

结合问题特征利用 SE-Tree 反向深度求解冲突集的方法

欧阳丹彤^{1,3}, 刘伯文^{1,3}, 周建华^{2,3}, 张立明^{1,3}

(1. 吉林大学计算机科学与技术学院, 吉林长春 130012; 2 吉林大学软件学院, 吉林长春 130012;

3. 符号计算与知识工程教育部重点实验室(吉林大学), 吉林长春 130012)

摘要: 基于模型诊断是人工智能领域内的一个重要研究方向,求解极小冲突集在基于模型诊断中有着重要应用. 在对结合 CSISE-Tree 求解冲突集方法深入研究的基础上,根据冲突集求解特征重构了结合枚举树的计算冲突集的过程,提出基于深度优先反向搜索求解冲突集的方法. 针对 CSISE-Tree 方法求解时占用内存空间与元件总数指数级相关的缺点,构建反向深度搜索方法减小求解时所占用内存空间;针对 CSISE-Tree 方法不能对部分非极小的冲突集进行剪枝的问题,给出对非冲突集和更多非极小的冲突集进行剪枝的方法,有效减少了求解时调用 SAT(Boolean SATisfiability problem)求解器的次数;实验结果表明,与 CSISE-Tree 方法相比,本文提出的方法求解效率有明显的提升,并避免了求解时的内存爆炸问题.

关键词: 基于模型诊断; 冲突集; 布尔约束可满足; 集合枚举树

中图分类号: TP306 **文献标识码:** A **文章编号:** 0372-2112 (2017)05-1175-07

电子学报 URL: <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2017.05.021

A Method of Computing Minimal Conflict Sets Combining the Structure Property with the Anti-depth SE-Tree

OUYANG Dan-tong^{1,3}, LIU Bo-wen^{1,3}, ZHOU Jian-hua^{2,3}, ZHANG Li-ming^{1,3}

(1. College of Computer Science and Technology, Jilin University, Changchun, Jilin 130012, China;

2. College of Software, Jilin University, Changchun, Jilin 130012, China;

3. Key Laboratory of Symbolic Computation and Knowledge Engineering (Jilin University), Ministry of Education, Changchun, Jilin 130012, China)

Abstract: Model-based diagnosis is an important problem in the field of artificial intelligence. In model-based diagnosis, how to get the minimal conflict sets is a well-known problem with extensive applications. In this paper, according to the characteristics of the conflict sets, we use the enumeration tree to reconstruct the process of solving conflict sets and then design a reverse depth algorithm based on the previous algorithm CSISE-Tree. Firstly, this proposed reverse depth search algorithm can reduce as many memory spaces as possible when obtaining some conflict sets, while CSISE-Tree have to expend some unnecessary memory spaces in this case, where the consume of memory spaces exponentially grows with the number of circuit elements. Secondly, compared with CSISE-Tree, our algorithm can effectively cut down the number of calling the SAT(Boolean SATisfiability problem) solver by pruning some non-minimal conflict sets and non-conflict sets. The experimental results show that our algorithm performs better than its competitor CSISE-Tree in terms of run time in most instances. More importantly, our algorithm avoids the memory explosion when solving some large instances.

Key words: model-based diagnosis; conflict set; Boolean SATisfiability problem (SAT); set-enumeration tree(SE-Tree)

1 引言

基于模型诊断 (Model-based Diagnosis, MBD) 是人工智能领域内的重要的研究方向之一,对整个人工智能领域的发展起着十分重要的作用^[1].著名诊断学者 de Kleer 首次提出了冲突集的概念^[2],对基于模型诊断的求解起到了巨大的推动作用,在人工智能其它领域也得到了广泛的应用.

著名 MBD 专家 Reiter 指出求解极小冲突集 (Minimal Conflict Set, MCS) 和求解极小冲突集的极小碰集 (Hitting-Set, HS) 是求解最终诊断结果的两个重要步骤^[3].随后有许多学者参与到求解极小冲突集与求解极小碰集的研究中^[4-6],出现了许多求解冲突集的算法.早期冲突集求解方法主要有利用定理证明器的方法 DART^[7]和 Haenni^[8]使用归结的方法.国内也有许多学者对冲突集的求解做过研究,如栾尚敏等给出的用系统结构信息来求解极小冲突集^[9],代树武等人利用元件参数矩阵求解极小冲突集^[10].Feldman 等^[11]学者指出冲突集和诊断集间存在二元关系,即冲突集的碰集是诊断,而诊断的碰集是冲突集,并给出了基于冲突集和诊断集二元性的诊断求解方法.Amgoud 等^[12]学者则将冲突集用于基于逻辑的辩论系统.

