Front. Comput. Sci., 2023, 17(6): 176407
https://doi.org/10.1007/s11704-022-2261-8

RESEARCH ARTICLE

DiagDO: an efficient model based diagnosis approach with multiple observations

Huisi ZHOU[1,2], Dantong OUYANG[1,2], Xinliang TIAN[1,2], Liming ZHANG (✉)[1,2]

1 Laboratory of Symbolic Computation and Knowledge Engineering, Jilin University, Changchun 130000, China
2 College of Computer Science and Technology, Jilin University, Changchun 130000, China

Abstract Model-based diagnosis (MBD) with multiple observations shows its significance in identifying fault location. The existing approaches for MBD with multiple observations use observations which is inconsistent with the prediction of the system. In this paper, we proposed a novel diagnosis approach, namely, the Diagnosis with Different Observations (DiagDO), to exploit the diagnosis when given a set of pseudo normal observations and a set of abnormal observations. Three ideas are proposed in this paper. First, for each pseudo normal observation, we propagate the value of *system inputs* and gain fanin-free edges to shrink the size of possible faulty components. Second, for each abnormal observation, we utilize filtered nodes to seek surely normal components. Finally, we encode all the surely normal components and parts of dominated components into hard clauses and compute diagnosis using the MaxSAT solver and MCS algorithm. Extensive tests on the ISCAS'85 and ITC'99 benchmarks show that our approach performs better than the state-of-the-art algorithms.

Keywords model based diagnosis, maximum satisfiability, top-level diagnosis, cardinality-minimal diagnosis, subset-minimal diagnosis

1 Introduction

Automated diagnosis is a challenging problem which is concerned with reasoning about the health state of systems, including detecting and isolating faulty components, identifying abnormal behavior of systems, and predicting the behavior of systems under different conditions. Model-based diagnosis (MBD) is a principled approach for detecting and isolating faulty components [1], and it has a wide range of successful practical applications, including qualitative models [2], debugging of web services [3], discrete event systems [4], debugging of relational specifications [5], hybrid systems [6], and spreadsheet debugging [7], among many others.

Diagnosis approaches help in locating faults in failing circuits. Observations with respect to the complex diagnosis system offer a lot of diagnostic system information, which promotes the efficiency of deriving diagnoses. Recent works aim at analyzing multiple observations for the identification of fault locations [8–12]. Without a model of the diagnosed system, some diagnosis approaches explore and learn the system model when the system is in a normal state, which helps in dealing with automated diagnosis [9]. In MBD with multiple observations, the model of the diagnosed system is built firstly. Then multiple observations about the system are used to predict the state of components according to the model. Recent studies of MBD with multiple observations consider some abnormal observations which are inconsistent with the prediction of the system. In this paper, we propose that some pseudo normal observations which are consistent with the prediction are also important when diagnosing a single flipped fault. In addition, we also focus on the information of abnormal observations which are inconsistent with the prediction of the diagnosed system. Note that both pseudo normal observations and abnormal observations are observations when the system in a faulty state.

The first contribution of this paper is analyzing the multiple pseudo normal observations. Each observation consists of a set of *system inputs* and a set of *system outputs*. We provide theoretical analysis for the state of components by propagating the value of *system inputs* w.s.t each pseudo normal observation. Correspondingly, we propose an approach for partitioning components into two parts: surely normal components and possible faulty components. **The second contribution** of this paper is based on multiple abnormal observations. We iteratively find filtered components w.r.t each observation and theoretically analyze the reason why the state of these filtered nodes is surely normal. **The third contribution** of this paper is proposing an approach about how to encode all the components when encoding MBD into the propositional logic formulas. Experimental evaluation building on the well-known ISCAS'85 benchmark and ITC'99 benchmark [13] shows that our DiagDO approach outperforms the state-of-the-art algorithms, namely HSD and IHSD.

The paper is organized as follows. In Section 2, we introduce the related works. In Section 3, we introduce the notations and definitions used throughout this paper. In

Received May 2, 2022; accepted October 31, 2022

E-mail: limingzhang@jlu.edu.cn

Section 4, we state a novel approach to compute a diagnosis. In Section 5, we present the experimental results. In the last section, we present the conclusion of the paper.

2 Related work

Since Reiter proposed the first MBD algorithm, many MBD algorithms have been proposed for diagnosing complex and large-scale systems [14–23]. One classic approach searches diagnosis with an observation that is inconsistent with the prediction according to the model of the system [14–21]. These methods include stochastic diagnosis algorithm [21], the compilation-based algorithm in which OBDDs [24,25] and DNNF [26] are two popular compilation targets, inductive learning methods [27], SAT-based algorithm [18,28,29], breadth-first search based algorithms [16], and conflict-directed diagnosis algorithm [19,20]. Among these algorithms, breadth-first search based algorithms and their improvements make use of the tree structure, in which each node is checked to see if it denotes a minimal diagnosis. These approaches are complete. Obviously, they return all the solutions within enough time. However, they needs a considerable amount of time, which is useless to solve large real-world problems. Today, due to improvement in CPU, some techniques are used to parallelize the construction of the Hitting Set Tree (HS-Tree) and this approach can compute all the diagnoses. Compilation-based approaches successfully exploit the hierarchy of the system and compute candidate solutions in a DNF hierarchy. As a stochastic search algorithm to solve MBD, SAFARI computes a diagnosis in shorter time than many Maximum Satisfiability (MaxSAT) solvers [17,18]. SAFARI randomly removes a component to reduce the cardinality of diagnosis until no component can be removed. Obviously, SAFARI is not guaranteed to return a minimal cardinality diagnosis. Motivated by the continuous performance improvement made to propositional satisfiability (SAT) and MaxSAT solvers, SAT-based approaches have aroused widespread attention. In [18] Feldman compiles the diagnosed circuit into the MaxSAT problem and states that this approach runs longer time than SAFARI. In contrast, SATbD proposed by Metodi considers the immediate dominators of the circuit graph and finds out all diagnoses of minimal cardinality efficiently [30]. In 2015, a novel approach, named Dominator Oriented Encoding (DOE), reduces the structure of the system by filtering some edges and nodes [29].

Other approaches consider the MBD problem with multiple observations [8,10–12]. The DiagCombine (DC) algorithm [10] not only generates abundant redundant diagnoses but also is infeasible in runtime. DC* [12] improves the performance of the DC algorithm in terms of the running time. However, both DC and DC* fail to guarantee to return a minimal diagnosis. Some conflict-directed approaches, such as the implicit Hitting Set Dualization (HSD) algorithm [12] and its improvement, namely, the Improved implicit Hitting Set Dualization (IHSD) algorithm [31], have led to significant development in the field. Given a set of observations that is inconsistent with the system model, they reduce the number of diagnoses but have poor performance with multiple faults. Among these algorithms, they return possible fault locations but are not accurate. Both HSD and IHSD have a drawback concerning computational tractability: HSD computes an exponentially large number of explanations which increases the time cost. Although IHSD uses dominated relationship to reduce the number of explanations, it needs many queries to an NP oracle.

