是逻辑运算。

对于待诊断系统, 考虑如下两个断言:

(1) 给定静态的待诊断系统, 针对每一次观察, 系统所有元件都有一个实际的输入输出值与系统当前的输入输出相对应。

(2) 如果元件的输入输出值与其功能描述不一致, 则该元件故障。

上述断言总是成立的, 因为本文考虑的是确定系统。如果可以建立一个关于系统所有元件对应的输入输出值的合理假说, 就可以根据断言(2)得到一个诊断。最简单的方法是给所有的系统变量(系统输入输出除外)在其定义域中任意赋一个值, 然后根据元件的功能判断输入输出值与其功能描述是否一致得到一个诊断。尽管可以快速得到一个诊断(针对每个元件计算一次), 但是它往往不是极小诊断, 在极端情况下, 诊断结果是所有元件都是故障元件。这并不是我们希望得到的结果。因此, 必须改进假说的产生方法。值得注意的是, 上述方法与先提出诊断假设(假设某些元件故障, 其余元件正常), 然后根据当前观测, 通过一致性检查验证其是否是候选诊断的方法是不同的。一致性检查问题是 NP 难的, 而上述方法只需要针对每个元件完成一次计算。

系统元件的值传递可以从正向和逆向两方面考虑, 正向的值传递是根据元件的输入确定输出, 逆向的值传递是根据元件的输出和部分输入确定其它输入。为了寻找极小诊断, 首先对所有元件的状态不加限定, 然后在保证一致性的前提下, 尽可能假设系统元件正常(其余元件的状态待定), 并且根据系统当前的输入输出进行值传递推理, 直到确定了系统所有元件的输入输出值, 即得到了上述的假说。已经完

成值传递的元件,输入输出值与其功能描述是一致的,对于未进行值传递的元件,如果通过上述推理确定的输入输出值与其功能描述不一致,则认为该元件故障.这样得到的候选诊断可以合理地解释当前的观察.由于元件的值传递可以从正向和逆向两方面进行,对于同一个诊断问题,可能存在不同的推理过程,分别得到不同的候选诊断.

例 1. 考虑图 1 所示的系统:元件 $M1, M2$ 和 $M3$ 是乘法器, $A1$ 和 $A2$ 是加法器,系统输入为 $A=3, C=3, E=3, B=2, D=2, F=2$,系统输出为 $G=10, H=12$. 采用 Reiter 的方法,极小冲突集有两个: $\{M1, A1, M2\}$ 和 $\{M1, A1, A2, M3\}$, 由此产生的极小一致诊断有 4 个: $\{M1\}, \{A1\}, \{M2, A2\}, \{M2, M3\}$.

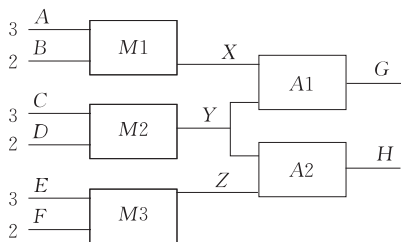


图 1 Polybox 系统

通过值传递推理,同样可以找到上述极小诊断.首先,假设 $M3$ 正常,根据 $M3$ 的输入 E 和 F 可以确定它的输出 $Z=6$;再假设 $A2$ 正常,根据 $A2$ 的输入 Z 和输出 H 可以确定它的另一个输入 $Y=6$;最后假设 $A1$ 正常,根据 $A1$ 的输入 Y 和输出 G 可以确定它的另一个输入 $X=4$.至此,所有元件的输入输出值都确定了.显然,完成了值传递的元件 $M3, A2, A1$ 的输入输出值和它的功能是一致的.考察元件 $M1$ 和 $M2$,可以确定 $M2$ 的输入输出和它的功能一致: $C=3, D=2, Y=6, C \times D=Y$;而 $M1$ 的输入输出和它的功能不一致: $A=3, B=2, X=4, A \times B \neq X$;由此可以得到一个极小诊断 $\{M1\}$.该诊断能合理地解释当前的现象:由于 $M1$ 故障,产生错误输出 $X=4, M2$ 正常,输出 $Y=6, A1$ 正常,输出 $G=10; M3$ 正常,输出 $Z=6, A2$ 正常,输出 $H=12$.

同样,由序列 $M1-M2-M3$ 进行值传递也可以确定所有元件的输入输出,考察元件 $A2$ 和 $A1$,可以得到极小诊断 $\{A1\}$.通过类似的方法也可以找到其它的极小诊断 $\{M2, M3\}$ 和 $\{M2, A2\}$.

值得注意的是,当系统输入输出改变时,可以通过同样的推理路径确定所有元件的输入输出并得到诊断.如 $A=3, C=3, E=3, B=2, D=2, F=2$,

$H=10, G=10$,由序列 $M3-A2-A1$ 可以确定所有元件的输入输出,考察元件 $M1$ 和 $M2$,可以得到极小诊断 $\{M2\}$,由序列 $M1-M2-M3$ 可以得到极小诊断 $\{A1, A2\}$,通过其它的序列分别可以得到极小诊断 $\{A1, M3\}, \{M1, A2\}$ 和 $\{M1, M3\}$.

例 2. 考虑图 2 所示的全加器诊断问题,元件 $X1$ 和 $X2$ 是异或门, $A1$ 和 $A2$ 是与门, $O1$ 是或门,系统输入 $A=1, B=1, C=0$,系统输出 $F=1, H=1$.采用 Reiter 的诊断方法,可以得到 4 个极小诊断: $\{X2\}, \{X1, A2\}, \{X1, A1\}, \{X1, O1\}$.

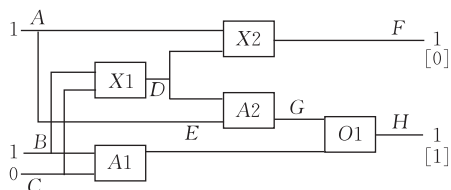


图 2 全加器

采用值传递推理仍然可以找到上述极小诊断.假设 $X1$ 正常,根据 $X1$ 的输入可以确定它的输出 $D=1$;再假设 $A2$ 正常,根据 $A2$ 的输入 D 和 A 可以确定它的输出 $G=1$;最后假设 $A1$ 正常,根据 $A1$ 的输入可以确定它的输出 $E=0$.至此,所有元件的输入输出值都确定了.考察元件 $X2$ 和 $O1$,可以确定 $O1$ 的输入输出和它的功能一致: $G \text{ or } E = H$;而 $X2$ 的输入输出和它的功能不一致: $A \text{ xor } D \neq F$;由此可以得到一个极小诊断 $\{X2\}$.同样,通过序列 $X2-A1-O1$ 的值传递推理可以得到极小诊断 $\{X1, A2\}$,序列 $X2-A2-O1$ 可以得到极小诊断 $\{X1, A1\}$,序列 $X2-A1-A2$ 可以得到极小诊断 $\{X1, O1\}$.当系统输入输出改变时,我们仍然可以通过上述推理路径得到极小诊断.

由上面分析可以看出:只要选择若干元件(不必是全部),假定它们是正常的,其余的元件状态暂时没有限定,就可以用系统的输入输出以及值传递过程,确定所有元件的输入和输出,从而决定其余的元件是否正常,由此得出诊断.在值传递过程中,每次只考虑一个元件,根据表示其功能的约束关系完成值传递.与先提出诊断假设,然后通过一致性检查验证其是否是候选诊断的方法不同,值传递方法只需局部考虑单个元件的约束关系,以判定该元件是否可以值传递和计算通过值传递确定的变量.与采用冲突集计算诊断的方法也不同,值传递方法不需要计算冲突集,而计算冲突集是 NP 难的.因此,值传递诊断方法是高效的.而且值传递过程是独立于输入和输出的,可以把这种值传递过程记录下

来,用于以后的诊断. 于是,值传递过程可以作为对待诊断系统加工后所得到的一种新的描述,当系统出现故障时直接用于诊断.

然而,有些问题必须考虑:(1)通过该方法找到的候选诊断是否是极小诊断;(2)通过该方法是否一定能得到极小诊断;(3)如何确定系统所有可能的值传递过程;(4)对于规模较大的系统,系统可能的值传递过程会很多,如何有效地进行约简.

3 值传递诊断方法的形式描述

3.1 系统表示和诊断的形式定义

Reiter^[1]采用一阶逻辑描述系统模型,适合采用逻辑方法查找冲突集.为了显式地考虑系统的值传递,本文采用不同于经典模型的系统模型,用有向图表示系统结构,值约束关系表示元件功能.

定义 1. 系统模型.

系统 Σ 的模型 M_Σ 是一个有向图 $G(V, E)$:

- (1) $V = I(\Sigma) \cup O(\Sigma) \cup COMPS(\Sigma)$;
- (2) $E = \{e_{ij}, i, j \in COMPS(\Sigma) \mid i \text{ 的输出是 } j \text{ 的输入}\}$.

其中 $I(\Sigma)$ 是系统 Σ 的输入节点集合, $O(\Sigma)$ 是系统 Σ 的输出节点集合, $COMPS(\Sigma)$ 是系统 Σ 的元件集合. 当元件 c 处于正常状态时,它的输入和输出满足反映元件功能的值约束关系 $R_{normal}(c)$, 约束变量为 x_1, \dots, x_n , 其中 x_1, \dots, x_{n-1} 表示元件 c 的输入, x_n 表示输出. $R_{normal}(c) = \{\langle a_1, \dots, a_n \rangle \mid f(a_1, \dots, a_{n-1}) = a_n \text{ 且 } a_i \in D(x_i)\}$, 其中 $D(x_i)$ 是变量 x_i 的可能取值的集合, 函数 $f: x_1, \dots, x_{n-1} \rightarrow x_n$ 表示元件 c 的功能. 当元件 c 处于故障状态时,它的输入 x_1, \dots, x_{n-1} 和输出 x_n 满足的值约束关系 $R_{abnormal}(c) = D(x_1) \times \dots \times D(x_n) - R_{normal}(c)$. E 中的每条边都有约束变量与之对应.

如图 1 所示系统的元件 $M1$, A 和 B 是输入变量, X 是输出变量, 功能是 $A \times B = X$. $M1$ 正常状态下对应的值约束关系为

$$R_{normal}(M1) = \{\langle a_1, a_2, a_3 \rangle \mid a_1 \times a_2 = a_3 \text{ 且 } a_1 \in D(A), a_2 \in D(B), a_3 \in D(X)\}.$$

由于系统是静态的, 不存在反馈结构, 所以 $G(V, E)$ 是有向无环图. 图 3 就是图 1 所示系统对应的有向图.

定义 2. 系统赋值.