20 世纪 90 年代,Hou 提出了 CS-Tree^[13]方法,能够利用枚举树求解极小冲突集,然而此方法由于使用剪枝策略在某些情况下使得某些极小冲突集丢失,之后学者 Han 和 Lee 提出了 Inverse CS-Tree、CS-Tree with Mark Set^[14],对 CS-Tree 进行了改进,然而枚举树中仍然存在大量冗余节点没有被剪除.

近几年,SAT 求解器的效率得到了大幅提升,人工智能领域中许多问题也都转化为 SAT 问题进行求解^[15,16].国内赵相福学者提出了求解冲突集的 CSSE-Tree^[17]与 CSISE-tree^[18]方法,通过调用 SAT 求解器求解极小冲突集.在判断枚举树中的节点是否为冲突集时假设此节点包括的所有元件都正常,将电路的合取范式 (Conjunctive Normal Form, CNF)、观测、正常元件的行为描述合并为一个 CNF 文件,调用 SAT 求解器判断 CNF 的可满足性.如果 CNF 不可满足,代表此元件集为冲突集.CSSE-Tree 方法利用广度优先遍历生成正向枚举树,从根节点开始遍历,同时基于冲突集的除自身外的超集一定不是极小冲突集的原理,对枚举树进行了剪枝;CSISE-Tree 方法利用广度优先遍历,以所有元件集合为根节点生成一个反向枚举树,基于非冲突集的子集一定不是冲突集的原理,对枚举树进行了剪枝,同时 CSISE-Tree 方法利用极小冲突集的性质,提出并结合标识规则^[18],直接求得所有极小冲突集.两种方法都

成功减少了求解极小冲突集的时间,CSISE-Tree 方法求解效率比 CSSE-Tree 方法更高.

本文在对 CSSE-Tree 和 CSISE-Tree 方法分析的基础上,对求解冲突集方法进行改进.CSISE-Tree 方法主要对非冲突集进行剪枝,CSSE-Tree 方法只对非极小的冲突集进行剪枝.针对上述两种方法未能充分剪枝的问题,提出冲突集求解新方法 CSRDSSE (CS ReverseDepth SE-Tree, CSRDSSE),在对非冲突集进行剪枝的同时,尽可能的剪除部分能够确定的非极小的冲突集,从而提高求解极小冲突集的效率.

2 预备知识

本节主要介绍诊断问题的基本定义以及如何将诊断问题转化为 SAT 问题.

定义 1^[3]:诊断问题被定义为三元组 $DS = (SD, COMPS, OBS)$,其中:SD 是谓词公式的集合,是系统的描述;COMPS 是一个有限的常量集,是系统元件的集合;OBS 是谓词公式的有限集,是观测的集合.

定义 2^[3]:设元件集的一个子集 $\Delta \subseteq COMPS$,如果: $SD \cup OBS \cup \{-AB(c) \mid c \in COMPS - \Delta\} \cup \{AB(c) \mid c \in \Delta\}$ 是可满足的,则 Δ 是关于 $(SD, COMPS, OBS)$ 的一个基于一致性诊断.

定义 3^[3]:冲突集是一个部件集 $\{c_1, c_2, \dots, c_n\} \in COMPS$,使得 $SD \cup OBS \cup \{-AB(c_1), -AB(c_2), \dots, -AB(c_n)\}$ 不一致.

称某个冲突集为极小冲突集,当且仅当该冲突集的任意真子集都不是冲突集.

SAT 求解器的输入为 CNF 文件,在诊断问题的电路系统描述中,每一个元件与观测均可转化为 CNF 的格式,将转化的 CNF 子句存储到文件后,便将一个诊断问题转化为 SAT 问题.

以上部分给出了电路与 SAT 问题的转化关系,接下来给出使用 SAT 求解器求解极小冲突集的方法.