In this paper, we propose a diagnosis approach with multiple observations for a single flipped fault. This novel approach not only reduces the number of diagnoses but also improves diagnostic accuracy.

3 Preliminaries

This paper discusses the MBD problem with multiple observations and computes a cardinality-minimal diagnosis.

3.1 MBD problem

There are three entities in MBD: the diagnosed system description (SD) which is expressed by a set of first-order sentences; the set ($Comps$) of components in the diagnosed system; and an observation (Obs) which is inconsistent with the expected system behavior. The task in the MBD problem is to find an assignment (healthy or faulty) for $Comps$ to explain the inconsistency between the diagnosed system and the observation [1].

Definition 1 (Diagnosis problem)**.** Assuming that the state of each component $c \in Comps$ is healthy, which is denoted by $\neg Ab(c)$, a diagnosis problem exists when system description SD is inconsistent with a given observation Obs, namely:

$$SD \wedge Obs \wedge \{\neg Ab(c) \mid c \in Comps\} \models \perp . \tag{1}$$

Similarly, In the MBD problem with multiple observations, there are three entities: the diagnosed system description (SD) which is expressed by a set of first-order sentences; the set ($Comps$) of components in the diagnosed system; and a set of the observed system behavior ($ObsSet$) in which each observation is inconsistent with the expected system behavior. The task in the MBD problem with multiple observations is to find an assignment (healthy or faulty) for $Comps$ to explain all the observations. Related definitions are shown as follows:

Definition 2 (Diagnosis with multiple observations)**.** Given an MBD problem with multiple observations, $\langle SD, Comps, ObsSet\rangle$, where $ObsSet$ is a set of inconsistent observations (Obs_i represents the ith observation in the $ObsSet$) and SD is the union of all systems *w.r.t.* each observation. A diagnosis with multiple observations is defined as a subset of components $\Delta \subseteq Comps$ when

$$SD \wedge \bigwedge_{i=1}^{m} Obs_i \wedge \{Ab(c) \mid c \in \Delta\} \wedge \{\neg Ab(c) \mid c \in Comps \setminus \Delta\} \not\models \perp . \tag{2}$$

An aggregated diagnosis Δ is subset-minimal iff its any subset is not an aggregated diagnosis. An aggregated diagnosis is cardinality-minimal iff no other subset $\Delta' \subsetneq Comps$ with $|\Delta'| \leqslant |\Delta|$ is an aggregated diagnosis.

This paper uses corresponding notions about system inputs, system outputs and system variables used in [9]. The set of system variables is the union of all the components' inputs and outputs. The union of all the components' inputs that are not the output of any component in the system represents the set of the system inputs, denoted by SysIns. The values of system inputs are set externally by the user. The system outputs, denoted SysOuts, are the components' outputs that are not the input of any component in the system.

3.2 (Partial) MaxSAT problem

Given a set of m Boolean variables $\{x_1, x_2, \dots, x_m\}$, a positive literal is denoted by a variable x, of which polarity is 1, and a negative literal is denoted by negation of a variable $\neg x$, of which polarity is 0. A Conjunctive Normal Form (CNF) formula is a conjunction of clauses (i.e., $F = C_1 \wedge C_2 \wedge \cdots \wedge C_n$), in which clause C_i ($1 \leqslant i \leqslant n$) is a disjunction of literals (i.e., $C_i = l_{i_1} \vee l_{i_2} \vee \cdots \vee l_{i_j}$).

MaxSAT problem is an optimisation version of SAT problem, whose aim is to maximize the number of satisfied clauses. The Partial MaxSAT (PMS) problem is a generalization of MaxSAT in which clauses are divided into hard and soft clauses and it aims at satisfying all hard clauses and as many soft clauses as possible.

3.3 The basic model for MBD

Many recent works model MBD with MaxSAT [11,29]. In MBD modelled by MaxSAT, SD is expressed by a set of hard clauses, *Obs* is expressed by a set of unit hard clauses, and *Comps* is expressed by a set of unit soft clauses. More details about the notions of clauses have been provided in [32].

Figure 1 illustrates c17 circuit from the ISCAS'85 benchmark with six components $\{G_1, G_2, G_3, G_4, G_5, G_6\}$, five *system inputs* $\{i_1, i_2, i_3, i_4, i_5\}$, two *system outputs* $\{o_1, o_2\}$ and inner unobserved *system variables* $\{z_1, z_2, z_3, z_4\}$. Equation (1) lists the propositional formulas of the system description SD for the c17 circuit. There are six conjuncts in Eq. (3) and each conjunct models a single component. For example, the subexpression in the first line means that when component G_1 is normal, the wire modelled by variable z_1 has its value by the wires modelled by variable i_1 and i_2.

$$M = \begin{bmatrix} \neg G_1 \rightarrow (z_1 \Leftrightarrow \neg(i_1 \wedge i_3)) \\ \neg G_2 \rightarrow (z_2 \Leftrightarrow \neg(i_3 \wedge i_4)) \\ \neg G_3 \rightarrow (z_3 \Leftrightarrow \neg(i_2 \wedge z_2)) \\ \neg G_4 \rightarrow (z_1 \Leftrightarrow \neg(i_5 \wedge z_2)) \\ \neg G_5 \rightarrow (o_1 \Leftrightarrow \neg(z_1 \wedge z_3)) \\ \neg G_6 \rightarrow (o_2 \Leftrightarrow \neg(z_3 \wedge z_4)) \end{bmatrix}. \tag{3}$$

With multiple observations, the system description is obtained by replicating the system description for each observation and MaxSAT can be used for solving the union of the clauses. We list the propositional logic formulas for system description with multiple observations in Eq. (4). Each line in Eq. (4) consists of multiple conjuncts. Each conjunct represents the model of SD for each observation. In each conjunct, we use subscript to represent the index of observation. The scale of encoded clauses will be very large with this model when the number of given observations is very large. In some works, all the replied systems share system variables but the assignment of observations is distinct, which is effective for diagnosing with low-cardinality fault. More details have been provided in [11].

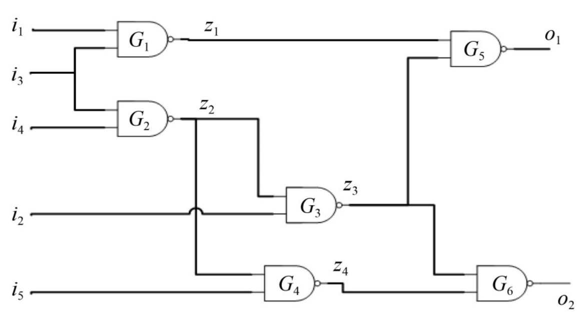

Fig. 1 c17 circuit.

