Two Compacted Models for Efficient Model-Based Diagnosis

Huisi Zhou,[1,2] Dantong Ouyang,[1,2*] Xiangfu Zhao,[3] Liming Zhang [1,2*]

[1] College of Computer Science and Technology, Jilin University, Changchun 130010, China
[2] Key Laboratory of Symbolic Computation and Knowledge Engineering, Jilin University, Changchun 130010, China
[3] School of Computer and Control Engineering, Yantai University, Yantai 264005, China
zhouhsi@163.com,ouyd@jlu.edu.cn,xiangfuzhao@gmail.com,limingzhang@jlu.edu.cn

Abstract

Model-based diagnosis (MBD) with multiple observations is complicated and difficult to manage over. In this paper, we propose two new diagnosis models, namely, the Compacted Model with Multiple Observations (CMMO) and the Dominated-based Compacted Model with Multiple Observations (D-CMMO), to solve the problem in which a considerable amount of time is needed when multiple observations are given and more than one fault is injected. Three ideas are presented in this paper. First, we propose to encode MBD with each observation as a subsystem and share as many $system\ variables$ as possible to compress the size of encoded clauses. Second, we utilize the notion of gate dominance in the CMMO approach to compute Top-Level Diagnosis with Compacted Model (CM-TLD) to reduce the solution space. Finally, we explore the performance of our model using three fault models. Experimental results on the ISCAS-85 benchmarks show that CMMO and D-CMMO perform better than the state-of-the-art algorithms.

Introduction

Model-based diagnosis (MBD) is an important approach to analyze complex systems and explain why a system fails, and this problem has been researched for many years in the artificial intelligence (AI) field (Reiter 1987; de Kleer and Williams 1987; Keren, Kalech, and Rokach 2011; Nica et al. 2013). In general, MBD algorithms either use an abductive model in which all behaviors for each component in the system are known and defined (Friedrich, Gottlob, and Nejdl 1990), or a consistent model in which all behaviors for each component in the system are undefined (Reiter 1987; Feldman et al. 2020; Console, Dupré, and Torasso 1991). These algorithms are widely used in various areas including qualitative models (Struss and Price 2004), debugging of web services (Ardissono et al. 2005), discrete event systems (Pencolé and Cordier 2005), debugging of relational specifications (Torlak, Chang, and Jackson 2008), hybrid systems (Narasimhan and Biswas 2007), and spreadsheet debugging (Jannach and Schmitz 2016), among many others.

In most cases, MBD algorithms compute diagnoses using a given system model and one or more observations. In fact, multiple observations with respect to the diagnosis system help promote the efficiency of the diagnosis process as they may offer more diagnostic system information when deriving diagnoses (Ignatiev et al. 2019). Furthermore, obtaining multiple observations is feasible and inexpensive in real-world systems. Many diagnoses that take into account this information have been presented in recent years.

The DiagCombine (DC) algorithm (Lamraoui and Nakajima 2014) aims at computing all covering sets of every set of diagnoses with respect to each observaion. DC* (Lamraoui and Nakajima 2016) improves the runtime of the DC algorithm by reducing the number of redundant diagnoses. However, both DC and DC* fail to guarantee returning a minimal diagnosis. Several algorithms propose a conflict-directed approach, which collects conflicts for all observations and merges them to derive diagnoses. The implicit Hitting Set Dualization (HSD) algorithm (Ignatiev et al. 2019) focuses on computing implicit hitting sets with minimal correction set (MCS) (Marques-Silva et al. 2013; Previti et al. 2018) and maximum satisfiability (MaxSAT) (Narodytska and Bacchus 2014; Cai and Lei 2020) algorithms between diagnoses and conflicts, and outperforms DC and DC* when injecting a single stuck-at fault. Although HSD outperforms DC and DC* in most instances, it also exhibits poor performance when injecting more faults. Its improvement, namely, the Improved implicit Hitting Set Dualization (IHSD) algorithm (Zhou et al. 2021), first uses the notion of gate dominance when deducing a diagnosis with multiple observations. The One-SAT algorithm (Kalech, Stern, and Lazebnik 2021) compiles all the observations into a Satisfiability (SAT) formula (Gu et al. 1999) and derives the health assignment of all components using a SAT solver. This approach will generate a very large number of clauses when given a large number of observations.