3 CSISE-Tree 方法

根据第 2 节中介绍的基本知识,使用 SAT 求解器求解极小冲突集过程时,首先将给定的元件集与观测转化为 CNF 格式存储到文件中.然后结合集合枚举树在文件末尾追加所有可能故障元件的判断描述,最后调用 SAT 求解器对 CNF 子句进行判断.若 CNF 不可满足,则此判断对应的元件集为冲突集,否则此元件集不是冲突集.

3.1 ISCS 方法求解冲突集

ISCS 方法用来判断电路中某个元件集对应的 CNF 是否可满足,即是否为冲突集,算法 1 为此方法的伪代码.

算法 1 ISCS 方法求解冲突集伪代码

```
function ISCS( obs[ ], sub[ ], filename) void;
begin
1  initialize( );
2  addtoconf( obs[ ] );
3  addtoconf( sub[ ] );
4  fp ← fopen( filename, "r" );
5  jud ← picosat( fp );
6  fclose( fp );
7  if( jud = 0 )
8      return 1;
9  else
10     return 0;
end
```

算法 1 中第 1 步主要对算法进行初始化. 第 2 步将观测的单元子句追加到目标 CNF 文件中. 第 3 步是将组件的单元子句追加到目标 CNF 文件中. 第 4 ~ 5 步读取 CNF 文件, 并且调用 Picosat^[19] 求解器对 CNF 的可满足性进行判断. 算法第 7 ~ 10 步, 如果当前判断的组件集不可满足, 则此组件集是冲突集, 返回 1; 否则, 此组件集不是冲突集, 返回 0.

利用枚举树求解极小冲突集的基本思想如下: 枚举树上每一个点为一个组件集. 依次访问枚举树上的每一个节点, 调用 ISCS 方法判断此节点是否为冲突集, 如果是冲突集, 将此组件集加入到冲突集节点集合中. 枚举完所有节点后, 在所有冲突集中找出极小冲突集.

接下来介绍一个带有剪枝策略的求解所有极小冲突集的方法.

3.2 CSISE-Tree 方法求解极小冲突集

CSISE-Tree^[18] 方法是通过搜索翻转的集合枚举树, 对枚举树进行剪枝, 然后调用 ISCS 方法并且结合标识规则求解极小冲突集的方法.

首先介绍翻转的集合枚举树 ISE_Tree, 其形式化描述图如图 1.

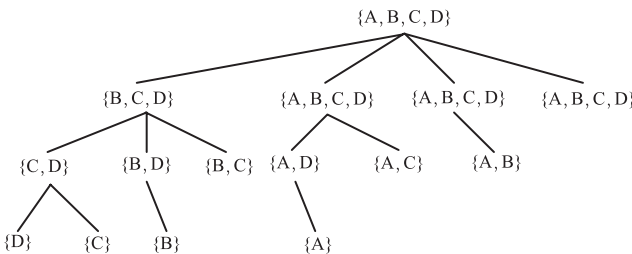


图1 翻转的集合枚举树 ISE-Tree

CSISE-Tree 方法利用冲突集的特征与枚举树的结构对 ISE_Tree 进行剪枝, 从而减少对许多不必要的节

点的扩展.

CSISE-Tree 方法的修剪规则如下:

(1) 宽度优先生成 ISE_Tree, 从右向左搜索集合枚举树. 当某个节点不是冲突集时, 根据此节点的子集一定不是冲突集原理, 对此节点停止扩展.

(2) 在扩展子节点时, 首先判断将要扩展的子节点是否为右侧某个非冲突集的子集, 若为某个非冲突集的子集, 则其必然为非冲突集, 不对其进行扩展.

CSISE-Tree 方法的标识规则如下:

若被搜索的节点 N 不可满足, 则把此节点标识为 0, 把所有已经标识为 0 并且为节点 N 的超集的节点标识为 1.

CSISE-Tree 方法的优点:

(1) 通过两条修剪规则, 可以剪掉许多不是冲突集的节点, 减小了搜索空间, 节省了计算时间.

(2) 通过标识规则, 充分利用了冲突集的性质, 能够直接得到极小冲突集.