系统赋值是二元组 $\langle I_V, I_E \rangle$, I_V 对系统所有元件 $c \in COMPS(\Sigma)$ 指派一个状态 $s \in \{normal,$

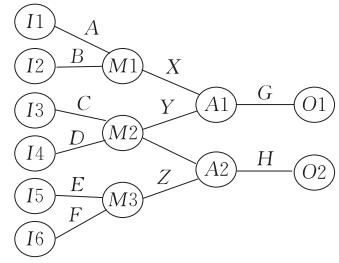


图 3 Polybox 系统对应的有向图

$abnormal\}$, I_E 对所有约束变量 x 指派一个值 $a \in D(x)$. 如果 $I_{VP} \subseteq I_V, I_{EP} \subseteq I_E$, 则称 $\langle I_{VP}, I_{EP} \rangle$ 是部分系统赋值.

定义 3. 一致的(部分)系统赋值.

系统赋值 $\langle I_V, I_E \rangle$ 是一致的, 当且仅当任意元件 $c \in COMPS(\Sigma)$, 满足 $\langle I_E(x_1), \dots, I_E(x_n) \rangle \in R_m(c)$, 其中 x_1, \dots, x_n 是元件 c 对应的约束变量, $m = I_V(c)$; 部分系统赋值 $\langle I_{VP}, I_{EP} \rangle$ 是一致的, 当且仅当任意元件 $c \in COMPS(\Sigma)$, 满足 $\langle I_{EP}(x_1), \dots, I_{EP}(x_n) \rangle \in R_m(c)$, 其中 x_1, \dots, x_n 是元件 c 对应的约束变量, $m = I_{VP}(c)$.

定义 4. 诊断问题.

诊断问题是一个三元组 (M_Σ, IN, OUT) , 其中 M_Σ 是系统模型, IN 是系统所有输入对应的约束变量的赋值, OUT 是所有系统输出对应的约束变量的赋值.

诊断问题 (M_Σ, IN, OUT) 和 Reiter^[1] 定义的 $(SD, COMPONENTS, OBS)$ 是相对应的: M_Σ 对应 SD , $COMPS(\Sigma)$ 对应 $COMPONENTS$, $IN \cup OUT$ 对应 OBS .

定义 5. 系统诊断.

系统诊断是一个一致的系统赋值 $\langle I_V, I_E \rangle$, 且对任意与系统输入对应的约束变量 x , $I_E(x) = IN(x)$, 对任意与系统输出对应的约束变量 y , $I_E(y) = OUT(y)$.

系统诊断是对系统的当前行为的一种可能解释: 如果系统元件赋值 I_V 表明所有元件正常, 则系统约束变量赋值 I_E 给出了系统的正常行为描述; 如果 I_V 表明某些元件故障, 则 I_E 给出了系统对应的行为描述: 由于某些元件故障而输出异常, 这些输出值经过其它正常元件的传播, 最终导致系统当前的异常输出.

实际上, 本文定义的诊断和 Reiter 定义的基于一致性的诊断是等价的(见定理 1): 本文的诊断是对系统的当前行为的一种合理解释, 因此在逻辑上是一致的, 反之亦然.

定理 1. $\langle I_V, I_E \rangle$ 是系统诊断当且仅当 $D = \{c \in COMPS(\Sigma) \mid I_V(c) = abnormal\}$ 是基于一致性的诊断.

证明. 对给定系统, 易知 $COMPS(\Sigma) = COMPONENTS, IN \cup OUT$ 中的元素和 OBS 中的语句一一对应.

如果 $\langle I_V, I_E \rangle$ 是系统诊断, 设 $D = \{c \in COMPS(\Sigma) \mid I_V(c) = abnormal\}$, 则对 $COMPS(\Sigma) - D$ 中的任意元件 c , 满足 $f(a_1, \dots, a_{n-1}) = a_n$, 其中 $a_i = I_E(x_i)$, x_i 是元件 c 对应的输入输出变量. I_E 使得 SD 中的语句为真, 又因为 $IN \cup OUT$ 中的元素和 OBS 中的语句一一对应, 且对任意与系统输入对应的约束变量 x , $I_E(x) = IN(x)$, 对任意与系统输出对应的约束变量 y , $I_E(y) = OUT(y)$. 所以 I_E 是 $SD \cup OBS \cup \{\neg ab(c) \mid c \in COMPONENTS - D\}$ 的一个模型, 因此 $SD \cup OBS \cup \{\neg ab(c) \mid c \in COMPONENTS - D\}$ 是一致的, 所以 $D = \{c \in COMPS(\Sigma) \mid I_V(c) = abnormal\}$ 是一个基于一致性的诊断.

如果 $D = \{c \in COMPS(\Sigma) \mid I_V(c) = abnormal\}$ 是基于一致性的诊断, 设 I_E 是 $SD \cup OBS \cup \{\neg ab(c) \mid c \in COMPONENTS - D\}$ 的一个模型, 由 (M_Σ, IN, OUT) 和 $(SD, COMPONENTS, OBS)$ 的对应关系和定义 5 可知 $\langle I_V, I_E \rangle$ 是系统诊断. 证毕.

定义 6. 极小诊断.
称系统诊断 $\langle I_V, I_E \rangle$ 是极小诊断, 当且仅当不存在系统诊断 $\langle I'_V, I'_E \rangle$, 满足 $\{c' \in COMPS(\Sigma) \mid I'_V(c') = abnormal\} \subset \{c \in COMPS(\Sigma) \mid I_V(c) = abnormal\}$.

可以把系统极小诊断 $\langle I_V, I_E \rangle$ 简化地表示为故障元件集合 $D = \{c \in COMPS(\Sigma) \mid I_V(c) = abnormal\}$. 这与 Reiter 的极小诊断表示方法是相同的.

3.2 值传递模型

定义 7. 元件的值传递.
对任意正常元件 c , 通过已知部分变量值 $x_1 = a_1, \dots, x_k = a_k$ 和关系 $R_{normal}(c)$ 确定其它变量的值 $y_1 = b_1, \dots, y_j = b_j$ 称为元件 c 的值传递, 记为 $c[(x_1/a_1, \dots, x_k/a_k) \rightarrow (y_1/b_1, \dots, y_j/b_j)]$. 即如果 $I_V(c) = normal, x_1 = a_1, \dots, x_k = a_k$, 则 $y_1 = b_1, \dots, y_j = b_j$.

根据元件的输入确定输出是正向的值传递, 根据元件的输出和部分输入确定其它输入是逆向的值传递. 例如图 1 中的元件 M1, 由 $A=3$ 和 $B=2$ 确定 $X=6$ 就是元件 M1 的值传递.

如何判定一个元件可以进行值传递和如何完成

值传递是一个实际问题. 对于可以用简单代数方程表示其功能的元件(如例 1), 可以通过方程求解来确定该未知变量. 对于例 2 中的情况, 元件的功能可以用约束关系显式地给出, 因为约束变量的取值范围是有限的. 如果与已知变量相匹配的元组唯一, 则可以确定未知的变量. 该方法适用于所有满足约束变量的取值范围是有限的情况.

例 3. 如例 2 中的与门 A1, 其变量是 B, C, E , 约束关系如表 1 所示.

表 1 与门 A1 对应的约束关系		
B	C	E
1	1	1
1	0	0
0	1	0
0	0	0

如果已知 $B=1, E=1$, 在表 1 中与之匹配的元组只有一个, 根据该元组可以确定 $C=1$; 如果已知 $C=0, E=0$, 则无法确定变量 B , 因为表中与已知变量匹配的元组有两个.

定义 8. 系统值传递.
系统值传递是一个形如 $\langle I_{VP_1}, I_{EP_1} \rangle \dots \langle I_{VP_n}, I_{EP_n} \rangle$ 的部分系统赋值序列, 满足: (1) 序列中任意两个相邻的元素 $\langle I_{VP_i}, I_{EP_i} \rangle$ 和 $\langle I_{VP_{i+1}}, I_{EP_{i+1}} \rangle$, 存在一个值传递 $c[(x_1/a_1, \dots, x_k/a_k) \rightarrow (y_1/b_1, \dots, y_j/b_j)]$, 满足 $\{x_1/a_1, \dots, x_k/a_k\} \subseteq I_{EP_i}$, 使得 $I_{VP_i} \cup \{c/normal\} = I_{VP_{i+1}}, I_{EP_i} \cup \{y_1/b_1, \dots, y_j/b_j\} = I_{EP_{i+1}}$. (2) $I_{VP_1} = \emptyset, I_{EP_1} = IN \cup OUT, EP_n = E$. 记 I_{VP_n} 为 I_{VP} , 称 $\langle I_{VP}, I_E \rangle$ 为系统值传递的结果, 其中 VP 是所有完成值传递的元件集合.

系统值传递是给定系统输入和输出的情况下, 通过元件的值传递对系统 $G(V, E)$ 的节点和边赋值的过程. 系统值传递的结果是对图 $G(V, E)$ 所有边(约束变量)的赋值 I_E 和对完成值传递的元件的赋值 I_{VP} . 值得注意的是, 在系统值传递过程中, 一致性总是得到保证的(见定理 2).

定理 2. 系统值传递 $\langle I_{VP_1}, I_{EP_1} \rangle \dots \langle I_{VP_n}, I_{EP_n} \rangle$ 中的任意元素 $\langle I_{VP_i}, I_{EP_i} \rangle$ 都是一致的.

证明. 采用归纳法, 由定义 3 和定义 8 可知, $\langle I_{VP_1}, I_{EP_1} \rangle = \langle \emptyset, IN \cup OUT \rangle$ 是一致的. 假设 $\langle I_{VP_i}, I_{EP_i} \rangle$ 是一致的, 只需证明 $\langle I_{VP_{i+1}}, I_{EP_{i+1}} \rangle$ 也是一致的.

由于 $I_{VP_i} \cup \{c/normal\} = I_{VP_{i+1}}, I_{EP_i} \cup \{y_1/b_1, \dots, y_j/b_j\} = I_{EP_{i+1}}$, 且 $\langle I_{VP_i}, I_{EP_i} \rangle$ 是一致的, 我们只需证明 $\langle I_{EP_{i+1}}(x_1), \dots, I_{EP_{i+1}}(x_n) \rangle \in R_{normal}(c)$. 由定

义 8 知, $c[(x_1/a_1, \dots, x_k/a_k) \rightarrow (y_1/b_1, \dots, y_j/b_j)]$, $\langle I_{EP_{i+1}}(x_1), \dots, I_{EP_{i+1}}(x_n) \rangle \in R_{\text{normal}}(c)$. 所以 $\langle I_{VP_{i+1}}, I_{EP_{i+1}} \rangle$ 也是一致的. 证毕.