This paper explores the MBD problem with multiple observations. **The first contribution** of this paper involves the presentation of a novel compacted diagnosis model with multiple observations. A set of observations consists of a series of $system\ inputs$ and the corresponding $system\ outputs$. Taking the first $system\ inputs$ w.r.t the first observation as a basis, we analyze the difference between the other $system\ inputs$ and the basis. Correspondingly, we propose a selection strategy and the Compacted Model with Multiple Observations (CMMO) encodes constraints for se-

*Corresponding author

lected $system\ variables$. **The second contribution** of this paper is the use of the notion of dominated gates in our new compacted model to compute Top-Level Diagnosis with Compacted Model (CM-TLD). Note that CMMO can also be used for computing either cardinality-minimal aggregated diagnoses or subset-minimal aggregated diagnoses. However, the Dominated-based Compacted Model with Multiple Observations (D-CMMO) is only applicable to computing cardinality-minimal aggregated diagnoses. **The third contribution** of this paper involves the exploration of the compacted model and its improved version using three fault models: (1) Stack-At-one (S-A-1), (2) Stack-At-zero (S-A-0), and (3) flipped. Experimental evaluation building on the well-known ISCAS-85 benchmark (Brglzz and Fujtwara 1985) shows that CMMO outperforms the state-of-the-art algorithms, namely HSD, IHSD, DC, DC*, and One-SAT. Additionally, we conclude that D-CMMO performs better than CMMO in most instances.

The paper is organized as follows. The next section introduces the notations and definitions used throughout this paper. Furthermore, we discuss two system descriptions for the MBD problem with multiple observations in recent work. In Section 3, we state a novel approach to encode MBD into MaxSAT when addressing multiple observations. In Section 4, we present the experimental results. In the last section, we present the conclusion of the paper.

Preliminaries

This paper discusses the MBD problem and computes the diagnosis with multiple observations. More specifically, we focus on a cardinality-minimal diagnosis in a weak-fault model (WFM).

In the MBD problem with a single observation, 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 the observed system behavior (*Obs*) which are a finite set of first-order sentences.

Assuming that the state of each component $c \in \mathit{Comps}$ is healthy, which is denoted by $\neg Ab(c)$, and observations are certain, i.e., there are no errors caused by noisy sensors, a diagnosis problem exists when *SD* is inconsistent with *Obs*, namely:

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

Most diagnosis algorithms focus on how to compute a set of possible consistent-based diagnoses which intends to explain the inconsistency between system description and observations.

Definition 1 (Diagnosis (Reiter 1987))**.** *Given an MBD problem, $\langle \mathit{SD}, \mathit{Comps}, \mathit{Obs}\rangle$, a diagnosis is defined as a set of components $\triangle \subseteq \mathit{Comps}$ when*

$$\begin{aligned} &\mathit{SD} \wedge \mathit{Obs} \wedge \{Ab(c) \mid c \in \triangle\} \\ &\wedge \{\neg Ab(c) \mid c \in \mathit{Comps} \setminus \triangle\} \not\models \perp \end{aligned} \tag{2}$$

A diagnosis $\triangle$ is a subset-minimal diagnosis iff any subset $\triangle' \subset \triangle$ is not a diagnosis, and $\triangle$ is a cardinality-minimal diagnosis iff any one of the other diagnoses $\triangle'$ subjects to $\mid \triangle' \mid \geqslant \mid \triangle \mid$.

When given more than one observation for the MBD problem, one needs to return an assignment for *Comps* to explain all the observations. We use the definition of diagnosis with multiple observations used in previous work (Ignatiev et al. 2019).

Definition 2 (Diagnosis with Multiple Observations (Aggregated Diagnosis) (Ignatiev et al. 2019))**.** *Given an MBD problem with multiple observations, $\langle \mathit{SD}, \mathit{Comps}, \mathit{ObsSet}\rangle$, Assuming that the state of each component $c \in \mathit{Comps}$ is healthy, which is denoted by $\neg Ab(c)$, a diagnosis with multiple observations is defined as a subset of components $\triangle \subseteq \mathit{Comps}$ when*

$$\begin{aligned} \mathit{SD} \wedge \bigwedge_{i=1}^{m} Obs_i \wedge \{Ab(c) \mid c \in \triangle\} \wedge \\ \{\neg Ab(c) \mid c \in \mathit{Comps} \setminus \triangle\} \not\models \perp \end{aligned} \tag{3}$$

Here *ObsSet* is a set of observations (Obs_i represents the i-th observation in the *ObsSet*), *SD* is the union of all systems w.r.t. each observation, and *Comps* is the set of components in the system.