CSISE-Tree 方法无法剪除部分非极小的冲突集, 当一些极小冲突集在树的上层时, 剪枝规则所剪掉的节点数会下降. 通过对 CSISE-Tree 方法的分析, 提出了基于深度优先反向搜索正向枚举树的方法, 此方法既可以对部分非冲突集进行剪枝, 也可以对部分非极小的冲突集进行剪枝, 进而提高算法的效率.

4 CSRDSE 方法

在上一节介绍了 CSISE-Tree 方法求解极小冲突集的基本思想, CSISE-Tree 方法虽然优化了求解空间, 但是 CSISE-Tree 方法的空间复杂度与元件个数成指数级相关, 并且无法剪掉部分非极小的冲突集. 针对此问题, 提出基于深度优先的反向深度搜索方法 CSRDSE. 下面给出 CSRDSE 方法的相关定义、基本思想和相关证明.

4.1 基本思想

首先, 给出如下与冲突集相关的推论:

设元件集 $C = \{c_1, c_2, \dots, c_n\} \subseteq COMPS$, 根据冲突集的定义, 可以得出下面推论:

推论 1^[18] 若元件集 C 是冲突集, 则 C 除自身外的任意的超集都不是极小冲突集.

推论 2 若元件集 C 不是冲突集, 则 C 的任意子集都不是冲突集.

推论 3^[18] 元件集 B, C 是冲突集, 若元件集 B 是 C 的真子集, 则 C 不是极小冲突集.

下面给出集合枚举树 SE_Tree 和 CSRDSE 方法的相关定义.

定义 4(反向深度搜索) 对于给定的集合枚举树 SE_Tree, 称从最底层节点向枚举树的根节点反向深度优先遍历枚举的过程为反向深度搜索.

定义 5(分支子树) 设 N 为 SE_Tree 中节点, 以节点 N 为根节点及其下面的所有节点构成的子树称为以 N 为根节点的分支子树。

在图 2 SE_Tree 中, 以 $\{A, B\}$ 为根节点分支子树如图 3 所示。

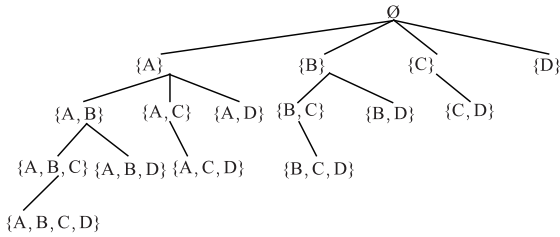


图2 正向集合枚举树SE-Tree

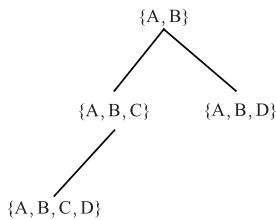


图3 分支子树

定义 6(兄弟节点) 设节点 M, N 分别为 SE_Tree 中的不同节点, 如果 M, N 的拥有同一个父节点, 则称 M, N 互为兄弟节点。

定义 7(最左分支) 分支子树中最左侧的分支称为最左分支。

定义 8(最左节点) 分支子树中最左分支上最底层的节点称为此分支子树的最左节点。

例 1 图 2 中 $\{A, B\}, \{A, C\}, \{A, D\}$ 有公共父节点 $\{A\}$, 则他们互为兄弟节点。

例 2 图 3 中以 $\{A, B\}$ 为根节点的分支子树的最左分支为:

$\{A, B\} \rightarrow \{A, B, C\} \rightarrow \{A, B, C, D\}$. 最左节点为:
 $\{A, B, C, D\}$.

根据反向深度搜索的思想, 此方法从 SE_Tree 最左节点向根节点枚举求解, 设节点 N 为 SE_Tree 中当前遍历的节点. 下面给出枚举求解中用于剪枝的三条修剪规则。

剪枝规则 1 若 N 是冲突集, 且 N 是其父节点对应最左分支上的节点, 则由节点 N 反向深度向上回溯搜索, 并且当回溯到的节点 M 满足下述条件之一时停止向上回溯: ①节点 M 不再是其父节点对应最左分支上的节点; ②节点 M 的父节点不是冲突集. 此时, 剪掉节点 M 下的所有子节点, 按照反向深度搜索的规则跳转到下一个需要判断的节点。