例 4. 如图 1 所示的系统, 序列 $u_1 u_2 u_3 u_4$ 就是一个系统值传递的实例, u_4 就是系统值传递的结果. 其中:

$$u_1 = \langle \emptyset, \{A/3, C/3, E/3, B/2, D/2, F/2, G/10, H/12\} \rangle;$$

$$u_2 = \langle \{M2/normal\}, \{A/3, C/3, E/3, B/2, D/2, F/2, G/10, H/12, Y/6\} \rangle;$$

$$u_3 = \langle \{M2/normal, A1/normal\}, \{A/3, C/3, E/3, B/2, D/2, F/2, G/10, H/12, Y/6, X/4\} \rangle;$$

$$u_4 = \langle \{M2/normal, A1/normal, A2/normal\}, \{A/3, C/3, E/3, B/2, D/2, F/2, G/10, H/12, Y/6, X/4, Z/6\} \rangle;$$

$u_1 u_2$ 对应的元件值传递是 $M2[(C/3, D/2) \rightarrow (Y/6)]$;

$u_2 u_3$ 对应的元件值传递是 $A1[(G/10, Y/6) \rightarrow (X/4)]$;

$u_3 u_4$ 对应的元件值传递是 $A2[(H/12, Y/6) \rightarrow (Z/6)]$.

系统值传递的结果的含义如下: 给定系统输入输出并确定部分元件正常, 如果通过这些元件可以完成系统值传递, 则系统值传递的结果反应了系统当前的必然行为(见定理 3).

定理 3. 给定系统输入输出, 设正常元件集合为 N , I 为所有元件输入输出的“变量-实际值”对应关系集合, 如果存在系统值传递得到结果 $\langle I_{VP}, I_E \rangle$, 且 $VP \subseteq N$, 则 $I_E = I$.

证明. 通过系统值传递, 图 $G(V, E)$ 所有边的值都被正向的值传递或逆向值传递确定. 设 I_{EP} 对应正向的值传递元件完成的赋值, I_{EN} 对应逆向的值传递完成的赋值, $IN \cup OUT \cup I_{EP} \cup I_{EN} = I_E$, 并且 $I_{EP} \cap I_{EN} = \emptyset$. 按照对 I_E 的划分, 对 I 做同样的划分, I_P 对应 I_{EP} , I_N 对应 I_{EN} , $IN \cup OUT \cup I_P \cup I_N = I$, 并且 $I_P \cap I_N = \emptyset$.

由于 $VP \subseteq N$, N 为已知正常的元件集合, 易知 $I_P = I_{EP}$.

由于图 $G(V, E)$ 是有向无环图, 在逆向的值传递元件中, 存在元件 c 的输出被 OUT 赋值. 设 $I_{EN}(x)$ 是 c 进行值传递得到的输入赋值, 由于 $VP \subseteq N$, 所以 c 是正常元件, 根据元件值传递的定义, 可知 $I_N(x) = I_{EN}(x)$, 否则不可能得到当前的系统输

出. 如果直接提供上述输入的元件 c' 也是逆向的值传递元件, $I_{EN}(x')$ 是 c' 进行值传递得到的输入赋值, 通过同样的方法可以证明 $I_N(x') = I_{EN}(x')$. 通过对所有逆向的值传递元件做上述判断, 可以证明 $I_N = I_{EN}$.

由上述结论可知, $I_E = I$.

证毕.

3.3 值传递和极小诊断

接下来讨论系统的值传递和系统诊断之间的关系, 以回答是否一定能通过值传递方法找到极小诊断.

定理 4. 给定诊断问题 (M_Σ, IN, OUT) , 存在系统值传递可以得到系统值传递结果 $\langle I_{VP}, I_E \rangle$.

证明. 由于系统行为是确定的, 对于系统的任意元件, 确定其所有输入值, 就可以确定其输出值. 给定系统的输入 IN 和输出 OUT , 如果存在未被赋值的变量, 其中一定存在一个变量 x 是元件 c 的输出变量, c 的所有输入变量值都已确定, 可以通过 c 的正向值传递确定 x 的值. 因此, 总是可以通过正向的值传递确定未被赋值的变量的值, 尽管也可能通过逆向的值传递确定该变量的值. 所以定理成立.

证毕.

定理 5. 给定诊断问题 (M_Σ, IN, OUT) , 如果存在系统值传递得到结果 $\langle I_{VP}, I_E \rangle$, 则存在一个系统极小诊断 $\langle I_V, I_E \rangle$ 与 $\langle I_{VP}, I_E \rangle$ 对应.

证明. 给定诊断问题 (M_Σ, IN, OUT) , 如果存在系统值传递得到结果 $\langle I_{VP}, I_E \rangle$, 构造对未经值传递的元件的赋值 I , 满足对任意未经值传递的元件 c , 如果 $\langle I_E(x_1), \dots, I_E(x_n) \rangle \in R_{\text{normal}}(c)$, 其中 x_1, \dots, x_n 是 c 对应的约束变量, 则 $I(c) = normal$, 否则 $I(c) = abnormal$. 令 $I_V = I \cup I_{VP}$, 由系统诊断的定义和定理 2 可知 $\langle I_V, I_E \rangle$ 是一个系统诊断.

假设 $\langle I_V, I_E \rangle$ 不是系统 Σ 的极小诊断, $D = \{c \in COMPS(\Sigma) \mid I_V(c) = abnormal\}$, 则存在 $\langle I'_V, I'_E \rangle$ 也是系统诊断, $D' = \{c' \in COMPS(\Sigma) \mid I'_V(c') = abnormal\}$, 满足 $D' \subset D$. 构造与待诊断系统模型相同的系统 Σ' , D' 中的元件均为故障元件, $COMPS(\Sigma') - D'$ 中的元件均为正常元件, 显然, 给定与待诊断系统相同的输入, Σ' 会得到与待诊断系统相同的输出, 令 I'_E 为所有元件的输入输出的变量-值对应关系集合, I'_E 与 $D - D'$ 中元件的功能一致. 由于 $VP \subset COMPS(\Sigma') - D'$, 在 Σ' 中执行相同的计算过程仍会得到值传递结果 $\langle I_{VP}, I_E \rangle$, I_E 与 $D - D'$ 中元件的功能不一致. 由于 $VP \subset COMPS(\Sigma') - D'$, 由定理 3 可知 $I_E = I'_E$, 得到矛盾结果. 所以 D 是极

小诊断. 证毕.

定理 4 和定理 5 给出了系统的值传递和系统诊断之间的关系. 定理 4 表明通过值传递的方法一定可以得到系统值传递结果. 定理 5 说明通过值传递方法得到的系统值传递结果与系统的极小诊断相对应. 如定理 5 的证明过程所说, 可以根据系统值传递结果构造与之对应的极小诊断: 对 $COMPS(\Sigma)-VP$ 中的元件(未经值传递的元件), 如果其功能与 I_E 中对应的输入输出值不一致, 则该元件故障. 可以得到如下推论.

推论 1. 给定诊断问题 (M_Σ, IN, OUT) , 存在系统值传递可以得到系统极小诊断 $\langle I_V, I_E \rangle$.

证明. 由定理 4 和定理 5 可证.

至此, 本文给出了基于值传递模型的故障诊断方法的理论依据. 例 5 给出了一个基于值传递模型的故障诊断的具体过程.

例 5. 如图 1 所示系统, 序列 $u_1 u_2 u_3 u_4$ (见例 4) 是一个系统值传递, 可以得到系统值传递结果 $\langle I_{VP}, I_E \rangle$:

$$I_{VP} = \{M2/normal, A1/normal, A2/normal\};$$
$$I_E = \{A/3, C/3, E/3, B/2, D/2, F/2, G/10, H/12, Y/6, X/4, Z/6\}.$$

从而可以得到系统极小诊断 $\langle I_V, I_E \rangle$, 简记为 $D = \{M1\}$:

$$I_V = \{M2/normal, A1/normal, A2/normal, \underline{M1/abnormal}, M3/normal\};$$
$$I_E = \{A/3, C/3, E/3, B/2, D/2, F/2, G/10, H/12, Y/6, X/4, Z/6\}.$$

4 查找一个极小诊断的值传递算法

算法 1. 计算一个极小诊断.

FIND-ONE-DIAG (M_Σ, IN, OUT)

Begin

$I_V \leftarrow \emptyset$;

$I_E \leftarrow IN \cup OUT$;

$K \leftarrow$ 所有可以进行值传递的元件;

while $(K \neq \emptyset)$

do {

 任意选择 K 中一个元件 c ;

 执行 $c[(x_1/a_1, \dots, x_k/a_k) \rightarrow (y_1/b_1, \dots, y_j/b_j)]$ 并标记 c ;

$I_V \leftarrow I_V \cup \{c/normal\}$;

$I_E \leftarrow I_E \cup \{y_1/b_1, \dots, y_j/b_j\}$;

$K \leftarrow (K - \{c\}) \cup \{\text{新增可以进行值传递的元件}\}$;

}

for (每个未标记的元件 c')

do {

 if (输入输出与功能不一致)

 then $I_V \leftarrow I_V \cup \{c'/abnormal\}$ else

$I_V \leftarrow I_V \cup \{c'/normal\}$;

}

$D \leftarrow \{c \in COMPS(\Sigma) \mid I_V(c) = abnormal\}$;

return D ;

End

由于系统行为是确定的, 对于系统的任意元件, 确定其所有输入值, 就可以确定其输出值. 所以, 给定系统的输入 IN 和输出 OUT , 如果存在未被赋值的变量, 总是可以通过正向的值传递确定该变量的值, 尽管也可能通过逆向的值传递确定该变量的值. 换句话说, 如果存在未被赋值的变量, 一定存在可以进行值传递的元件. 因此, 算法 1 一定会终止并得到一个值传递结果. 可以证明如下命题.

命题 1. 如果算法 1 返回 D 非空, 则系统存在故障元件, D 为系统的极小诊断.

证明. 假设系统 Σ 的所有元件正常, I 为所有元件的输入输出的变量-值对应关系集合. 令算法得到值传递结果为 $\langle I_{VP}, I_E \rangle$. 由于 $VP \subset COMPS(\Sigma)$, 由定理 3 可知 $I_E = I$. 由于系统所有元件正常, I_E 与所有元件的功能一致, 这与返回集合 D 非空矛盾, 所以系统存在故障元件. 由定理 5 可知, D 是极小诊断. 证毕.

由上述命题可知, 如果系统异常, 算法会得到一个极小诊断. 如果返回集合 D 是空集合, 表示系统当前行为正常. 算法 1 并没有给出选择可进行值传递的元件的策略. 如果能先验地给出系统元件的故障发生概率, 可以采用可靠性高的元件优先的选择策略, 这样可以得到可能性较大的极小诊断.