Essentially, this notion is equivalent to the aggregated diagnosis, which is interpreted in (Ignatiev et al. 2019). In this paper, we consider a diagnosis with multiple observations to be an aggregated diagnosis. An aggregated diagnosis $\triangle$ is subset-minimal iff none of its proper subsets is also an aggregated diagnosis. An aggregated diagnosis $\triangle$ is cardinality-minimal iff any one of the other diagnoses $\triangle'$ subjects to $\mid \triangle' \mid \geqslant \mid \triangle \mid$.

To illustrate these notions, consider the c17 circuit from ISCAS-85, as depicted in Figure 1.

Example 1. *Shown in Table 1, we have three observations and corresponding diagnoses with respect to each observation, and we can obtain aggregated diagnoses such as $\{G_6\}$, $\{G_2, G_3\}$, $\{G_4\}$, $\{G_4, G_6\}$. Obviously, $\{G_4, G_6\}$ is neither subset-minimal aggregated diagnoses nor cardinality-minimal aggregated diagnoses. $\{G_4\}$ and $\{G_6\}$ are cardinality-minimal aggregated diagnoses, as well as subset-minimal aggregated diagnoses. $\{G_2, G_3\}$ is a subset-minimal aggregated diagnosis.*

Two System Descriptions for the MBD Problem with Multiple Observations

Many recent works model MBD with MaxSAT, in which *SD* is modelled by a set of hard clauses, *Comps* is modelled by a set of unit soft clauses, and *Obs* is modelled by a set of unit hard clauses (Marques-Silva et al. 2015; Ignatiev et al. 2019). More details about the notions of clauses have been provided in (Cai and Lei 2020).

To the best of our knowledge, there are two ways to encode system description for an MBD problem with multiple observations. The first one, as reported in (Ignatiev et al. 2019), derives a set of system descriptions of each single observation. The diagnosis algorithm is run to obtain a set of diagnoses and a set of explanations. An aggregated diagnosis is a minimal hitting set of the union of all explanations.

Obs_1 :	$\{i_1, -i_2, i_3, i_4, -i_5, o_1, o_2\}$	D_1 :	$\{\{G_3\}, \{G_4\}, \{G_6\}\}$
Obs_2 :	$\{i_1, -i_2, i_3, i_4, i_5, o_1, o_2\}$	D_2 :	$\{\{G_2\}, \{G_4\}, \{G_6\}\}$
Obs_3 :	$\{i_1, -i_2, i_3, -i_4, i_5, o_1, -o_2\}$	D_3 :	$\{\{G_2\}, \{G_3\}, \{G_4\}, \{G_6\}\}$

Table 1: An example for diagnosis with multiple observations

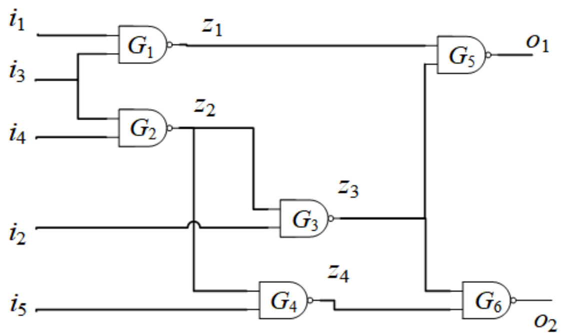

Figure 1: c17 circuit.

The formulas of the system description *SD* with respect to a single observation of the c17 circuit from the ISCAS-85 benchmark (as shown in Figure 1) are listed in Equation 4. There are seven wires and six components in circuit c17, where $\{i_1, i_2, i_3, i_4, i_5\}$ are the $system\ inputs$, $\{o_1, o_2\}$ are the $system\ outputs$, $\{z_1, z_2, z_3, z_4\}$ are internal unobserved $system\ variables$ and $\{G_1, G_2, G_3, G_4, G_5, G_6\}$ are components. Correspondingly, there are six conjuncts in Equation 4 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 modeled by variable z_1 has its value by the wires modeled by variable i_1 and i_3. The whole system description for all observations is obtained by replicating the system description for each observation. Roughly speaking, all the replicated systems share $system\ variables$, but the assignment of observations is distinct. This replication was also done in Bounded Model Checking (BMC), and diagnosis of sequential circuit (Pill and Quaritsch 2013), and distinguishing tests (Alur, Courcoubetis, and Yannakakis 1995).