剪枝规则 2 若节点 N 不是冲突集, 剪掉节点 N 对

应的所有祖先节点, 并将节点 N 存储到非冲突集的集合中, 按照反向深度搜索的规则跳转到下一个需要判断的节点。

剪枝规则 3 若 N 为非冲突集的集合中某个非冲突集的真子集, 则直接得到 N 不是冲突集, 按剪枝规则 2 对节点 N 进行剪枝并继续搜索其它节点。

标识规则^[18]: 设节点 N 为本次遍历的节点, 且为冲突集, 将冲突集 $\{N\}$ 标识为“0”, 同时对当前冲突集的集合中其它每一个标识为“0”的冲突集做如下操作: 若冲突集 $\{N\}$ 为此冲突集的子集, 则将此冲突集标识为“1”, 表示此冲突集不是极小冲突集。

4.2 CSRDSE 方法

在上一小节介绍了 CSRDSE 方法的基本思想, 算法 2 是结合修剪规则和标识规则 CSRDSE 算法的伪代码。

算法 2 结合修剪规则和标识规则 CSRDSE 算法的伪代码

```
function CSRDSE(inputf ilename, obs[ ]) void;
begin
1 initialize();
2 while  $N \neq \phi$  do
3   notCS ← isunsolve( $N$ );
4   if notCS = 0 then
5     judge ← ISCS(obs[ ],  $N$ );
6     if judge = 0 then
7       addtosolve( $N$ );
8       if  $N$  不是从底层向上回溯后的节点 then
9          $N \leftarrow N$  的下一个要判断叶子节点;
10      else
11         $N \leftarrow$  以  $N$  的左子节点的下一个兄弟节点为根的分支子树的最左节点;
12      else
13        addtosolve( $N$ );
14        if  $N$  是以  $N$  的父亲节点为根的分支子树上的最左分支上的节点 then
15           $N \leftarrow N$  的父亲节点;
16        else if  $N$  是从底层向上回溯后的节点 then
17           $N \leftarrow$  以  $N$  的下一个兄弟节点为根的分支子树的最左节点;
18        else
19           $N \leftarrow N$  的下一个叶子节点;
20        else
21          if  $N$  不是从底层向上回溯后的节点 then
22             $N \leftarrow N$  的下一个叶子节点;
23          else
24             $N \leftarrow$  以  $N$  的子节点的下一个兄弟节点为根的分支子树的最左节点;
25        out(solve[ ]);
26      return;
end
```

函数 `addtounsolve(int [])` 功能是将当前部件集加入到非冲突集集合中。

函数 `addtosolve(int [])` 功能是根据标识规则将当前部件集加入到冲突集集合中。

算法 2 的输入为 CNF 文件名与观测,在第 1 步,对算法进行初始化,包括对非冲突集集合和冲突集集合初始化为空等。第 2 步进入循环,直到枚举整个 SE_Tree 上所有节点后退出循环。在第 3 步, `isunsolve(int [])` 函数是判断当前部件集 N 是否为已求得非冲突集的子集,若是子集则返回 1,否则返回 0。根据剪枝规则 3,若其返回值为 1,则 N 必定不是冲突集,进入第 4 步,否则进入算法第 20 步。第 4、5 步表示 N 不为已求得非冲突集的子集,因此调用 ISCS 算法判断 N 是否为冲突集。第 6~11 步,如果当前部件集 N 不是冲突集,调用函数 `addtounsolve(N)`,将 N 加入到非冲突集集合中,然后根据剪枝规则 2 跳转至下一个需要判断的节点。第 12~19 步,调用函数 `addtosolve(N)` 将部件集 N 加入到冲突集集合中,并且根据标识规则更新所有冲突集的标识。然后根据修剪规则(1)跳转至下一个需要判断的节点。第 20~24 步, N 是已知某个非冲突集的子集,因此 N 不是冲突集,根据修剪枝规则 2 跳转至下一个需要判断的节点。25、26 步,如果 N 为空,所有节点判断完毕,输出最后结果,算法结束。

4.3 算法的分析

剪枝规则的正确性:

剪枝规则 1: 设 M 是从节点 N 按反向深度将回溯到的节点,且 M 是冲突集。由 SE_Tree 特征可得,其下面所有子孙节点都是 M 的超集。根据推论 1,所有子孙节点都不是极小冲突集,因此可以被剪掉。