5 值传递过程的抽象及路径空间约简

5.1 值传递过程抽象

算法 1 只执行了一种可行的值传递, 从而得到一个极小诊断. 如果考虑所有可能的值传递, 可以得到多个极小诊断, 这从第 3 节的例子可以看出. 进一步说, 如果记录进行值传递的元件构成一个序列, 该序列就是系统值传递过程的一种抽象. 在系统输入输出改变的情况下, 该序列可以重用. 类似的序列有许多, 给定系统输入输出, 不同的序列可能得到不同的极小诊断.

定义 9. 元件的值传递模式.

对任意正常元件 c , 通过部分变量(确定变量) x_1, \dots, x_k 和元件的值约束关系确定其它变量(被确定变量) y_1, \dots, y_j 的方式称为元件的值传递模式, 记为 $c[(x_1, \dots, x_k) \rightarrow (y_1, \dots, y_j)]$.

元件的值传递模式是元件的值传递的抽象, 而元件的值传递则是元件的值传递模式的例化, 如图 1 中由变量 A 和 B 确定变量 X 则是元件 $M1$ 的一种值传递模式.

定义 10. 系统值传递路径.

系统值传递路径 P 是一个没有重复元素的价值传递模式序列.

定义 11. 有效值传递路径.

称路径 P 为有效值传递路径, 当且仅当对除系统输入输出变量外的任意变量 z , P 满足条件: (1) 在 P 中存在唯一的元素 $c[(x_1, \dots, x_k) \rightarrow (y_1, \dots, y_j)]$, $z \in \{y_1, \dots, y_j\}$; (2) 如果值传递模式 $c'[(x'_1, \dots, x'_k) \rightarrow (y'_1, \dots, y'_j)]$ 是 P 中的元素, 且 $z \in \{x'_1, \dots, x'_k\}$, 则路径 P 中存在唯一的元素 $c[(x_1, \dots, x_k) \rightarrow (y_1, \dots, y_j)]$ 出现在元素 $c'[(x'_1, \dots, x'_k) \rightarrow (y'_1, \dots, y'_j)]$ 之前, 且 $z \in \{y_1, \dots, y_j\}$.

上述定义中条件(1)的直观含义是: 除系统输入输出变量外的任意变量都有可能的值传递确定其值; 条件(2)的直观含义是: 如果某变量作为确定另一变量的已知变量, 它必须首先被确定.

与元件值传递和价值传递模式的关系类似, 有效值传递路径可以看做系统值传递的抽象. 如果把系统值传递看做一个推理过程, 有效值传递路径就是推理过程的抽象, 为推理过程的重用提供了基础.

例 6. 如图 1 所示的系统, 路径 $P = M2[(C, D) \rightarrow (Y)] - A1[(Y, G) \rightarrow (X)] - A2[(Y, H) \rightarrow (Z)]$ 就是一条有效值传递路径. 设系统输入为 $A=3, C=3, E=3, B=2, D=2, F=2$, 系统输出为 $G=10, H=12$, 序列 $u_1 u_2 u_3 u_4$ (见例 4)就是通过路径 P 进行的系统值传递.

定义 12. 可接受输入输出集合.

通过有效值传递路径 P 可以得到值传递结果的系统输入输出 (IN, OUT) 构成的集合称为 P 可接受输入输出集合, 记为 $ACCEPT(P)$. 给定系统输入输出 (IN, OUT) , 用 $P(IN, OUT)$ 表示通过路径 P 得到的值传递结果.

定义 13. 系统值传递路径空间.

系统 Σ 的值传递路径空间 $S(\Sigma) = \{P | P \text{ 是有效值传递路径}\}$.

系统值传递路径与系统结构和功能有关, 而与

具体的值无关. 因此, 系统值传递路径空间可以在系统设计和生产过程中根据系统的设计模型生成. 然而, 对于现实系统, 系统值传递路径空间可能十分庞大, 有必要进行约简.

5.2 值传递路径空间约简

定义 14. 等价有效路径.

称两条有效路径 $P1$ 和 $P2$ 等价, 记为 $P1 \leftrightarrow P2$, 当且仅当满足条件:

(1) $ACCEPT(P1) = ACCEPT(P2)$;

(2) 对任意 $(IN, OUT) \in ACCEPT(P1)$, $P1(IN, OUT) = P2(IN, OUT)$.

直接根据定义判定两条路径等价是困难的, 因为路径可接受的输入输出集合的基数可能很大, 甚至可能是无限集合. 根据下述定理 6, 可以有效地在一组路径中寻找等价路径. 在证明定理 6 之前, 先给出有效路径对应的值传递半序集的定义.

定义 15. 有效路径的值传递半序集.

记有效路径 P 的值传递半序集为 $S_P = (E_P, <)$, E_P 是路径 P 中元素组成的集合, 称 $c[(x_1, \dots, x_k) \rightarrow (y_1, \dots, y_j)] < c'[(x'_1, \dots, x'_k) \rightarrow (y'_1, \dots, y'_j)]$, 当且仅当满足下述条件之一:

(1) $\{y_1, \dots, y_j\} \cap \{x'_1, \dots, x'_k\} \neq \emptyset$;

(2) 存在 $c''[(x''_1, \dots, x''_k) \rightarrow (y''_1, \dots, y''_j)] \in E_P$, 满足 $c[(x_1, \dots, x_k) \rightarrow (y_1, \dots, y_j)] < c''[(x''_1, \dots, x''_k) \rightarrow (y''_1, \dots, y''_j)]$, 并且 $c''[(x''_1, \dots, x''_k) \rightarrow (y''_1, \dots, y''_j)] < c'[(x'_1, \dots, x'_k) \rightarrow (y'_1, \dots, y'_j)]$.

值传递半序关系表示元件值传递必须满足的先后关系. 该定义中条件 1 的直观含义是通过元件 c 相应模式下进行值传递的部分或全部被确定变量就是元件 c' 在相应模式下进行值传递的部分或全部确定变量.

定理 6. 如果两条有效路径 $P1$ 和 $P2$ 的元素集合相等, 则 $P1 \leftrightarrow P2$.

证明. 由定义 15 可知, 如果两条有效路径 P 和 $P2$ 的元素集合相等, 则 $S_{P1} = S_{P2}$. 根据定义 14, 只需证明如果 $S_{P1} = S_{P2}$, 则 $ACCEPT(P1) = ACCEPT(P2)$ 且得到的值传递结果相等.

由定义 15 可知, 如果 $c[(x_1, \dots, x_k) \rightarrow (y_1, \dots, y_j)] < c'[(x'_1, \dots, x'_k) \rightarrow (y'_1, \dots, y'_j)]$, 则元件 c 完成值传递一定在元件 c' 完成值传递之前, 且元件 c 的值传递结果会影响元件 c' 的值传递结果; 否则, 元件 c 的值传递结果与元件 c' 的值传递结果无关. 因此, 有效路径 P 的值传递半序关系代表了通过 P 进行值传递的方式; 如果 $S_{P1} = S_{P2}$, 则通过 $P1$ 和 $P2$ 进行的值传递方式相同, 所以 $ACCEPT(P1) =$

$ACCEPT(P2)$,且对任意 $(IN,OUT) \in ACCEPT(P1)$,
 $P1(IN,OUT) = P2(IN,OUT)$. 证毕.

例 7. 如图 1 所示的系统,有效路径 $P1 = M2[(C,D) \rightarrow (Y)] - A1[(Y,G) \rightarrow (X)] - A2[(Y,H) \rightarrow (Z)]$ 和有效路径 $P2 = M2[(C,D) \rightarrow (Y)] - A2[(Y,H) \rightarrow (Z)] - A1[(Y,G) \rightarrow (X)]$ 的元素集合相等,对应的值传递半序集都是 $\{M2[(C,D) \rightarrow (Y)] < A1[(Y,G) \rightarrow (X)], M2[(C,D) \rightarrow (Y)] < A2[(Y,H) \rightarrow (Z)]\}$,因此 $P1$ 和 $P2$ 是等价路径.

定义 16. 系统值传递路径空间的约简空间.
 $S(\Sigma)$ 的约简空间 $S_{red}(\Sigma) \subseteq S(\Sigma)$,满足条件:对任意路径 $P \in S(\Sigma)$,存在路径 $P' \in S_{red}(\Sigma)$,且 $P \leftrightarrow P'$.

由等价有效路径和约简空间的定义可知,通过系统值传递路径空间中的路径得到的值传递结果和通过约简空间中的路径得到的值传递结果是相等的.系统值传递路径空间可以对应多个约简空间.根据定理 6,我们可以计算得到约简空间.约简空间在实际的诊断过程中可以重用(见第 6 节).通过下述深度优先搜索算法,可以得到约简空间.

算法 2. 构造约简空间.
ROUTESPACE (M_{Σ})
Begin
 $Space \leftarrow \emptyset$;
 $P \leftarrow \emptyset$;
CALL SEARCH (M_{Σ});

return $Space$;
End
过程 1: 路径空间搜索
SEARCH (M_{Σ})
Begin
找出所有满足条件的值传递模式 $c[(x_1, \dots, x_k) \rightarrow (y_1, \dots, y_j)]$ 的集合 K ; //条件见注 1
while (K 非空)
do{
 任意选择模式 $c[(x_1, \dots, x_k) \rightarrow (y_1, \dots, y_j)] \in K$;
 从 K 中删除模式 $c[(x_1, \dots, x_k) \rightarrow (y_1, \dots, y_j)]$;
 模式 $c[(x_1, \dots, x_k) \rightarrow (y_1, \dots, y_j)]$ 加入路径 P ;
 更新系统模型 M_{Σ} ; //条件见注 2
 ROUTESPACE (M_{Σ});
}
if (集合 $Space$ 中不存在与 P 等价的路径) then
 P 加入集合 $Space$;
End

注 1. 如果与元件 c 在相应模式下进行值传递的确定变量 x_1, \dots, x_k 对应的边直接相连的节点都是系统的输入或输出节点,则 $c[(x_1, \dots, x_k) \rightarrow (y_1, \dots, y_j)]$ 满足条件.
注 2. 在图 G 中删除元件 c 在相应模式下进行值传递的确定变量 x_1, \dots, x_k 对应的边和没有边与之相连的节点,如果 c 进行正向值传递,把 c 作为新系统的输入节点,否则 c 作为新系统的输出节点.

例 8. Polybox 系统的值传递路径空间及其约简空间的比较结果从表 2 可以看出,原空间包含 22 条路径,约简空间只包含 8 条路径.