$$M = \begin{bmatrix} \neg G_1 \rightarrow (z_1 \Leftrightarrow \neg(i_1 \wedge i_3)) \\ \neg G_2 \rightarrow (z_2 \Leftrightarrow \neg(i_3 \wedge i_4)) \\ \neg G_3 \rightarrow (z_3 \Leftrightarrow \neg(i_2 \wedge z_2)) \\ \neg G_4 \rightarrow (z_4 \Leftrightarrow \neg(i_5 \wedge z_2)) \\ \neg G_5 \rightarrow (o_1 \Leftrightarrow \neg(z_1 \wedge z_3)) \\ \neg G_6 \rightarrow (o_2 \Leftrightarrow \neg(z_3 \wedge z_4)) \end{bmatrix} \quad (4)$$

The other method to obtain a system description is to generate one formula that encodes the system's knowledge from all the observations (Kalech, Stern, and Lazebnik 2021). This approach finds a solution that enables all the observations to be consistent with the system description by a MaxSAT solver. In contrast to the indistinguishable replication used in the last approach, this method reassigns an integer number for each $system\ variable$ rather than shares the integer number for $system\ variables$. Furthermore, it shares the integer number assigned to all components. We list propositional logic formulas for system description with multiple observations (as in Equation 5). Each line in Equation 5 consists of multiple conjuncts. However, in this model, the number of encoded clauses will be very large when the number of given observations is very large.

Solving Aggregated Diagnosis

In this paper, we study aggregated diagnosis using a compacted model consisting of a set of subsystems with respect to each observation. The compacted model is a Boolean formula and we can obtain aggregated diagnosis by calling MaxSAT solver once. The CMMO approach and its improvement, namely, the D-CMMO approach, are proposed to compact the model with multiple observations. The process of two approaches is described below.

The CMMO Approach

Let SD_i denote the subsystem with respect to Obs_i. Initially, the first subsystem with respect to Obs_1 is generated by Equation 4. The remaining subsystems are obtained by encoding some components and some wires selectively instead of replicating the entire system. This paper uses corresponding notions about $system\ inputs$, $system\ outputs$ and $system\ variables$ used in (Stern and Juba 2019). 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. The following part introduces how to use these corresponding notions and establish variable mapping when generating the subsystems.

For a given set of observations $\{Obs_1, Obs_2, ..., Obs_m\}$, in this paper, we explore the relation between the first observation and the remaining observations and identify the difference between these observations, which is defined as follows.

Definition 3 (Different Inputs (DI)). *The Different Inputs ($SysIns_x$,$SysIns_y$), denoted $DI_{x,y}$, is a subset of the system inputs, and consists of the inputs with opposite values between $SysIns_x$ and $SysIns_y$.*

Example 2. *Given two system inputs with $SysIn_1$ = $\{-i_1, i_2, i_3, -i_4, i_5\}$, and $SysIn_2$=$\{-i_1, i_2, -i_3, i_4, -i_5\}$, the $DI_{1,2}$ is $\{i_3, i_4, i_5\}$.*

A circuit can be considered as a graph in which edges and nodes are wires and components, respectively.

$$M = \left[\begin{array}{ccccccc} (\neg G_1 \rightarrow (f_1 \Leftrightarrow \neg(a_1 \wedge b_1))) & \bigwedge & \dots & \bigwedge & (\neg G_1 \rightarrow (f_n \Leftrightarrow \neg(a_n \wedge b_n))) & \bigwedge & \dots \\ (\neg G_2 \rightarrow (g_1 \Leftrightarrow \neg(b_1 \wedge c_1))) & \bigwedge & \dots & \bigwedge & (\neg G_2 \rightarrow (g_n \Leftrightarrow \neg(b_n \wedge c_n))) & \bigwedge & \dots \\ (\neg G_3 \rightarrow (h_1 \Leftrightarrow \neg(g_1 \wedge d_1))) & \bigwedge & \dots & \bigwedge & (\neg G_3 \rightarrow (h_n \Leftrightarrow \neg(g_n \wedge d_n))) & \bigwedge & \dots \\ (\neg G_4 \rightarrow (i_1 \Leftrightarrow \neg(g_1 \wedge e_1))) & \bigwedge & \dots & \bigwedge & (\neg G_4 \rightarrow (i_n \Leftrightarrow \neg(g_n \wedge e_n))) & \bigwedge & \dots \\ (\neg G_5 \rightarrow (j_1 \Leftrightarrow \neg(f_1 \wedge h_1))) & \bigwedge & \dots & \bigwedge & (\neg G_5 \rightarrow (j_n \Leftrightarrow \neg(f_n \wedge h_n))) & \bigwedge & \dots \\ (\neg G_6 \rightarrow (k_1 \Leftrightarrow \neg(h_1 \wedge i_1))) & \bigwedge & \dots & \bigwedge & (\neg G_6 \rightarrow (k_n \Leftrightarrow \neg(h_n \wedge i_n))) & \bigwedge & \dots \end{array} \right] \tag{5}$$