4 Diagnosis for a single flipped fault

In this work, we assume that there is a single flipped fault in the diagnosed system. In this setting, a set of observations that are consistent with the prediction of the system is easily collected. We call these observations pseudo normal observations in the next discussion. Also, we collect a set of observations which are inconsistent with the prediction of the system. We call these observations abnormal observations in the next discussion.

4.1 Diagnosis with pseudo normal observations

Recent literatures research on MBD by exploring the inconsistent formula (2) where some observations which are inconsistent with the prediction of the system. However, when collecting the observations, the observer ignores pseudo normal observations which are consistent with the prediction of the system. When the system is in an abnormal state, there exist some observations which are consistent with the system, especially for the system with a single fault. In this section, we show that pseudo normal observations and diagnoses are closely related. Next, we introduce an approach to get some surely normal components by analyzing pseudo normal observations.

Given an observation, assuming that all the components are healthy, the logic value of each edge can be computed by propagating the value of system inputs. In this paper, all the fanouts of a gate represent all the connections denoting the output of this gate. All the fanins of a gate represent all the connections denoting the input of this gate. According to the gate type, some concepts are proposed in this paper, which is defined as follows:

Definition 3 (Fanin-free edge)**.** A fanin edge E of component C is a Fanin-Free Edge if the value of E doesn't work to the fanout value of C. i.e., Whenever the value assigned to E is, the output value of C remains unchanged.

Example 1 Given a NAND gate C which has p fanin branches, $\{-i_1, i_2, i_3, \dots, i_p\}$, where i_1 is assigned value 0. Clearly, assuming that C is healthy, output(C)=1 whatever the values assigned to the other fanins are. Especially, when more than one fanin is assigned value 0, all the fanins of C are Fanin-Free Edges.

Proposition 1 Given a pseudo normal observation, we propagate the value of observation and delete all the fanin-free edges, if there is a path from the fanout edge of component C to the SysOuts, C is a surely normal component.

Proof Assuming that component G' is inserted a flipped fault and there is a path from the fanout edge of G' to $SysOuts$ after deleting all the fanin-free edges. There are two situations to be discussed. The first situation is that the fanout of G' is a *system out*, then the observation is inconsistent with the prediction of the system. The second situation is the fanout of G' is a fanin of another component G. According to the definition of fanin-free edge, the value of fanout edges of G is flipped. Finally, there will be a system out flipped in the system. This hypothesis is conflicting with the fact that the observation is consistent with the system, so G' is a surely normal component. □

Remarks 1 Given a set of pseudo normal observations $PNObs = \{PNObs_1, PNObs_2, ..., PNObs_n\}$, let S be the set of surely normal components for $PNObs$ and S_i be the set of surely normal components for $PNObs_i$. Then, $S = S_1 \cup \cdots \cup S_n$.

Remarks 2 A diagnosis contains at least one of the possible faulty components and does not contain anyone of surely normal components.

The method used to partition components into surely normal components and possible faulty components with pseudo normal observations is summarized in Algorithm 1. Let NC be the set of surely normal components and FC be the set of possible faulty components. For each observation $PNObs_i$ ($i \in \{1, ..., N\}$), in each iteration, Algorithm 1 obtains a set of all the edges in the system by reading a file of the system description(to see line 3) and computes the values of all the edges (to see line 5). By Definition 3, Algorithm 1 deletes all the fanin-free edges temporarily. By proposition 1, some components are identified as surely normal components. Surely normal components are added into NC (see line 11). Finally, Algorithm 1 returns two sets: NC and FC (see line 17).

$$M = \begin{bmatrix} (\neg G_1 \to (z_1^1 \Leftrightarrow \neg(i_1^1 \wedge i_3^1))) & \wedge & \dots & \wedge & (\neg G_1 \to (z_1^n \Leftrightarrow \neg(i_1^n \wedge i_3^n))) \\ (\neg G_2 \to (z_2^1 \Leftrightarrow \neg(i_3^1 \wedge i_4^1))) & \wedge & \dots & \wedge & (\neg G_2 \to (z_2^n \Leftrightarrow \neg(i_3^n \wedge i_4^n))) \\ (\neg G_3 \to (z_3^1 \Leftrightarrow \neg(z_2^1 \wedge i_2^1))) & \wedge & \dots & \wedge & (\neg G_3 \to (z_3^n \Leftrightarrow \neg(z_2^n \wedge i_2^n))) \\ (\neg G_4 \to (z_4^1 \Leftrightarrow \neg(z_2^1 \wedge i_5^1))) & \wedge & \dots & \wedge & (\neg G_4 \to (z_4^n \Leftrightarrow \neg(z_2^n \wedge i_5^n))) \\ (\neg G_5 \to (o_1^1 \Leftrightarrow \neg(z_1^1 \wedge z_3^1))) & \wedge & \dots & \wedge & (\neg G_5 \to (o_1^n \Leftrightarrow \neg(z_1^n \wedge z_3^n))) \\ (\neg G_6 \to (o_2^1 \Leftrightarrow \neg(z_2^1 \wedge z_4^1))) & \wedge & \dots & \wedge & (\neg G_6 \to (o_2^n \Leftrightarrow \neg(z_2^n \wedge z_4^n))) \end{bmatrix}. \tag{4}$$

In this algorithm, $PNObs = \{PNObs_1, ..., PNObs_N\}$ denotes the set of pseudo normal observations and E denotes the number of all the edges in the system. The maximum and minimum value of the number of $SysIns$ w.r.t. $PNObs_i$ are denoted as Max and Min, respectively. The time complexity of Algorithm 1 is $O(E\ .\ Min\ .\ N)$ in the best case and $O(E\ .\ Max\ .\ N)$ in the worst case.

For the example of Algorithm 1 in Fig. 2, we insert a single flipped fault into $Gate2$ and now we illustrate the process of Algorithm 1 by a pseudo normal observation $PNObs = \{i_1 = 0, i_2 = 1, i_3 = 1, o_1 = 1\}$. In this case, by propagating the value of observation, the output of gate2 is assigned value 0. As a result, the edge $\langle Gate1, Gate3\rangle$ becomes fanin-free edges. Also, the edge $\langle Gate3, Gate4\rangle$ becomes fanin-free edges. According to proposition 1, $\{Gate1, Gate3\}$ are surely normal components.