表 2 路径空间约简

值传递路径空间	约简空间
$M3[(E,F) \rightarrow (Z)] - A2[(Z,H) \rightarrow (Y)] - A1[(Y,G) \rightarrow (X)]$	$M3[(E,F) \rightarrow (Z)] - A2[(Z,H) \rightarrow (Y)] - A1[(Y,G) \rightarrow (X)]$
$M2[(C,D) \rightarrow (Y)] - A1[(Y,G) \rightarrow (X)] - A2[(Y,H) \rightarrow (Z)]$	$M2[(C,D) \rightarrow (Y)] - A1[(Y,G) \rightarrow (X)] - A2[(Y,H) \rightarrow (Z)]$
$M2[(C,D) \rightarrow (Y)] - A2[(Y,H) \rightarrow (Z)] - A1[(Y,G) \rightarrow (X)]$	
$M2[(C,D) \rightarrow (Y)] - M3[(E,F) \rightarrow (Z)] - A1[(Y,G) \rightarrow (X)]$	
$M2[(C,D) \rightarrow (Y)] - A1[(Y,G) \rightarrow (X)] - M3[(E,F) \rightarrow (Z)]$	$M2[(C,D) \rightarrow (Y)] - M3[(E,F) \rightarrow (Z)] - A1[(Y,G) \rightarrow (X)]$
$M3[(E,F) \rightarrow (Z)] - M2[(C,D) \rightarrow (Y)] - A1[(Y,G) \rightarrow (X)]$	
$M1[(A,B) \rightarrow (X)] - A1[(X,G) \rightarrow (Y)] - A2[(Y,H) \rightarrow (Z)]$	$M1[(A,B) \rightarrow (X)] - A1[(X,G) \rightarrow (Y)] - A2[(Y,H) \rightarrow (Z)]$
$M1[(A,B) \rightarrow (X)] - M3[(E,F) \rightarrow (Z)] - A2[(Y,H) \rightarrow (Z)]$	
$M3[(E,F) \rightarrow (Z)] - M1[(A,B) \rightarrow (X)] - A2[(Y,H) \rightarrow (Z)]$	$M1[(A,B) \rightarrow (X)] - M3[(E,F) \rightarrow (Z)] - A2[(Y,H) \rightarrow (Z)]$
$M3[(E,F) \rightarrow (Z)] - A2[(Y,H) \rightarrow (Z)] - M1[(A,B) \rightarrow (X)]$	
$M3[(E,F) \rightarrow (Z)] - M1[(A,B) \rightarrow (X)] - A1[(X,G) \rightarrow (Y)]$	
$M1[(A,B) \rightarrow (X)] - M3[(E,F) \rightarrow (Z)] - A1[(X,G) \rightarrow (Y)]$	$M3[(E,F) \rightarrow (Z)] - M1[(A,B) \rightarrow (X)] - A1[(X,G) \rightarrow (Y)]$
$M1[(A,B) \rightarrow (X)] - A1[(X,G) \rightarrow (Y)] - M3[(E,F) \rightarrow (Z)]$	
$M1[(A,B) \rightarrow (X)] - M2[(C,D) \rightarrow (Y)] - A2[(Y,H) \rightarrow (Z)]$	
$M2[(C,D) \rightarrow (Y)] - M1[(A,B) \rightarrow (X)] - A2[(Y,H) \rightarrow (Z)]$	$M1[(A,B) \rightarrow (X)] - M2[(C,D) \rightarrow (Y)] - A2[(Y,H) \rightarrow (Z)]$
$M2[(C,D) \rightarrow (Y)] - A2[(Y,H) \rightarrow (Z)] - M1[(A,B) \rightarrow (X)]$	
$M1[(A,B) \rightarrow (X)] - M2[(C,D) \rightarrow (Y)] - M3[(E,F) \rightarrow (Z)]$	
$M1[(A,B) \rightarrow (X)] - M3[(E,F) \rightarrow (Z)] - M2[(C,D) \rightarrow (Y)]$	
$M2[(C,D) \rightarrow (Y)] - M1[(A,B) \rightarrow (X)] - M3[(E,F) \rightarrow (Z)]$	
$M2[(C,D) \rightarrow (Y)] - M3[(E,F) \rightarrow (Z)] - M1[(A,B) \rightarrow (X)]$	$M1[(A,B) \rightarrow (X)] - M2[(C,D) \rightarrow (Y)] - M3[(E,F) \rightarrow (Z)]$
$M3[(E,F) \rightarrow (Z)] - M1[(A,B) \rightarrow (X)] - M2[(C,D) \rightarrow (Y)]$	
$M3[(E,F) \rightarrow (Z)] - M2[(C,D) \rightarrow (Y)] - M1[(A,B) \rightarrow (X)]$	

通常,约简空间的基数远小于系统值传递路径空间的基数;假设系统元件数为 n ,有效路径的长度为 i ,则系统值传递路径空间的规模为 P_n^i ,而约简空间的规模为 C_n^i . 如果 i 和 n 接近, P_n^i 和 C_n^i 的差异较大. 从表 2 可以看出,原空间包含 22 条路径,约简空间只包含 8 条路径. 如果再加一个元件,则系统路径空间包含 61 条路径,约简空间只包含 11 条路径,具体的实验结果如表 3.

表 3 值传递路径空间和约简空间路径数的比较

系统元件数	值传递路径空间	约简空间
5	22	8
6	61	11
7	272	14
8	920	20
9	5472	26

由此可见,随着系统规模的加大,系统路径空间的生长速度远大于约简空间的生长速度,说明约简的效果是很好的.

6 利用值传递的故障诊断

获得了系统的约简空间 $S_{red}(\Sigma)$ 之后,就可以利用约简空间作为一种新的系统描述,当系统出现故障时直接用于诊断. 通过计算约简空间中的每一条路径获得极小诊断,可以得到一个基于值传递的故障诊断算法.

6.1 值传递故障诊断方法

算法 3. 基于值传递的故障诊断方法.

FIND-DIAG($S_{red}(\Sigma)$, IN , OUT)

Begin

$S \leftarrow \emptyset$;

for ($P \in S_{red}(\Sigma)$)

do {

$D \leftarrow \emptyset$;

进行值传递;

if ($((IN, OUT) \in ACCEPT(P))$) then

//如果完成值传递

do {

for ($c \in COMPS(\Sigma) - VP$)

//每个未经值传递的元件

do {

if ($(\langle I_E(x_1), \dots, I_E(x_n) \rangle \notin R_{normal}(c))$) then

$D \leftarrow D \cup \{c\}$; //其中 x_1, \dots, x_n 是元件 c

//对应的约束变量

}

$S \leftarrow S \cup \{D\}$;

}

去掉 S 中的重复元素;

return S ;

End

命题 2. 如果算法 3 返回集合 S 非空,则系统存在故障元件,且 S 中的元素都是极小诊断.

证明. 假设系统所有元件正常, I 为所有元件的输入输出的变量-值对应关系集合. 返回集合 S 非空,则存在路径 P 得到值传递结果 $\langle I_{VP}, I_E \rangle$. 由于 $VP \subset COMPS(\Sigma)$,由定理 3 可知 $I_E = I$. 由于系统所有元件正常, I_E 与所有元件的功能一致,这与返回集合 S 非空矛盾,所以如果返回集合 S 非空,则系统存在故障元件. 对任意 $D \in S$,令计算路径 P 得到值传递结果 $\langle I_{VP}, I_E \rangle$,找到诊断 D ,由定理 5 可知 D 是极小诊断.

证毕.

给定系统输入和输出,如果算法 3 返回集合 S 是空集合,则表示系统的实际输出和预期输出一致. 与算法 1 不同的是,算法 3 可以得到多个极小诊断.

例 9. 考察图 1 的 Polybox 系统,当系统输入为 $A=3, C=3, E=3, B=2, D=2, F=2$,系统输出为 $G=10, H=12$ 时,算法 3 可以得到所有的极小诊断(见表 4): $\{M1\}, \{M2, M3\}, \{A1\}, \{M2, A2\}$.

表 4 当 G 输出 10, H 输出 12 时的极小诊断

约简空间	未进行值传递的元件	极小诊断
$M3[(E, F) \rightarrow (Z)] - A2[(Z, H) \rightarrow (Y)] - A1[(Y, G) \rightarrow (X)]$	$M1, M2$	$\{M1\}$
$M2[(C, D) \rightarrow (Y)] - A1[(Y, G) \rightarrow (X)] - A2[(Y, H) \rightarrow (Z)]$	$M1, M3$	$\{M1\}$
$M2[(C, D) \rightarrow (Y)] - M3[(E, F) \rightarrow (Z)] - A1[(Y, G) \rightarrow (X)]$	$M1, A2$	$\{M1\}$
$M1[(A, B) \rightarrow (X)] - A1[(X, G) \rightarrow (Y)] - A2[(Y, H) \rightarrow (Z)]$	$M2, M3$	$\{M2, M3\}$
$M1[(A, B) \rightarrow (X)] - M3[(E, F) \rightarrow (Z)] - A2[(Y, H) \rightarrow (Z)]$	$M2, A1$	$\{A1\}$
$M3[(E, F) \rightarrow (Z)] - M1[(A, B) \rightarrow (X)] - A1[(X, G) \rightarrow (Y)]$	$M2, A2$	$\{M2, A2\}$
$M1[(A, B) \rightarrow (X)] - M2[(C, D) \rightarrow (Y)] - A2[(Y, H) \rightarrow (Z)]$	$M3, A1$	$\{A1\}$
$M1[(A, B) \rightarrow (X)] - M2[(C, D) \rightarrow (Y)] - M3[(E, F) \rightarrow (Z)]$	$A1, A2$	$\{A1\}$

如果系统输入输出改变,设系统输入为 $A=3, C=3, E=3, B=2, D=2, F=2$,系统输出为 $G=10,$

$H=10$,算法 3 仍然可以得到所有的极小诊断(见表 5): $\{M2\}, \{M1, M3\}, \{M1, A2\}, \{M3, A1\}, \{A1, A2\}$.

表 5 当 G 输出 10, H 输出 10 时的极小诊断

约简空间	未进行值传递的元件	极小诊断
$M3[(E,F) \rightarrow (Z)] - A2[(Z,H) \rightarrow (Y)] - A1[(Y,G) \rightarrow (X)]$	$M1, M2$	$\{M2\}$
$M2[(C,D) \rightarrow (Y)] - A1[(Y,G) \rightarrow (X)] - A2[(Y,H) \rightarrow (Z)]$	$M1, M3$	$\{M1, M3\}$
$M2[(C,D) \rightarrow (Y)] - M3[(E,F) \rightarrow (Z)] - A1[(Y,G) \rightarrow (X)]$	$M1, A2$	$\{M1, A2\}$
$M1[(A,B) \rightarrow (X)] - A1[(X,G) \rightarrow (Y)] - A2[(Y,H) \rightarrow (Z)]$	$M2, M3$	$\{M2\}$
$M1[(A,B) \rightarrow (X)] - M3[(E,F) \rightarrow (Z)] - A2[(Y,H) \rightarrow (Z)]$	$M2, A1$	$\{M2\}$
$M3[(E,F) \rightarrow (Z)] - M1[(A,B) \rightarrow (X)] - A1[(X,G) \rightarrow (Y)]$	$M2, A2$	$\{M2\}$
$M1[(A,B) \rightarrow (X)] - M2[(C,D) \rightarrow (Y)] - A2[(Y,H) \rightarrow (Z)]$	$M3, A1$	$\{M3, A1\}$
$M1[(A,B) \rightarrow (X)] - M2[(C,D) \rightarrow (Y)] - M3[(E,F) \rightarrow (Z)]$	$A1, A2$	$\{A1, A2\}$

例 10. 考察图 2 所示的全加器, 当系统输入 $A=1, B=1, C=0$, 系统输出 $F=1, H=1$ 时, 算法 3

得到的极小诊断如下表, 可以得到所有的极小诊断 (见表 6): $\{X2\}, \{X1, O1\}, \{X1, A1\}, \{X1, A2\}$.