$$M = \left[\begin{array}{ccccccc} (\neg G_1 \rightarrow (f_1 \Leftrightarrow \neg(a_1 \wedge b_1))) & \bigwedge & \dots & \bigwedge & (\neg G_1 \rightarrow (f_1 \Leftrightarrow \neg(a_1 \wedge b_1))) & \bigwedge & \dots \\ (\neg G_2 \rightarrow (g_1 \Leftrightarrow \neg(b_1 \wedge c_1))) & \bigwedge & \dots & \bigwedge & (\neg G_2 \rightarrow (g_n \Leftrightarrow \neg(b_1 \wedge c_n))) & \bigwedge & \dots \\ (\neg G_3 \rightarrow (h_1 \Leftrightarrow \neg(g_1 \wedge d_1))) & \bigwedge & \dots & \bigwedge & (\neg G_3 \rightarrow (h_n \Leftrightarrow \neg(g_n \wedge d_1))) & \bigwedge & \dots \\ (\neg G_4 \rightarrow (i_1 \Leftrightarrow \neg(g_1 \wedge e_1))) & \bigwedge & \dots & \bigwedge & (\neg G_4 \rightarrow (i_n \Leftrightarrow \neg(g_n \wedge e_1))) & \bigwedge & \dots \\ (\neg G_5 \rightarrow (j_1 \Leftrightarrow \neg(f_1 \wedge h_1))) & \bigwedge & \dots & \bigwedge & (\neg G_5 \rightarrow (j_n \Leftrightarrow \neg(f_1 \wedge h_n))) & \bigwedge & \dots \\ (\neg G_6 \rightarrow (k_1 \Leftrightarrow \neg(h_1 \wedge i_1))) & \bigwedge & \dots & \bigwedge & (\neg G_6 \rightarrow (k_n \Leftrightarrow \neg(h_n \wedge i_n))) & \bigwedge & \dots \end{array} \right] \tag{6}$$

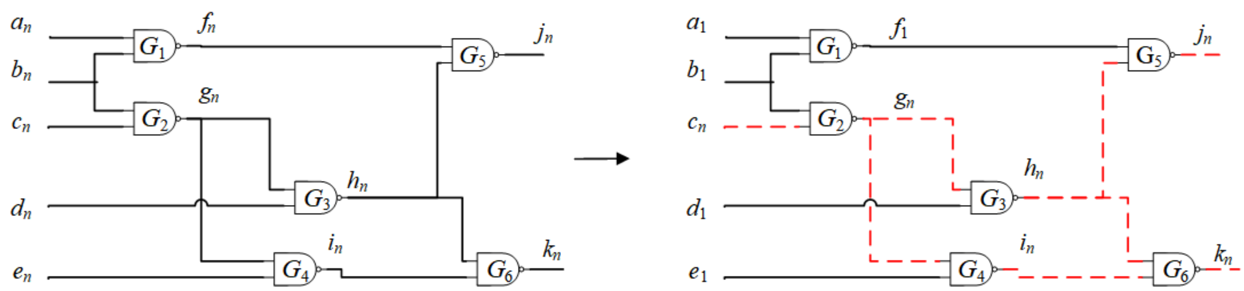

Figure 2: Explanation for c17 circuit with multiple observations.

Definition 4 (Propagation Edges (*P-Edges*)). *Given a system input $i \in SysIns$, P-Edges (i) is a subset of system variables, which are all traversed edges from i to all reachable system outs. Given a subset of system inputs, denoted $SubSysIns$, $SubSysIns \subseteq SysIns$, P-Edges ($SubSysIns$) is a subset of system variables, which are all traversed edges from i to all reachable system outs, where $i \in SubSysIns$.*

Definition 5 (Propagation Nodes (*P-Nodes*)). *Given a system input $i \in SysIns$, P-Nodes (i) is a subset of Comps, which are all traversed nodes from i to all reachable system outs. Given a set of system inputs, denoted $SubSysIns$, $SubSysIns \subseteq SysIns$, P-Nodes ($SubSysIns$) is a subset of system variables, which are all traversed nodes from i to all reachable system outs, where $i \in SubSysIns$.*