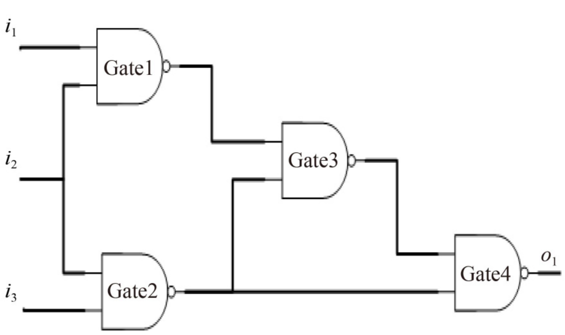

Fig. 2 An example

4.2 Diagnosis with abnormal observations

Algorithm 2 builds on the recent work on the preprocessing of the components and the edges when computing a Top-Level Diagnosis (TLD). The DOE algorithm considers the backbone components and blocked connections of the circuit graph. These are useful in computing a cardinality-minimal diagnosis [4]. In this paper, we propose that filtered nodes with respect

Algorithm 1 Partitioning components into surely normal components and possible faulty components with pseudo normal observations

Input: SD,the system description
Input: $Comps$, the system component
Input: $PNObs_1$,...,$PNObs_N$
Output: A partitioning of the components into two parts: NC and FC.

```
1:  (NC,FC) ← (∅,∅);
2:  for i ∈ {1, 2, ..., N} do
3:      Edges ← Read(SD);
4:      for SysIn ∈ PNObs_i do
5:          Apply DFS on the edges of the system, with source=SysIn;
6:          if edge is a fanin-free Edge then
7:              Delete edge from Edges.
8:          end if
9:          for SysOut ∈ PNObs_i do
10:             if c is in the path from SysIn to SysOut then
11:                 NC ← NC ∪ {c}.
12:             end if
13:         end for
14:     end for
15: end for
16: FC ← Comps \ NC
17: return ( NC, FC )
```

Algorithm 2 Partitioning components into surely normal components and possible faulty components with abnormal observations

Input: SD, the system description
Input: $Comps$, a set of the system components
Input: $ABNObs_1$,...,$ABNObs_M$
Output: A partitioning of the components into two parts: NC and FC.

```
1: (NC,FC) ← (∅,∅);
2: for i ∈ {1, 2, .., M} do
3:    for e ∈ SysIns do
4:       FixedEdges ← e;
5:    end for
6:    for n ∈ dominated nodes do
7:       FixedEdges ← Fanout(n);
8:    end for
9:    Find Filtered Edges and Filtered Nodes;
10:   for c ∈ Filtered Nodes do
11:      NC ← NC ∪ {c};
12:   end for
13: end for
14: FC ← Comps \ NC
15: return ( NC, FC )
```

to each observation are also useful for diagnosis with multiple observations.

Next, we introduce corresponding concepts related to the structure of the system and how to use filtered nodes when diagnosing with multiple observations.

Definition 4 (Dominator)**.** Component G_2 is a dominator of G_1 if all paths from G_1 to the SysOuts include G_2. In other words, G_1 is dominated by G_2. Especially, if all the dominators of G_1, except G_2, is also a dominator of G_2, we say that G_2 is an immediate dominator of G_1.

As an example in Fig. 1, G_1 is dominated by G_5 since the unique path from G_1 to the system output includes G_5.

Definition 5 (Backbone Node(B-Node))**.** Component B is a Backbone Node(B-Node) if it is a dominated component and its fanout has a fixed value for any TLD.

Example 2 In the case of component G_1 in Fig. 1, as mentioned above, it is dominated by G_5. When given an observation, i_1 and $> i_3$ are fixed and the fanout of G_1 is fixed since it is dominated, then, G_1 becomes a B-Node.

Definition 6 (Blocked Edge(B-Edge))**.** A fanin edge E of component C is a Blocked Edge (B-Edge) if the value of E does not change the value of the fanout of C. i.e., whatever the value assigned to E is, the output value of C remains unchanged.

Example 3 In the case of a NAND gate G_2 in Fig. 1, when given an observation $Obs = \{i_4 = 0\}$, the output of G_2 must be 1 for any value of i_3. So edge i_3 is a B-Edge.

Definition 7 (Fixed Edge(F-Edge)])**.** An edge E in the system is a Fixed Edge (F-Edge) if the value of E is fixed for any TLD.

Remarks 3 The edges *w.r.t* SysIns are fixed edges. Besides, the fanout edge of a dominated component for which all the fanin are fixed edges is a fixed edge.

Example 4 In the case of G_1 in Fig. 1, when given an observation $Obs = \{i_1 = 0,\ i_3 = 1\}$, z_1 is a F-Edge since G_1 is dominated by G_5 and i_1 and i_3 are F-Edges.

Definition 8 (Filtered Node)**.** A component is a Filtered Node if all of its fanout edges are Filtered Edge.

Definition 9 (Filtered Edge)**.** An edge is a Filtered Edge if it is a B-Edge or its fanout component is a Filtered Node.

Example 5 As mentioned above, G_1 is a dominated component. Given one observation $Obs = \{i_1 = 1\}$. Then, when $z_1 = 0$ is propagated, the value of z_3 does not work to G_5, that is, the output value of G_5 is a fixed value 1. Thus, $\langle G_3, G_5\rangle$ is a B-Edge. As a result, $\langle G_3, G_5\rangle$ is a filtered edge. Assume that $\langle G_3, G_6\rangle$ is also a filtered edge, component G_3 becomes a filtered node since all of its input edges are filtered edges.

Proposition 2 Filtered nodes w.r.t each observation are surely normal components.

Proof Given a a filter node N w.r.t. an abnormal observation $ABObs_p$, then $\{N\}$ is not a minimal diagnosis since the output of N dose not work to the $SysOut$. Namely, the value of $SysOuts$ cannot be changed by the output of N. Thus, there exist another component N' which can explain the inconsistency between the observations and prediction of the system. Assuming that N is an element in a cardinality-minimal diagnosis, it is conflicting with the fact that there exists a cardinality-minimal diagnosis whose cardinality is 1, so N is a surely normal component. □

With a set of abnormal observations, the method used to partition components into surely normal components and possible faulty components is summarized in Algorithm 2. Let NC be the set of surely normal components and FC be the set of possible faulty components. As proposed in earlier work, filtered nodes are used in the DOE algorithm for simplifying the MBD problem instances when encoding MBD into the MaxSAT problem. Algorithm 2 iteratively computes filtered nodes w.s.t each observation and add them into NC (see line 10−12). In this algorithm, $ABNObs = \{ABNObs_1, ..., ABNObs_M\}$ denotes the set of abnormal observations and E denotes the number of all the edges in the system. The time complexity of Algorithm 2 is $O(E\ .\ M)$.

For the example of Algorithm 2 in Fig. 2, we insert a single flipped fault into $Gate2$. Consider an observation $ABNObs = \{i_1 = 1, i_2 = 1, i_3 = 1, o_1 = 1\}$, the edge $\langle Gate1, Gate3\rangle$ and $\langle Gate2, Gate3\rangle$ are assigned value 0, thus, $\langle Gate1, Gate3\rangle$ and $\langle Gate2, Gate3\rangle$ are filtered edges. As a result, $Gate1$ becomes a filtered node and a surely normal component.

Note that, Proposition 1 and Proposition 2 proposed in this paper are suitable for the cases that a single flipped fault exists in the circuit. Proposition 1 and Proposition 2 are invalid for other cases.