剪枝规则 2: 若节点 N 不是冲突集,其所有祖先节点为节点 N 的真子集。根据推论 2,其所有祖先节点都不是冲突集,因此可以被剪掉。

剪枝规则 3: 如果节点 N 为某个非冲突集的子集,根据推论 2,节点 N 一定不是冲突集,则节点 N 可以被剪掉。

完备性: 此方法按照特定的反向深度顺序搜索 SE_Tree 中所有节点,因此可以产生所有冲突集。剪枝规则 1 并不会剪掉极小冲突集,剪枝规则 2 和 3 不会剪掉冲突集。因此,若极小冲突集存在,必然在此方法所求得的冲突集集合中。结合标识规则,可以得到所有的极小冲突集。

CSRSE 方法的优点:

(1) 从底部开始搜索 SE_Tree,会优先搜索到包含元件数目较多的非冲突集集合,因此能够剪除较多的非冲突集。

(2) 当搜索到 SE_Tree 上层节点时,若节点 N 是冲

突集,且将回溯到节点 M 不是冲突集。此时,剪掉节点 N 下面的大量子孙节点。

(3) 采用标识规则,当算法结束时可以直接得到极小冲突集。

(4) 此方法采用基于深度优先的底层向上回溯的搜索方法,不需要存储接下来待遍历的节点,CSRSE 方法仅存储的当前求解的节点相关信息,占用内存资源远小于 CSISE-Tree 方法。

5 实验结果

实现了赵相福学者提出的 CSISE-Tree 方法与本文提出的 CSRSE 方法,并将两个方法的效率进行对比。实验平台如下: Dell Dimension C521, Ubuntu 12.04 LTS, GCC 编译器, AMD Athlon(tm) 64 X2 Dual Core Processor 3600+, 1.90GHz, 3GB RAM。CSISE-Tree 方法与 CSRSE 方法均采用了数组作为基本存储结构,同时调用了 SAT 求解器均为 Picosat^[19]。

本次测试的例子包括 C17、Fulladder_1、Fulladder_2、Polybox_5、Polybox_9、Fulladder_3、Fulladder_4、Fulladder_8。其中电路 C17、Fulladder_1、Fulladder_2、Polybox_5、Polybox_9 电路是源于文献[18]中的实验电路。电路 Fulladder_3、Fulladder_4、Fulladder_8 是在以上电路基础上进行扩充得到的电路。实验对每一个电路挑选了大量观测进行测试,同时每组观测都存在冲突。

表 1 给出了 CSISE-Tree 方法与 CSRSE 方法的对比,其中第 2、3 列为平均访问节点个数,第 4 列为平均访问节点提高百分比,第 5 列为 CSRSE 方法与 CSISE-Tree 方法最优提高百分比。从表 1 可以看出,CSRSE 方法比 CSISE-Tree 方法的求解效率有明显的提升,因为对于非冲突集的剪枝,两个方法剪枝效果基本上持平。但 CSRSE 方法又加入了对部分其它非极小的冲突集的剪枝,因此效率得到提升。从表 1 中最后一列可以发现,对于部分观测 CSRSE 方法的效率能够提升到 90% 以上。此观测对应的极小冲突集长度比较短,处于 SE_Tree 的上面几层,CSRSE 方法可以沿着最左分支向上回溯到其父亲节点不是冲突集的节点,剪除此节点下面大量的非极小的冲突集,因此 CSRSE 方法的效率相比于 CSISE-Tree 方法得到显著的提升。

表 2 给出了求解时间和占用空间的对比,其中第 2 列为每个测试用例测试对应的观测(每组观测都有冲突集)数目,第 3、4 列为两种方法求解所用时间,第 5、6 列为两种方法求解时访问节点队列的长度(设每个元件集长度为 1)。通过表 2 可以得出,当元件个数较少的时候,两种方法的 CPU 耗时几乎相同,随着元件数目的增加,CSRSE 方法的 CPU 耗时明显优于 CSISE-Tree 方法。随着元件数目增加,SAT 求解器的每次求解耗时增

加,同时非冲突集的数目增加.在 CSISE-Tree 方法中,对于每个要判断的节点,都需要对当前已有的非冲突集进行超集判断,此判断耗时是指数级的;而在 CSRSE 方法中,剪除的部分非极小的冲突集不再需要进行 SAT 判断与超集判断,节省了较多求解时间.因此随着元件数目的增加 CSRSE 方法的 CPU 耗时明显优于 CSISE-Tree 方法.