表 6 当输出 $F, H=1$ 时的极小诊断

约简空间	未进行值传递的元件	极小诊断
$X1[(B,C) \rightarrow (D)] - A2[(D,A) \rightarrow (G)] - A1[(B,C) \rightarrow (E)]$	$X2, O1$	$\{X2\}$
$X1[(B,C) \rightarrow (D)] - A2[(D,A) \rightarrow (G)] - O1[(G,H) \rightarrow (E)]$	$X2, A1$	$\{X2\}$
$X1[(B,C) \rightarrow (D)] - A1[(B,C) \rightarrow (E)] - O1[(E,H) \rightarrow (G)]$	$X2, A2$	$\{X2\}$
$X2[(A,F) \rightarrow (D)] - A1[(B,C) \rightarrow (E)] - A2[(D,A) \rightarrow (G)]$	$X1, O1$	$\{X1, O1\}$
$X2[(A,F) \rightarrow (D)] - A2[(D,A) \rightarrow (G)] - O1[(G,H) \rightarrow (E)]$	$X1, A1$	$\{X1, A1\}$
$X2[(A,F) \rightarrow (D)] - A1[(B,C) \rightarrow (E)] - O1[(E,H) \rightarrow (G)]$	$X1, A2$	$\{X1, A2\}$
$A1[(B,C) \rightarrow (E)] - O1[(E,H) \rightarrow (G)] - A2[(A,G) \rightarrow (D)]$	$X1, X2$	$\{X2\}$
$O1[(H) \rightarrow (G,E)] - A2[(A,G) \rightarrow (D)]$	$X1, X2, A1$	—

注: “—”表示无法得到诊断, 因为该路径不接受当前输入输出.

当系统输入输出改变时, 设系统输入 $A=1, B=1, C=0$, 系统输出 $F=0, H=0$, 仍然可以得到

所有的极小诊断 (见表 7): $\{O1\}, \{A2\}, \{X1, X2\}$.

表 7 当输出 $F, H=0$ 时的极小诊断

约简空间	未进行值传递的元件	极小诊断
$X1[(B,C) \rightarrow (D)] - A2[(D,A) \rightarrow (G)] - A1[(B,C) \rightarrow (E)]$	$X2, O1$	$\{O1\}$
$X1[(B,C) \rightarrow (D)] - A2[(D,A) \rightarrow (G)] - O1[(G,H) \rightarrow (E)]$	$X2, A1$	—
$X1[(B,C) \rightarrow (D)] - A1[(B,C) \rightarrow (E)] - O1[(E,H) \rightarrow (G)]$	$X2, A2$	$\{A2\}$
$X2[(A,F) \rightarrow (D)] - A1[(B,C) \rightarrow (E)] - A2[(D,A) \rightarrow (G)]$	$X1, O1$	$\{O1\}$
$X2[(A,F) \rightarrow (D)] - A2[(D,A) \rightarrow (G)] - O1[(G,H) \rightarrow (E)]$	$X1, A1$	—
$X2[(A,F) \rightarrow (D)] - A1[(B,C) \rightarrow (E)] - O1[(E,H) \rightarrow (G)]$	$X1, A2$	$\{A2\}$
$A1[(B,C) \rightarrow (E)] - O1[(E,H) \rightarrow (G)] - A2[(A,G) \rightarrow (D)]$	$X1, X2$	$\{X1, X2\}$
$O1[(H) \rightarrow (G,E)] - A2[(A,G) \rightarrow (D)]$	$X1, X2, A1$	$\{X1, X2\}$

6.2 值传递故障诊断方法的改进

算法 3 通过对约简空间中的所有路径进行值传递得到了所有的极小诊断. 但是, 从上述例子可以看出, 有一些路径会得到重复的结果. 实际上, 根据极小诊断和值传递故障诊断方法的性质, 可以避免这些重复结果的计算.

引理 1. 给定系统输入和输出, 如果通过计算路径 P 可以得到极小诊断, 则只可能得到一个极小诊断.

证明. 假设可以得到两个极小诊断 D_1 和 D_2 , 由极小诊断的性质可知, $D_1 \not\subseteq D_2$ 且 $D_2 \not\subseteq D_1$, 因此, 存在元素 $c \in D_1$ 且 $c \notin D_2$. 通过计算路径 P 得到的值传递结果是唯一的, 由 $c \in D_1$ 可知, c 的输入输出和元件功能不一致, 由 $c \notin D_2$ 可知, c 的输入输出和元件功能一致, 得到矛盾的结论, 所以引理成立.

证毕.

定理 7. 给定系统 Σ 的输入和输出, 如果集合 D 是极小诊断, 路径 P 完成值传递后未经传递的元件集合为 N , 如果 D 是 N 的子集, 称 P 覆盖 D , 则通过 P 找到的极小诊断也是 D .

证明. 令系统 Σ 通过路径 P 得到的值传递结果为 $\langle I_{VP}, I_E \rangle$, 则 I_E 使得 VP 中的所有元件的输入输出与功能一致, 且 $COMPS(\Sigma) - N = VP$.

构造与待诊断系统模型相同的系统 Σ' , D 中的元件均为故障元件, $COMPS(\Sigma') - D$ 中的元件均为正常元件, 显然, 给定与待诊断系统相同的输入, Σ' 会得到与待诊断系统相同的输出, 令 I'_E 为所有元件的输入输出的变量-值对应关系集合, I'_E 与 $COMPS(\Sigma') - D$ 中元件的功能一致. 由于 $D \subseteq N$, 所以 $COMPS(\Sigma') - N \subseteq COMPS(\Sigma') - D$, 在系统 Σ 中通过路径 P 得到的值传递结果也是 $\langle I_{VP}, I_E \rangle$, 又由于 $VP \subseteq COMPS(\Sigma') - D$, 由定理 3 可知 $I_E = I'_E$, 由

此可知通过 P 在系统 Σ' 中可找到极小诊断 D . 由于 Σ' 和待诊断系统 Σ 相同, 且它们的输入输出相同, 所以通过 P 在当前系统中也可找到极小诊断 D , 又由于通过计算路径 P 只可能得到一个极小诊断, 所以定理成立. 证毕.

根据定理 7, 可以在计算过程中对得到重复结果的路径进行剪枝, 保证所得到的每个极小诊断仅被计算一次, 得到如下改进的算法.

算法 4. 改进的基于值传递的故障诊断方法.

IMPROVED-FIND-DIAG ($S_{\text{red}}(\Sigma), IN, OUT$)

Begin

$S \leftarrow \emptyset$;

for ($P \in S_{\text{red}}(\Sigma)$)

do {

$D \leftarrow \emptyset$;

if (P 不覆盖 S 中的任意元素) then 进行值传递;

if ($(IN, OUT) \in ACCEPT(P)$) then

```
//如果完成值传递
do {
  for( $c \in COMPS(\Sigma) - VP$ )
    //每个未经值传递的元素
  do {
    if ( $(\langle I_E(x_1), \dots, I_E(x_n) \rangle \notin R_{\text{normal}}(c))$  then
       $D \leftarrow D \cup \{c\}$ ; //其中  $x_1, \dots, x_n$  是元素  $c$ 
      //对应的约束变量
    }
     $S \leftarrow S \cup \{D\}$ ;
  }
}
return  $S$ ;
End
```

例 11. 考察图 1 的 Polybox 系统, 当系统输入为 $A=3, C=3, E=3, B=2, D=2, F=2$, 系统输出为 $G=10, H=12$ 时, 算法 4 的剪枝过程和得到的极小诊断如表 8.

表 8 当 G 输出 10, H 输出 12 时的剪枝过程

约简空间	未进行值传递的元素	极小诊断	剪枝标记
$M3[(E, F) \rightarrow (Z)] - A2[(Z, H) \rightarrow (Y)] - A1[(Y, G) \rightarrow (X)]$	$M1, M2$	$\{M1\}$	
$M2[(C, D) \rightarrow (Y)] - A1[(Y, G) \rightarrow (X)] - A2[(Y, H) \rightarrow (Z)]$	$\underline{M1}, M3$	$\{M1\}$	*
$M2[(C, D) \rightarrow (Y)] - M3[(E, F) \rightarrow (Z)] - A1[(Y, G) \rightarrow (X)]$	$\underline{M1}, A2$	$\{M1\}$	*
$M1[(A, B) \rightarrow (X)] - A1[(X, G) \rightarrow (Y)] - A2[(Y, H) \rightarrow (Z)]$	$M2, M3$	$\{M2, M3\}$	
$M1[(A, B) \rightarrow (X)] - M3[(E, F) \rightarrow (Z)] - A2[(Y, H) \rightarrow (Z)]$	$M2, A1$	$\{A1\}$	
$M3[(E, F) \rightarrow (Z)] - M1[(A, B) \rightarrow (X)] - A1[(X, G) \rightarrow (Y)]$	$M2, A2$	$\{M2, A2\}$	
$M1[(A, B) \rightarrow (X)] - M2[(C, D) \rightarrow (Y)] - A2[(Y, H) \rightarrow (Z)]$	$M3, \underline{A1}$	$\{A1\}$	*
$M1[(A, B) \rightarrow (X)] - M2[(C, D) \rightarrow (Y)] - M3[(E, F) \rightarrow (Z)]$	$\underline{A1}, A2$	$\{A1\}$	*

如果系统输入输出改变, 设系统输入为 $A=3, C=3, E=3, B=2, D=2, F=2$, 系统输出为 $G=10, H=10$, 算法 4 通过剪枝后实际的计算路径和得到的诊断如表 9.