4.3 Computing a diagnosis with encoded propositional logic formulas

Encoding MBD into the propositional logic formulas is a necessary step for diagnosis. In this section, we introduce details about encoding procedure. With SD and $ObsSet$

encoded into hard clauses, surely normal components which are computed by Algorithm 1 and Algorithm 2 are encoded into hard clauses. In addition, when computing a diagnosis in [4], dominated components are encoded into hard clauses. In our approach, we just choose a part of dominated components to be encoded into hard clauses (the number of encoded clauses is decided by a parameter K). There are actually two reasons for it: the first is that fewer soft clauses can reduce the computation time with the MaxSAT solver. The second is that if all the surely normal components and all the dominated components are encoded into hard clauses, there may be no diagnosis. After encoding the MBD into the propositional logic formulas, we also compute diagnosis with implicit hitting set, which is introduced in [12].

5 Experimental evaluation

In this section, we present an experimental evaluation of our proposed approach for MBD with multiple observations. We evaluate our algorithm using ISCSA-85 benchmark Boolean circuit which is used in recent works [6,12,18,23,33,34] and compare the performance of our algorithm with the state-of-the-art algorithms, namely HSD [12], IHSD [31]. We also use a MaxSAT solver, namely RC2 [35], and LBX algorithm [36] which are also used in the HSD algorithm. Our experiments were conducted on Ubuntu 16.04 Linux with Intel Xeon E5-1607 @3.00G Hz, 16GB RAM. For each test case, we collect the execution runtime within 1000 s.

In this paper, we evaluate algorithms when computing cardinality-minimal aggregated diagnoses by using RC2 algorithm [35] and subset-minimal diagnosis by using LBX algorithm [36]. To the best of our knowledge, there are no standard data sets for MBD problem with multiple observations. In this paper, test cases are generated by mimicking a faulty system as in [12]. For each circuit, we generate a single flipped fault and we randomly generate a set of complete instantiations of *system inputs* and *system outputs* according to the fault behavior of the system. We generate two sets of observations, the first for collecting the pseudo normal observations which are consistent with the system, and the second for abnormal observations which are inconsistent with the system. The number of observations of the first set is 500 and the number of observations of the second set is 1000. The run time is measured in CPU seconds.

5.1 ISCAS'85 benchmark

It is an important step to know which components are surely normal components. We generate 50 test cases for each circuit, respectively. We compare the average running time of the three algorithms in the following order: the maximal number of cardinality-minimal diagnoses were set to 1, 5 and 10, respectively. Table 1 provides detailed results to clearly show the difference in the average running time of the three algorithms. The first row of Table 1 is the name of test circuit. For each algorithm, the row "**#1**" shows the average running time when computing a cardinality-minimal diagnosis. The row "**#5**" shows the results for at most 5 cardinality-minimal diagnoses. The row "**#10**" shows the results for at most 10 cardinality-minimal diagnoses. When the time limit is reached, the corresponding column is marked as "N/A" if the algorithm cannot still return all the diagnosis.

For each instance, DiagDO takes less time to obtain the cardinality-minimal diagnoses whatever the number of cardinality-minimal diagnoses is. When the number of cardinality-minimal diagnoses is set to 10, DiagDO proposed in this paper outperforms both HSD and IHSD by one magnitude. The HSD and IHSD algorithm cannot compute more diagnoses within a reasonable time on some large scale circuits, such as c3540, c6288. Except for c6288, the DiagDO algorithm can find the cardinality-minimal diagnoses whose number is set to 5 and 10 within the time limit. In fact, the DiagDO algorithm performs better when computing all the cardinality-minimal diagnoses. For almost of instances, DiagDO can find all the cardinality-minimal diagnoses since it shrinks the number of all the cardinality-minimal diagnoses.

In details, the comparison of HSD, IHSD, and DiagDO in terms of running time is shown in Figs. 3–12 when computing a cardinality-minimal diagnosis. We collect the execution runtime in 1000 seconds with different algorithms, which is presented on a logarithmic scale by the y-axis and the x-axis is the number of instances for each circuit. DiagDO exhibits improved performance compared with the HSD and IHSD for almost all instances except for c6288.

To figure out the reason, we count the average number of surely normal components computed by Algorithm 1 and Algorithm 2, respectively. The results are presented in Table 2. As observed in Table 2, Algorithm 1 is able to identify more surely normal components than Algorithm 2. Algorithm 2 fails to find a surely normal component for the c499 circuit. Both

Table 1 The average running time that all algorithms computes cardinality-minimal diagnoses when building on the ISCAS'85 benchmark

Circuit	HSD			IHSD			DiagDO		
	#1	#5	#10	#1	#5	#10	#1	#5	#10
c432	0.006	0.0179	0.0285	0.0049	0.0211	0.0351	0.0042	0.0042	0.0041
c499	0.018	0.0492	0.0906	0.0159	0.0456	0.149	0.0133	0.0132	0.0133
c880	0.0088	0.0184	0.0179	0.0074	0.0076	0.0076	0.0044	0.0047	0.0046
c1355	0.038	0.1194	0.2402	0.0276	0.0934	0.292	0.0243	0.025	0.0251
c1908	0.0750	0.3013	0.5609	0.0551	0.2678	0.5593	0.0488	0.0492	0.0491
c2670	0.0903	0.2753	0.5094	0.0677	0.2146	0.2862	0.0671	0.0682	0.0684
c3540	0.0902	N/A	N/A	0.0690	N/A	N/A	0.0628	0.0681	0.0686
c5315	0.1984	2.1801	2.8642	0.1385	0.6005	2.1789	0.1299	0.1414	0.1494
c6288	0.6493	N/A	N/A	0.5639	N/A	N/A	0.5636	N/A	N/A
c7552	0.3944	0.6203	0.627	0.2985	0.4762	0.4772	0.2809	0.2904	0.2904