表 1 平均访问节点个数和效率提高情况

测试用例	平均调用 SAT 次数		提高情况	
	CSRSE	CSISE-Tree	平均提高	最优提高
C17	16.4375	17.4271	0.0568	0.388889
Fulladder_1	11.042	11.75	0.0603	0.157895
Fulladder_2	96.469	237.902	0.5945	0.926357
Polybox_5	14.362	14.847	0.0327	0.142857
Polybox_9	151.772	162.236	0.161616	0.645
Fulladder_3	383.964	961.295	0.6006	0.977583
Fulladder_4	7542.589	10097.527	0.2530	0.994864
Fulladder_8	53303.617	88513.085	0.3978	0.999466

表 2 算法的求解时间和求解占用空间对比

测试用例	测试个数	求解总时间(s)		待访问队列长度	
		CSRSE	CSISE	CSRSE	CSISE
C17	96	0.02	0.02	1	42
Fulladder_1	24	0.00	0.00	1	24
Fulladder_2	224	0.53	2.27	1	736
Polybox_5	192	0.02	0.02	1	27
Polybox_9	768	2.01	3.07	1	379
Fulladder_3	448	9.70	33.87	1	3424
Fulladder_4	448	951.83	1633.71	1	44703
Fulladder_8	188	18665.77	60456.85	1	238644

通过表 2 中第 5,6 列可以看出,CSISE-Tree 方法在宽度访问搜索节点时,每次访问一个节点,都需更新待访问节点队列.随着元件数目的增加,待访问节点队列长度成指数级增长,因此占用内存空间成指数级增加.CSRSE 方法在每一次搜索 SE_Tree 时,根据修剪规则都可以直接跳转到下一个需要判断的节点,因此不会产生待访问的节点的序列,只需要存储当前节点,较大程度地节省了内存空间的消耗.

6 结束语

基于模型诊断是人工智能领域内的一个重要问题,求解极小冲突集在基于模型诊断中有着重要应用.CSISE-Tree 方法结合冲突集相关推论、系统输出等信息对翻转的集合枚举树进行了剪枝,提高了求解极小冲

突集的效率.但是在翻转的集合枚举树中,仍存在大量非极小的冲突集未被剪枝.CSRSE 方法不仅对 SE_Tree 中的非冲突集进行剪枝,同时还对部分非极小的冲突集进行剪枝,进一步提高了求解的效率.方法 CSRSE 在遍历 SE_Tree 中的某一个节点时,如果此节点是冲突集,会一直回溯到其祖先节点中长度最小的冲突集节点,并剪除此节点下的所有子孙节点,实现对 SE_Tree 中非极小的冲突集的剪枝;如果此节点不是冲突集,将此节点存储,同时剪除所有是此节点真子集的节点,实现了对 SE_Tree 的非冲突集的剪枝.在求解所有诊断时,CSRSE 方法求解时所占用内存空间只与系统中元件总个数线性相关,而 CSISE-Tree 方法求解时所占用内存空间与元件个数指数级相关.因此与 CSISE-Tree 方法相比,CSRSE 方法的求解效率有明显的提升,还避免了求解时的内存爆炸问题.