表 9 当 G 输出 10, H 输出 10 时的极小诊断

实际计算路径	未进行值传递的元素	极小诊断
$M3[(E, F) \rightarrow (Z)] - A2[(Z, H) \rightarrow (Y)] - A1[(Y, G) \rightarrow (X)]$	$M1, M2$	$\{M2\}$
$M2[(C, D) \rightarrow (Y)] - A1[(Y, G) \rightarrow (X)] - A2[(Y, H) \rightarrow (Z)]$	$M1, M3$	$\{M1, M3\}$
$M2[(C, D) \rightarrow (Y)] - M3[(E, F) \rightarrow (Z)] - A1[(Y, G) \rightarrow (X)]$	$M1, A2$	$\{M1, A2\}$
$M1[(A, B) \rightarrow (X)] - M2[(C, D) \rightarrow (Y)] - A2[(Y, H) \rightarrow (Z)]$	$M3, A1$	$\{M3, A1\}$
$M1[(A, B) \rightarrow (X)] - M2[(C, D) \rightarrow (Y)] - M3[(E, F) \rightarrow (Z)]$	$A1, A2$	$\{A1, A2\}$

例 12. 考察图 2 所示全加器, 当系统输入 $A=1, B=1, C=0$, 系统输出 $F=1, H=1$ 时, 算法 4 的剪枝过程和得到的极小诊断如表 10.

表 10 当系统输出 $F=1, H=1$ 时的剪枝过程

约简空间	未进行值传递的元素	极小诊断	剪枝标记
$X1[(B, C) \rightarrow (D)] - A2[(D, A) \rightarrow (G)] - A1[(B, C) \rightarrow (E)]$	$X2, O1$	$\{X2\}$	
$X1[(B, C) \rightarrow (D)] - A2[(D, A) \rightarrow (G)] - O1[(G, H) \rightarrow (E)]$	$\underline{X2}, A1$	$\{X2\}$	*
$X1[(B, C) \rightarrow (D)] - A1[(B, C) \rightarrow (E)] - O1[(E, H) \rightarrow (G)]$	$\underline{X2}, A2$	$\{X2\}$	*
$X2[(A, F) \rightarrow (D)] - A1[(B, C) \rightarrow (E)] - A2[(D, A) \rightarrow (G)]$	$X1, O1$	$\{X1, O1\}$	
$X2[(A, F) \rightarrow (D)] - A2[(D, A) \rightarrow (G)] - O1[(G, H) \rightarrow (E)]$	$X1, A1$	$\{X1, A1\}$	
$X2[(A, F) \rightarrow (D)] - A1[(B, C) \rightarrow (E)] - O1[(E, H) \rightarrow (G)]$	$X1, A2$	$\{X1, A2\}$	
$A1[(B, C) \rightarrow (E)] - O1[(E, H) \rightarrow (G)] - A2[(A, G) \rightarrow (D)]$	$X1, \underline{X2}$	$\{X2\}$	*
$O1[(H) \rightarrow (G, E)] - A2[(A, G) \rightarrow (D)]$	$X1, X2, A1$	—	

注: “—”表示无法得到诊断, 因为该路径不接受当前输入输出.

当系统输入输出改变时,设系统输入 $A=1$, 枝后实际的计算路径和得到的诊断如表 11.
 $B=1, C=0$, 系统输出 $F=0, H=0$, 算法 4 通过剪

表 11 当输出 $F=0, H=0$ 时的极小诊断

实际计算路径	未进行值传递的元件	极小诊断
$X1[(B,C) \rightarrow (D)] - A2[(D,A) \rightarrow (G)] - A1[(B,C) \rightarrow (E)]$	$X2, O1$	$\{O1\}$
$X1[(B,C) \rightarrow (D)] - A2[(D,A) \rightarrow (G)] - O1[(G,H) \rightarrow (E)]$	$X2, A1$	—
$X1[(B,C) \rightarrow (D)] - A1[(B,C) \rightarrow (E)] - O1[(E,H) \rightarrow (G)]$	$X2, A2$	$\{A2\}$
$X2[(A,F) \rightarrow (D)] - A2[(D,A) \rightarrow (G)] - O1[(G,H) \rightarrow (E)]$	$X1, A1$	—
$A1[(B,C) \rightarrow (E)] - O1[(E,H) \rightarrow (G)] - A2[(A,G) \rightarrow (D)]$	$X1, X2$	$\{X1, X2\}$

由上述例子可以看出,在实际的计算过程中,可以对约简空间进行了剪枝,保证所得到的每个极小诊断仅被计算一次,避免了重复计算. 且值传递算法具有如下性质.

性质 1. 如果系统的实际输出和预期输出不一致,值传递算法一定会得到极小诊断集合的一个非空子集.

证明. 由定理 4、定理 5 和系统诊断的定义可以证明.

该性质对于算法的实用性很重要:事实上,为了得到真正的诊断,必须对候选诊断进行测试,可以增加观测或进行元件替换^[28]. 因此,基于模型的诊断通常被理解为不断迭代的诊断产生、诊断测试和鉴别的过程,要完成该过程,保证每次诊断计算都能得到候选诊断非常重要. 文献[29]对诊断的鉴别进行了理论性的研究.

7 分析和讨论

7.1 复杂性分析

首先考虑判定一个元件是否可以进行值传递并计算值传递所确定的变量值的复杂性. (1) 如果变量的取值范围是有限集合,基数为 h ,且元件的输入变量数为 k ,则表示元件功能的约束关系的元组数目是 k^h ,在已知部分变量的情况下,对所有元组扫描一次可以判定该元件是否可以进行值传递并得到值传递所确定的变量值. (2) 如果元件的功能可以用代数方程来表示,则可以通过方程的求解来判定该元件是否可以进行值传递并计算值传递所确定的变量值.

在上述基础上考虑算法 1(FIND-ONE-DIAG)的复杂性. 如果系统元件数为 n ,找出所有可以进行值传递的元件集合 K 需要 n 次计算,更新可以进行值传递的元件集合最多需要 n 次计算,最多需要进行 n 次值传递. 因此,算法 1 的时间复杂性是

$O(k^h n^2)$.

然后讨论算法 4(IMPROVED-FIND-DIAG)的复杂性. 设系统元件数为 n ,根据约简空间中的路径完成一次值传递,或者判断该路径不接受当前输入输出,均可在 $O(p)$ 时间内完成, p 为路径长度,显然 $p \leq n$. 然而,在无法剪枝的极端情况下,需要对约简空间中的每条路径尝试完成一次值传递. 因此,约简空间的规模决定了基于值传递的故障诊断方法的复杂程度. 由 6.2 节中的分析可知,假设系统元件数为 n ,有效路径的长度为 i ,则约简空间的规模为 C_n^i . 下面我们将讨论 i 和 n 的关系.

设系统元件数为 n ,系统有 m 个输出,系统在进行值传递之前,有 $n-m$ 个变量未被确定,因为每个元件只有一个输出变量,尽管该输出可能是多个元件的输入. 每完成一次值传递至少确定一个新的变量,确定所有变量最多需要 $n-m$ 次值传递. 也就是说,有效路径的长度 $i \leq n-m$. 所以约简空间的规模为 $C_n^{n-m} = C_n^m$. 由此可得算法 4 的时间复杂度为 $O(k^h n^{m+1})$.

实际上,很多复杂系统采用许多简单元件构造,整个系统的外部输出较少且系统变量的取值范围是有限的. 这些系统满足如下假设: (1) 元件本身的复杂程度不会随系统规模的增加而加大,即元件的输入变量数不大于常数 k ; (2) 系统的输出变量数不大于常数 m ,即系统最多有 m 个外部输出; (3) 系统变量的取值范围是有限域. 在这种情况下,判定一个元件是否可以进行值传递并计算值传递所确定的变量值可在常数时间完成,此时,算法 1 的时间复杂性是 $O(n^2)$; 算法 4 的时间复杂度为 $O(n^{m+1})$.

7.2 值传递和一致性检测

诊断假设的一致性检查是 NP 难的,为什么值传递诊断方法可以在多项式时间得到一致性极小诊断呢? 首先,只有对满足本文假设的系统,才能有多项式时间得到极小诊断. 其次,一致性在值传递诊断过程中始终是满足的. 完成一次系统值传递,我们

并没有对所有的元件限定其状态(正常或故障),而只是逐步限定参与值传递的元件为正常状态. 元件的值传递是根据其功能约束关系进行的,因此系统的值传递结果的一致性得到保证的. 对于未参与值传递的元件,我们根据其输入输出值来确定其状态,因此系统的一致性满足的. 如果首先限定所有的元件的状态,再进行一致性检查,需要从全局上考虑所有元件的约束关系. 而本文的方法只需要独立考虑单个元件的约束关系,并且参与值传递的元件只执行一次值传递,从而避免了全局的一致性检查,但这是以舍弃完备性为代价的. 因此,本文的方法是一个近似算法,是可靠的但不完备. 尽管在本文的例子中,确实得到了所有的极小诊断.

从另一个角度说,如果首先限定所有的元件的状态,即给出诊断假设,我们也可以采用值传递方法检查其一致性,但是不能保证得到结果. 通过诊断假设限定的正常元件,系统值传递可能无法完成,因而无法判定其一致性. 当然,如果根据诊断假设可以完成系统值传递,则可以判定其一致性. 因此,采用值传递方法检查诊断假设的一致性也是可靠但不完备的.

7.3 实验结果

我们在 Windows 环境的 PC 机上对值传递诊断方法进行了测试,加大本文中 Polybox 系统的规模,得到的结果如表 12 所示.

表 12 实验结果		
系统元件个数	极小诊断个数	时间/ms
5	4	2
7	6	5
9	10	15
15	20	102
20	55	277

从结果可以看出:(1)极小诊断计算时间耗费很小,包含 20 个元件的系统诊断 277ms 完成;(2)随着系统元件数的增加,极小诊断的计算时间增长较平缓. 这对诊断的实际应用是有积极意义的. 如果采用 Reiter 的方法,对 20 个元件的情况,在 PC 机上已经得不出结果. 需要说明的是,随着系统元件数的增加,值传递路径空间的计算时间增长迅速. 但是约简空间独立于系统的输入输出,只需要一次计算,在具体的诊断问题中可以重复使用.

8 相关工作比较

Reiter^[1]给出了由冲突集来求诊断的方法,诊

断的过程分两步:(1)产生所有的极小冲突;(2)求极小碰集得到极小诊断. 然而通过定理证明器计算冲突集的复杂性是指数级的,de Kleer^[30]对该方法进行了修正补充. Davis^[2],de Kleer 和 Williams^[3]使用约束传播技术实现冲突集的查找. Freitag 等^[31]指出在对整个系统进行诊断时,系统的冲突和候选诊断是通过整个系统元件传播所有观测值得到的. 整个系统的大量元件会导致巨大的计算成本. Marquis^[21]和 Haenni^[22]采用归结方法产生所有结论来求诊断.