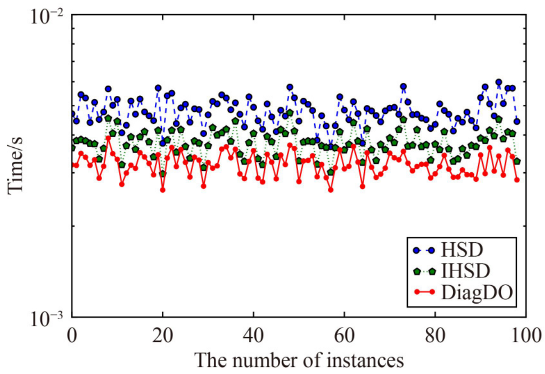

Fig. 3 Runtime for the c432 circuit

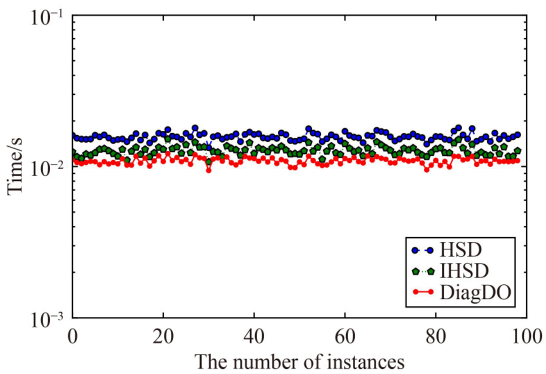

Fig. 4 Runtime for the c499 circuit

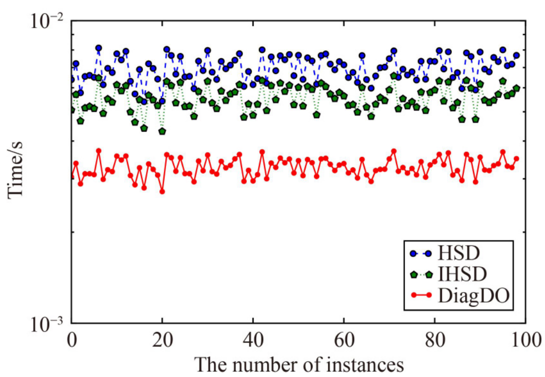

Fig. 5 Runtime for the c880 circuit

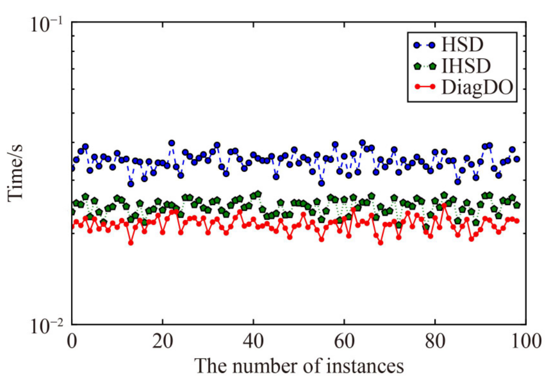

Fig. 6 Runtime for the c1355 circuit

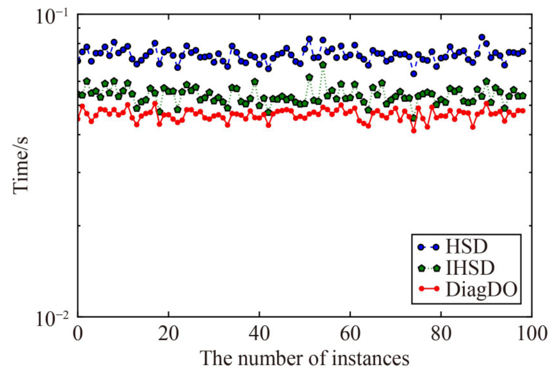

Fig. 7 Runtime for the c1908 circuit

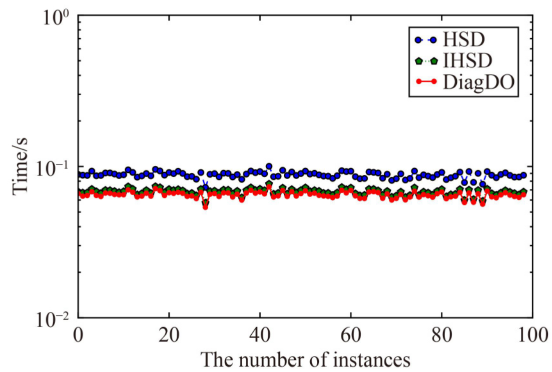

Fig. 8 Runtime for the c2670 circuit

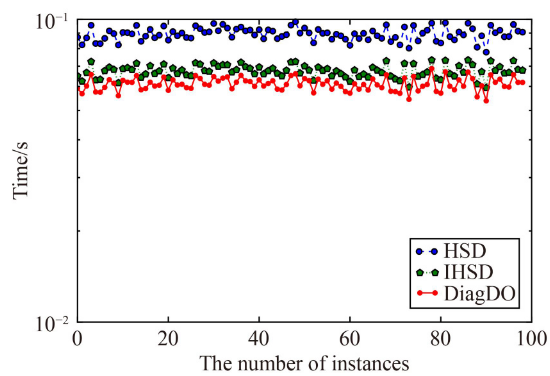

Fig. 9 Runtime for the c3540 circuit

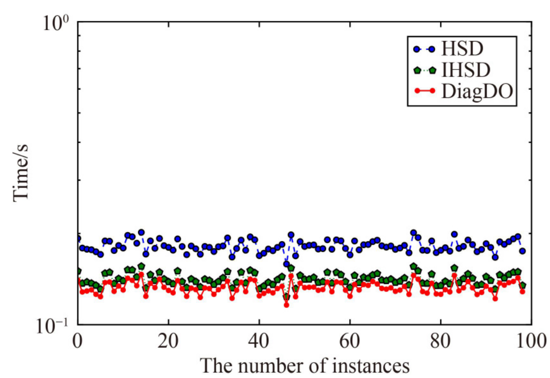

Fig. 10 Runtime for the c5315 circuit

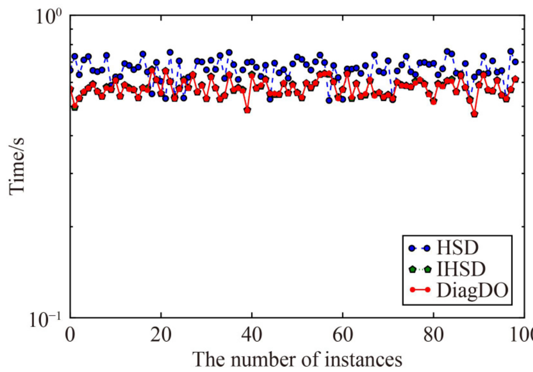

Fig. 11 Runtime for the c6288 circuit

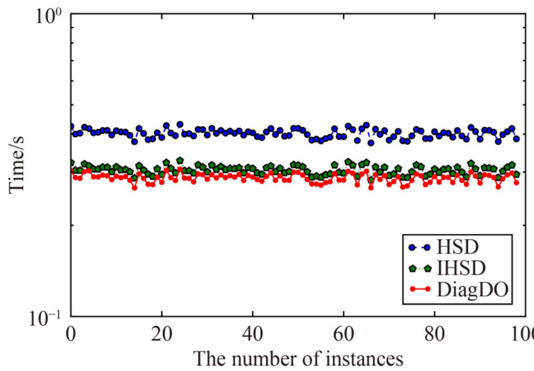

Fig. 12 Runtime for the c7552 circuit

Table 2 The number of surely normal observations with Algorithm 1 and Algorithm 2, respectively

Circuit	with Algorithm 2	with Algorithm 1
c432	59	159
c499	0	184
c880	233	380
c1355	208	510
c1908	101	839
c2670	482	1009
c3540	849	1528
c5315	1359	2201
c6288	0	0
c7552	631	3375

Algorithm 1 and Algorithm 2 can not identify a surely normal component for the c6288 circuit. It seems that this approach may be perturbed by the structure of the system.

When computing subset-minimal diagnoses, Algorithm 2 is useless for finding surely normal components. We generate 30 test cases for each circuit and we computes surely normal components only using Algorithm 1.