参考文献

- [1] Console L, Dressler O. Model-based diagnosis in the real world: lessons learned and challenges remaining [A]. Proceedings of 16th International Joint Conference on Artificial Intelligence [C]. Stockholm, Sweden, 1999. 1393 - 1400.
- [2] De Kleer J. Local methods for localizing faults in electronic circuits [D]. Cambridge, MA, MIT AI Memo 394, 1976.
- [3] Reiter R. A theory of diagnosis from first principles [J]. Artificial Intelligence, 1987, 32(1): 57 - 95.
- [4] Jannach D, Schmitz T, Shechekotkhin K. Parallel model-based diagnosis on multi-core computers [J]. Journal of Artificial Intelligence Research, 2016, 55: 835 - 887.
- [5] Zhao X, Ouyang D. Deriving all minimal hitting sets based on join relation [J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2015, 45(7): 1063 - 1076.
- [6] 刘娟, 欧阳丹彤, 王艺源, 张立明. 结合特征学习的粒子群求解极小碰集方法 [J]. 电子学报, 2015, 43(5): 841 - 845.
- [7] Liu Juan, Ouyang Dantong, et al. Computing minimal hitting sets with particle swarm optimization combined characteristics learning [J]. Acta Electronica Sinica, 2015, 43(5): 841 - 845. (in Chinese)
- [8] Genesereth M R. The use of design descriptions in automated diagnosis [J]. Artificial Intelligence, 1984, 24(1 - 3): 411 - 436.
- [9] Haenni R. A query driven anytime algorithm for argument-active and abduction [A]. Proceedings of 17th National Conference on Artificial Intelligence [C]. Texas, 2000. 337 - 342.
- [9] 栾尚敏, 戴国忠. 利用结构信息的故障诊断方法 [J]. 计算机学报, 2005, 28(5): 801 - 808.
- Luang Shangmin, Dai Guozhong. An approach to diagnos-

- ing system with structure information [J]. Chinese Journal of Computers, 2005, 28(5): 801–808. (in Chinese)
- [10] 代树武, 孙辉先. 基于模型故障诊断中的冲突求解[J]. 控制理论与应用, 2003, 20(4): 630–632.
Dai Shuwu, Sun Huixian. Computing conflict sets for model-based diagnosis [J]. Control Theory & Applications, 2003, 20(4): 630–632. (in Chinese)
- [11] Stern R T, Kalech M, Feldman A, et al. Exploring the duality in conflict-directed model-based diagnosis [A] AAAI Conference on Artificial Intelligence [C]. Canada: Toronto, 2012. 828–834.
- [12] Amgoud L. Postulates for logic-based argumentation systems [J]. International Journal of Approximate Reasoning, 2014, 55(9): 2028–2048.
- [13] Hou A. A theory of measurement in diagnosis from first principles [J]. Artificial Intelligence, 1994, 65(2): 281–328.
- [14] Han B, Lee S J. Deriving minimal conflict sets by CS-Trees with mark set in diagnosis from first principles [J]. IEEE Transactions on System, Man, and Cybernetics, 1999, 29(2): 281–286.
- [15] Smith A, Veneris A, Ali M F, et al. Fault diagnosis and logic debugging using Boolean satisfiability [J]. IEEE Trans on CAD of Integrated Circuits and Systems, 2005, 24(10): 1606–1621.
- [16] 严晓浪, 郑飞君, 葛海通, 杨军. 结合二叉判决图和布尔可满足性的等价性验证算法 [J]. 电子学报, 2004, 32(8): 1233–1235.
Yan Xiaolang, Zheng Feijun, Ge Haitong, Yang jun. Combining binary decision diagrams and Boolean satisfiability for equivalence checking [J]. Acta Electronica Sinica, 2004, 32(8): 1233–1235. (in Chinese)
- [17] Zhao X, Ouyang D. A method of combining SE tree to compute all minimal hitting sets [J]. Progress in Natural Science, 2006, 16(2): 169–174.
- [18] 赵相福, 欧阳丹彤. 使用 SAT 求解器产生所有极小冲突部件集 [J]. 电子学报, 2009, 37(4): 804–810.
Zhao Xiangfu, Ouyang Dantong. Deriving all minimal conflict sets using satisfiability algorithms [J]. Acta Electronica Sinica, 2009, 37(4): 804–810. (in Chinese)
- [19] A. Biere. PicoSAT essentials [J]. Journal on Satisfiability, Boolean Modeling and Computation, 2008, 4: 75–97.

作者简介



欧阳丹彤 女, 1968 年生于吉林长春, 博士, 吉林大学教授、博士生导师, 主要研究方向为基于模型的诊断、自动推理和模型检测。

E-mail: ouyangdantong@163.com



刘博文 男, 1993 年出生吉林延边, 吉林大学硕士研究生, 研究方向为基于模型诊断、SAT 问题。

E-mail: 1591365445@qq.com



张立明 (通信作者) 男, 1980 年出生, 吉林大学博士, 主要研究方向为基于模型诊断、模型检测、SAT 问题。

E-mail: limingzhang@jlu.edu.cn