注意到约束传播技术有如下特点:(1)约束传播可以从任意可获得测量的元件开始;(2)约束传播不一定沿着系统运算的方向,约束可以朝多个方向传播. 本文的方法吸取了约束传播技术的优点,但与通过约束传播技术进行一致性检查来计算诊断的方法不同. 本文方法的基本思想是通过值来确定元件的状态,因此始终是满足一致性要求的,因而不需计算冲突集,而直接求极小诊断,避免了进行全局一致性检查. Reiter 的方法和其后的改进方法虽然复杂性是指数级的,但它们是完备的. 本文的方法只是一种可靠的近似算法,不能保证得到所有极小诊断.

Fattah 和 Dechter^[16]、Stumptner 和 Wotawa^[17]给出了树型结构系统的多项式时间结束的诊断方法,但是他们没有把该方法推广到非树型结构系统上,这就限制了它的应用范围,因为实际系统往往是非树型结构系统. 文献^[18]给出了利用结构信息的故障诊断方法,对于一些特殊结构的系统可在多项式时间内结束. 本文的方法适合于更一般的情况,虽然不完备,但是会在多项式时间内结束,并得到极小诊断集合的一个非空子集. 有趣的是,在本文的例子中,我们得到了所有的极小诊断.

Console^[25]和 Milde^[26]等采用了判定树的方法应用于实际系统. 利用系统模型,通过穷举所有可能的系统错误,或者穷举所有可能的观察,构造判定树. 需要进行系统诊断时,仅需对判定树进行搜索即可完成. 但是,对于规模较大的系统,判定树的规模将十分庞大,因为判定树的构造不仅和系统的结构及功能有关,而且与系统输入的取值范围有关. 本文提出的值传递路径空间与系统输入的具体的值无关,并且可以得到约简空间,其规模很小. 需要进行系统诊断时,仅需对约简空间进行搜索即可完成.

Lamperti^[27]等采用基于相似性的推理技术实现诊断的重用. 本文则采用推理过程抽象的方法实现推理过程的重用,本文提出的值传递路径就是值

传递推理过程的抽象, 值传递路径与系统的输入输出值无关.

9 结 论

本文显式地考虑系统的值传递以实现极小诊断的高效计算, 给出了一种基于有向图上值传递的系统模型, 并重新定义了诊断. 在此基础上, 首先给出了找到一个极小诊断的算法, 在一定条件下, 其时间复杂度是 $O(k^h n^2)$. 然后, 给出值传递路径的概念, 并根据等价路径对系统路径空间进行约简, 同时给出了根据系统模型构造系统路径约简空间的算法. 路径约简空间与系统输入的具体的值无关, 可以作为对待诊断系统加工后所得到的一种新的描述, 当系统出现故障时直接用于诊断. 以系统的路径约简空间为基础, 给出了一种采用了剪枝技术的基于值传递的故障诊断近似算法, 时间复杂度为 $O(k^h n^{m+1})$, 对满足特定条件的待诊断系统, 该算法在多项式时间内结束. 如果系统的实际输出和预期输出不一致, 该算法一定会得到极小诊断集合的一个非空子集. 实验结果表明, 基于值传递的故障诊断方法是高效的, 随着系统元件数的增加, 极小诊断的计算时间增长较平缓. 这对诊断的实际应用是有积极意义的.

参 考 文 献

- [1] Reiter R. A theory of diagnosis from first principles. *Artificial Intelligence*, 1987, 32(1): 57-96
- [2] Davis R. Diagnostic reasoning based on structure and behavior. *Artificial Intelligence*, 1984, 24(3): 347-410
- [3] de Kleer J, Williams B C. Diagnosing multiple faults. *Artificial Intelligence*, 1987, 32(1): 97-130
- [4] Genesereth M R. The use of design descriptions in automated diagnosis. *Artificial Intelligence*, 1984, 24(1-3): 411-436
- [5] Poole D, Goebel R, Aleliunas R. THEORIST: A logic reasoning system for defaults and diagnosis//Cercione M, Calla G MC eds. *The Knowledge Frontier*. New York: Springer Verlag, 1987: 331-352
- [6] Console L, Theseider Dupre D, Torasso P. A theory of diagnosis for incomplete causal models//Proceedings of the 11th International Joint Conference on Artificial Intelligence. Detroit, 1989: 1311-1317
- [7] Cox P T, Pietrykowski T. General diagnosis by abductive inference//Proceedings of the IEEE Symposium on Logic Programming. San Francisco, 1987: 183-189
- [8] Console L, Torasso P. A spectrum of logical definitions of model-based diagnosis. *Computational Intelligence*, 1991, 7(3): 133-141
- [9] Chittaro Luca. Hierarchical model-based diagnosis based on structural abstraction. *Artificial Intelligence*, 2004, 155: 147-182
- [10] Baroni P. Diagnosis of large active systems. *Artificial Intelligence*, 1999, 110: 135-183
- [11] Console Luca, Picardi Claudia, Ribaudo Marina. Process algebras for systems diagnosis. *Artificial Intelligence*, 2002, 142: 19-51
- [12] Li Zhan-Shan, Jiang Yun-Fei, Wang Tao. The decomposition for model-based diagnosis problem and its algorithm. *Chinese Journal of Computers*, 2003, 26(9): 1177-1182(in Chinese)
(李占山, 姜云飞, 王涛. 基于模型的诊断问题分解及其算法. *计算机学报*, 2003, 26(9): 1177-1182)
- [13] Pencole Yannick. A formal framework for the decentralised diagnosis of large scale discrete event systems and its application to telecommunication networks. *Artificial Intelligence*, 2005, 164: 121-170
- [14] Ouyang Dan-Tong, Jiang Yun-Fei. Model based diagnosis of a general causal theory. *Journal of Computer Research and Development*, 1999, 36(1): 31-35(in Chinese)
(欧阳丹彤, 姜云飞. 广义因果理论的基于模型的诊断. *计算机研究与发展*, 1999, 36(1): 31-35)
- [15] Chen Rong, Jiang Yun-Fei. Model-Based diagnosis system with constraints. *Chinese Journal of Computers*, 2001, 24(2): 127-135(in Chinese)
(陈荣, 姜云飞. 含约束的基于模型的诊断系统. *计算机学报*, 2001, 24(2): 127-135)
- [16] Fattah Y E, Dechter R. Diagnosing tree-decomposable circuits//Proceedings of the International Joint Conference on Artificial Intelligence. Montreal, Canada, 1995: 572-578
- [17] Stumptner M, Wotawa F. Diagnosing tree-structured systems. *Artificial Intelligence*, 2001, 127(1): 1-29
- [18] Luan Shang-Ming, Dai Guo-Zhong. An approach to diagnosing a system with structure information. *Chinese Journal of Computers*, 2005, 28(5): 801-808(in Chinese)
(栾尚敏, 戴国忠. 利用结构信息的故障诊断方法. *计算机学报*, 2005, 28(5): 801-808)
- [19] Mozetic I. A polynomial-time algorithm for model-based diagnosis//Proceedings of the 10th European Conference on Artificial Intelligence. Vienna, 1992: 729-733
- [20] Childress R L, Valtorta M. Polynomial-time model-based diagnosis with the critical set algorithm//Proceedings of the 4th International Workshop on Principles of Diagnosis. Aberystwyth, Wales, 1993: 166-177
- [21] Marquis P. Consequence finding algorithms//Gabbay D et al. *Handbook on Defeasible Reasoning and Uncertainty Management System*. Boston: Kluwer Academic, 2000: 41-145
- [22] Haenni R. Query-driven anytime algorithm for argumentative and abductive reasoning//Proceedings of the 1st International Conference on Software. Belfast, 2002: 114-127

[23] Portinale L, Magro D, Torasso P. Multi-modal diagnosis combining case-based and model-based reasoning: A formal and experimental analysis. *Artificial Intelligence*, 2004, 158: 109-153

[24] Ying Mingsheng. Knowledge transformation and fusion in diagnostic systems. *Artificial Intelligence*, 2005, 163: 1-45

[25] Console Luca. Temporal decision trees: Model-based diagnosis of dynamic systems on-board. *Journal of Artificial Intelligence Research*, 2003, 19: 469-512

[26] Milde Heiko et al. Integrating model-based diagnosis techniques into current work processes-three case studies from the INDIA project. *AI Communications*, 2000, 13: 99-123

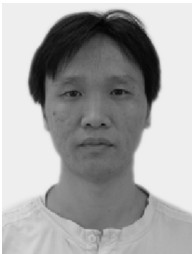
[27] Lamperti Gianfranco, Zanella Marina. Flexible diagnosis of discrete-event systems by similarity-based reasoning techniques. *Artificial Intelligence*, 2006, 170: 232-297

[28] Jiang Yun-Fei, Li Zhan-Shan. On component replacing and replacement tests for model based diagnosis. *Chinese Journal of Computers*, 2001, 24(6): 666-672(in Chinese)
(姜云飞, 李占山. 基于模型诊断的元件替换与替换测试. *计算机学报*, 2001, 24(6): 666-672)

[29] McIlraith S, Reiter R. On tests for hypothetical reasoning// Hamscher W, Console L, de Kleer J eds. *Readings in Model-based Diagnosis*. San Mateo: Morgan-Kaufmann Publishers, 1992: 89-96

[30] del Kleer, Mackworth A K, Reiter R. Charactering diagnoses and systems. *Artificial Intelligence*, 1992, 56(2/3): 197-222

[31] Freitag H, Friedrich G. Focusing on independent diagnosis problems. *Annals of Mathematics and Artificial Intelligence*, 1994, 11: 329-349



ZHANG Xue-Nong, born in 1973, Ph. D.. His main research interests include automated diagnosis and automated planning.

JIANG Yun-Fei, born in 1945, professor, Ph. D. super-

visor. His main research interests include automated reasoning, automated diagnosis and automated planning.

CHEN Ai-Xiang, born in 1979, Ph. D.. His main research interests include automated planning and automated diagnosis.

ZHANG Li-Cheng, born in 1975. His main research interest is automated diagnosis.

Background

Automated diagnosis is an active field of AI research. A diagnostic problem is characterized by a set of observations to account for the differences between the system’s actual behaviors and its expected behaviors. Model-based diagnosis addresses the systems whose nominal behaviors can be specified as a mapping from their input variables to their output variables. The classical method is built on the well-known consistency-based theory and the classical model describes the

system’s structure and behavior usually by the first-order language. This paper presents a system model based on value propagation and redefines the diagnosis. On the basis of the concept of value propagation route for abstracting the process of diagnosis, an algorithm to diagnosing system is introduced, which terminates in polynomial time for some special systems. This work is supported by the Natural Science Foundations of China under grant Nos. 60773201, and 60173039.