We compare the average running time of the three algorithms in the following order: the maximal number of subset-minimal diagnoses were set to 1, 5, 10, and 15, respectively. Table 3 provides detailed results to clearly show the difference in the average running time of the two algorithms. The first row of Table 3 is the name of test circuit. For each algorithm, the row "**#1**" shows the average running time when computing a subset-minimal diagnosis. The row "**#5**" shows the results for at most 5 subset-minimal diagnoses. The row "**#10**" shows the results for at most 10 subset-minimal diagnoses. The row "**#15**" shows the results for at most 10 cardinality-minimal diagnoses. When the time limit is reached, the corresponding column is marked as "N/A" if the algorithm cannot still return all the diagnosis. Note that, in this subsection, DiagDO does not compare with the IHSD algorithm since IHSD aims at computing cardinality-minimal diagnoses.

Note that HSD is time-consuming when computing all the subset-minimal diagnosis. In details, when computing at most 10 subset-minimal diagnosis, the execution runtime with different algorithms are presented in Fig. 13. *Y*-axis presents the runtime in seconds and the x-axis is the number of instances for each circuit. DiagDO proposed in this paper outperforms HSD with performance gains that most often range between 1 and 2 orders of magnitude. DiagDO outperforms HSD in the 95.3% of the instances. Most of the

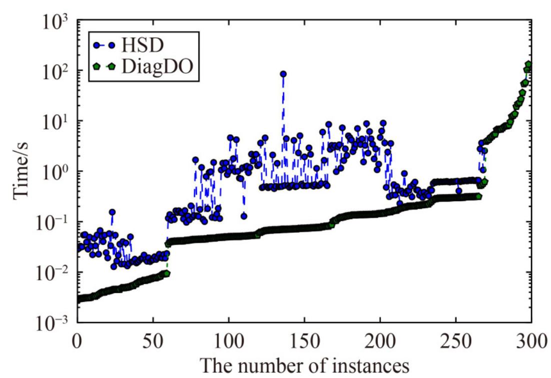

Fig. 13 Runtime of HSD and DiagDO

Table 3 The average running time that all algorithms computes subset-minimal diagnoses when building on the ISCAS'85 benchmark

Circuit	HSD				DiagDO			
	#1	#5	#10	#15	#1	#5	#10	#15
c432	0.0018	0.0095	0.0417	0.0196	0.0013	0.0013	0.0036	0.0013
c499	0.0061	0.0234	0.1317	0.0659	0.0043	0.0043	0.0416	0.0043
c880	0.0033	0.0066	0.0169	0.0068	0.0016	0.0017	0.0065	0.0019
c1355	0.0226	0.0989	0.4023	0.1998	0.0082	0.0083	0.1905	0.0078
c1908	0.302	0.425	1.4314	0.6571	0.0161	0.0169	0.0481	0.1669
c2670	0.0307	0.0917	0.4908	0.2441	0.022	0.2274	0.0689	0.02294
c3540	0.0312	1.1744	5.5455	1.6556	0.0213	0.0216	0.1271	0.22
c5315	0.0627	0.7252	3.8817	1.147	0.0451	0.0453	0.132	0.045
c6288	16.5406	17.6502	20.5641	19.5629	1.4152	2.1187	20.5561	3.6791
c7552	0.1363	0.2082	0.6065	0.2081	0.0968	0.0974	0.2871	0.0973

Table 4 The total running time that all algorithms computes cardinality-minimal diagnoses when building on the ITC'99 benchmark

Circuit	HSD		IHSD		DiagDO	
	#cardinality-minimal	#subset-minimal	#cardinality-minimal	#subset-minimal	#cardinality-minimal	#subset-minimal
b14	1.87	1.79	1.67	1.63	1.55	1.52
b15	21.42	21.86	17.44	17.77	16.10	16.49
b17	151.36	155.30	119.04	118.74	119.46	118.53
b18	290.37	276.70	201.88	156.63	131.82	132.93
b19	271.37	274.71	243.10	262.92	219.52	202.12
b20	63.23	63.04	50.61	50.43	47.18	47.50
b21	64.45	63.47	51.87	51.27	48.58	48.08
b22	95.81	97.25	77.02	78.35	71.76	73.36

instances for which DiagDO has the same performance as the HSD are derived from c6288, because Algorithm 1 fails to find any surely normal components for c6288.

5.2 ITC'99 benchmark

We evaluate algorithms on the ITC'99 benchmark and generate 50 test cases for each circuit, respectively. We also compare the total running time of the three algorithms when computing a cardinality-minimal diagnosis and computing a subset-minimal diagnosis, respectively. Table 4 provides detailed results. When the number of cardinality-minimal diagnoses is more than 1, it is hard for some test cases to get diagnoses within 1000s. Table 4 just lists the results that are computed within 1000s. The first row of Table 4 is the name of the test circuit. For each algorithm, the row "**#cardinality-minimal**" shows the total running time when computing a cardinality-minimal diagnosis. The row "**#subset-minimal**" shows the results for computing a subset-minimal diagnosis.

Compared with HSD and IHSD, DiagDO gets diagnoses within less time when computing a cardinality-minimal diagnosis and computing a subset-minimal diagnosis, respectively. When the number of cardinality-minimal diagnoses is more than 5, all algorithms get diagnoses with more time. We do not list the results for these test circuits in Table 4. In fact, for the test cases that are computed within 1000s, the DiagDO algorithm performs better than the HSD algorithm and IHSD algorithm. This result indicates that the DiagDO algorithm is effective in identifying fault location when a single component is faulty.

6 Conclusions

This paper focuses on the MBD problem by handling a set of pseudo normal observations and a set of abnormal observations at the same time. When there is a single flipped fault, we proposed three ideas for diagnosis. First, when observing a set of normal observations, we focus on reasoning about a set of surely components and finally encode them into hard clauses. Second, when observing a set of abnormal observations, we analyze the structure of the system and encode some components into hard clauses. Finally, some dominated components are encoded into hard clauses, which reduces the solution space for computing a diagnosis. In this paper, we detail the DiagDO approach and provide a conclusive experimental comparison about its performance with respect to the state-of-the-art algorithms. Experimental results show an effective improved performance of DiagDO with respect to HSD and IHSD.

Acknowledgements This work was supported by the National Natural Science Foundation of China (Grant Nos. 62076108, 61972360, and 61872159).

References

1. Reiter R. A theory of diagnosis from first principles. Artificial Intelligence, 1987, 32(1): 57–95
2. Struss P, Price C. Model-based systems in the automotive industry. AI Magazine, 2003, 24(4): 17–34
3. Ardissono L, Console L, Goy A, Petrone G, Picardi C, Segnan M. Cooperative model-based diagnosis of web services. In: Proceeding of the 16th International Workshop on Principles of Diagnosis. 2005, 125–132
4. Pencolé Y, Cordier M O. A formal framework for the decentralised diagnosis of large scale discrete event systems and its application to telecommunication networks. Artificial Intelligence, 2005, 164(1–2): 121–170
5. Torlak E, Chang F S H, Jackson D. Finding minimal unsatisfiable cores of declarative specifications. In: Proceeding of the 15th International Symposium on Formal Methods. 2008, 326–341
6. Narasimhan S, Biswas G. Model-based diagnosis of hybrid systems. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 2007, 37(3): 348–361
7. Jannach D, Schmitz T. Model-based diagnosis of spreadsheet programs: a constraint-based debugging approach. Automated Software Engineering, 2016, 23(1): 105–144
8. Lamraoui S M, Nakajima S. A formula-based approach for automatic fault localization of imperative programs. In: Proceeding of the 16th International Conference on Formal Engineering Methods. 2014, 251–266
9. Stern R, Juba B. Safe partial diagnosis from normal observations. In: Proceedings of the 33rd AAAI Conference on Artificial Intelligence. 2019, 3084–3091
10. Lamraoui S M, Nakajima S. A formula-based approach for automatic fault localization of multi-fault programs. Journal of Information Processing, 2016, 24(1): 88–98
11. Ignatiev A, Morgado A, Weissenbacher G, Marques-Silva J. Model-based diagnosis with multiple observations. In: Proceeding of the 28th International Joint Conference on Artificial Intelligence. 2019, 1108–1115
12. Ignatiev A, Morgado A, Marques-Silva J. Model based diagnosis of multiple observations with implicit hitting sets. 2017, arXiv preprint arXiv: 1707.01972
13. Brglzz F, Fujiwara H. A neutral netlist of 10 combinational benchmark circuits and a target translator in FORTRAN. In: Proceeding of the International Symposium on Circuits and Systems. 1985, 695–698
14. Liu M, Ouyang D, Cai S, Zhang L. Efficient zonal diagnosis with maximum satisfiability. Science China Information Sciences, 2018, 61(11): 112101
15. Siddiqi S, Huang J. Hierarchical diagnosis of multiple faults. In:

Proceedings of the 20th International Joint Conference on Artificial Intelligence. 2007, 581–586
16. Jannach D, Schmitz T, Shchekotykhin K. Parallelized hitting set computation for model-based diagnosis. In: Proceedings of the 29th AAAI Conference on Artificial Intelligence. 2015, 1503–1510
17. Feldman A, Provan G, Van Gemund A. Approximate model-based diagnosis using greedy stochastic search. Journal of Artificial Intelligence Research, 2010, 38: 371–413
18. Feldman A B, Provan G, De Kleer J, Robert S, Van Gemund A J C. Solving model-based diagnosis problems with Max-SAT solvers and vice versa. In: Proceedings of the 21st International Workshop on the Principles of Diagnosis. 2010, 185–192
19. Stern R, Kalech M, Feldman A, Provan G. Exploring the duality in conflict-directed model-based diagnosis. In: Proceeding of the 26th AAAI Conference on Artificial Intelligence. 2012, 828–834
20. Williams B C, Ragno R J. Conflict-directed A^* and its role in model-based embedded systems. Discrete Applied Mathematics, 2007, 155(12): 1562–1595
21. Feldman A, Provan G M, Van Gemund A. Computing minimal diagnoses by greedy stochastic search. In: Proceeding of the 23rd National Conference on Artificial Intelligence. 2008, 911–918
22. Diedrich A, Feldman A, Perdomo-Ortiz A, Abreu R, Niggemann O, de Kleer J. Applying simulated annealing to problems in model-based diagnosis. In: Proceeding of International Workshop on the Principles of Diagnosis. 2016, 4–7
23. Lamperti G, Zanella M, Zhao X. Diagnosis of deep discrete-event systems. Journal of Artificial Intelligence Research, 2020, 69: 1473–1532
24. Lai Y, Liu D, Yin M. New canonical representations by augmenting OBDDs with conjunctive decomposition. Journal Of Artificial Intelligence Research, 2017, 58: 453–521
25. Torasso P, Torta G. Computing minimum-cardinality diagnoses using OBDDs. In: Proceeding of the 26th Annual Conference on Artificial Intelligence. 2003, 224–238
26. Darwiche A. Model-based diagnosis using structured system descriptions. Journal of Artificial Intelligence Research, 1998, 8(1): 165–222
27. Console L, Dupre D T, Torasso P. On the relationship between abduction and deduction. Journal of Logic and Computation, 1991, 1(5): 661–690
28. Metodi A, Stern R, Kalech M, Codish M. Compiling model-based diagnosis to Boolean satisfaction. In: Proceeding of the 26th AAAI Conference on Artificial Intelligence. 2012, 793–799
29. Marques-Silva J, Janota M, Ignatiev A, Morgado A. Efficient model based diagnosis with maximum satisfiability. In: Proceedings of the 24th International Conference on Artificial Intelligence. 2015, 1966–1972
30. Metodi A, Stern R, Kalech M, Codish M. A novel sat-based approach to model based diagnosis. Journal of Artificial Intelligence Research, 2014, 51(1): 377–411
31. Zhou H, Ouyang D, Zhang L, Tian N. Model-based diagnosis with improved implicit hitting set dualization. Applied Intelligence, 2022, 52(2): 2111–2118
32. Lei Z, Cai S. Solving (Weighted) partial MaxSAT by dynamic local search for SAT. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence. 2018, 1346–1352
33. Siddiqi S. Computing minimum-cardinality diagnoses by model relaxation. In: Proceedings of the 22nd International Joint Conference on Artificial Intelligence. 2011, 1087–1092
34. De Kleer J. Minimum cardinality candidate generation. In: Proceedings of the 20th International Workshop on the Principles of Diagnosis. 2009, 397–402
35. Ignatiev A, Morgado A, Marques-Silva J. RC2: an efficient MaxSAT solver. Journal on Satisfiability, Boolean Modeling and Computation, 2019, 11(1): 53–64
36. Mencía C, Previti A, Marques-Silva J. Literal-based MCS extraction. In: Proceeding of the 24th International Conference on Artificial Intelligence. 2015, 1973–1979

Huisi Zhou received her MS degree from Jilin University, China in 2020. She is currently a PhD candidate of Jilin University, China. Her research interests include model-based diagnosis and partial maximum satisfiability problem.

Dantong Ouyang received her PhD degree from Jilin University, China in 1998 and she is currently the professor of Jilin University, China. Her research interests include the model-based diagnosis, satisfiability problem and model checking.

Xinliang Tian received his MS degree from Jilin Univercity, China in 2018. He is currently a PhD candidate of Jilin Univercity, China. His main research interests include model-based diagnosis and combinatorial optimization problem.

Liming Zhang received his MS and PhD degree from Jilin Univercity, China in 2009 and 2012, respectively. He is currently a Senior Engineer of Jilin Univercity, China. His main research interests are satisfiability, integrated circuit diagnosis and testing and maximum satisfiability.