Example 3. *Considering the system inputs i_1 in Figure 1, P-Edges (i_1) is $\{i_1, z_1, o_1\}$, and P-Nodes (i_1) is $\{G_1, G_5\}$.*

Example 4. *Considering a subset of system inputs $\{i_1, i_2\}$ in Figure 1, P-Edges ($\{i_1, i_2\}$) is $\{i_1, z_1, o_1, i_2, z_3, o_2\}$, and P-Nodes ($\{i_1, i_2\}$) is $\{G_1, G_3, G_5, G_6\}$.*

The CMMO approach compacts model using definitions 3, 4, and 5 when encoding MBD into MaxSAT. Equation 6 lists the propositional logic formulas for system description using the compacted model. As noted in Equation 6, the variable mapping process, named as *VarMapping*, works as follows: For the first observation, we map the components or wires into a new integer number respectively and build the first subsystem. For the i-th observation, we retain the map for components but reassign an integer number for the selected wires. In contrast to Equation 5, we compute the Different Inputs ($SysIn_1$,$SysIn_i$), denoted DI_{1i}, and reassign an integer number for $system\ variables$ in *P-Edges*(DI_{1i}). Iteratively, the subsystems are built by encoding components in *P-Nodes*(DI_{1i}).

The difference between Equation 5 and Equation 6 is illuminated in Figure 2. The left side of the Figure 2 shows the mapping of integer numbers for $system\ variables$ given the n-th observation using Equation 5 and the right side of the Figure 2 shows the mapping of integer numbers for $system\ variables$ given the n-th observation using Equation 6. In Figure 2, different variables represent different integer numbers. Assuming that different input between $SysIn_1$ and $SysIn_n$ is $\{c_n\}$, in the *VarMapping* process, $system\ variables$ in *P-Edges* (c_n) are reassigned integer numbers. These $system\ variables$ are represented by red dotted lines. When encoding the n-th subsystem, only components in *P-Nodes*(c_n) and its connections need to be encoded, i.e., $\{G_1\}$ and its connections are not contained in the system description of the n-th subsystem.

After encoding system description, CMMO approach encodes all the observations into hard clauses and encodes all the components into soft clauses.

Algorithm 1: MaxSAT-based diagnostic algorithm with multiple observations

Input: *SD*, the system description
Input: *Comps*, the system component
Input: Obs_1,...,Obs_m
Output: Δ.

1: $M_1 \leftarrow$ Encode(*SD*, *Comps*, Obs_1);
2: **for** $i \in \{2, ..., m\}$ **do**
3: Find *P-Nodes*(DI_{1i});
4: Find *P-Edges*(DI_{1i});
5: **for** $c \in$ *P-Nodes*(DI_{1i}) **do**
6: $M_c \leftarrow$ WCNF(c.Fan-in, c.Fan-out, c.GateType).
7: $M_i = M_i \wedge M_c$.
8: **end for**
9: $M = M \wedge M_i \wedge HardCls(DI_{1i})$.
10: **end for**
11: **if** $improvement$ is true **then**
12: **if** $c \in$ *DominatedComps* **then**
13: $M = M \wedge$ *HardCls*(c);
14: **else**
15: $M = M \wedge$ *SoftCls*(c);
16: **end if**
17: **else**
18: $M = M \wedge$ *SoftCls*(c);
19: **end if**
20: $\Delta \leftarrow$ MaxSAT(M)
21: return Δ

Improvements of the CMMO Approach

The second approach we propose for solving an aggregated diagnosis can be viewed as the improved version of CMMO. We call this approach Dominated-based Compacted Model with Multiple Observations (D-CMMO). Instead of computing a cardinality-minimal diagnosis with multiple observations, D-CMMO computes a particular aggregated diagnosis, Top-Level Diagnosis with Compacted Model (CM-TLD).

Definition 6 (Top-Level Diagnosis with Compacted Model (CM-TLD))**.** *A cardinality-minimal diagnosis with multiple observations δ is a Top-Level Diagnosis with Compacted Model if it does not contain any dominated components.*

Early work declares that returning a TLD instead of a diagnosis is indeed a timesaving method. More details about TLD and dominated components are provided in (Metodi et al. 2014). These details are not reiterated in this paper.

The method used to compute an aggregated diagnosis with the CMMO and the D-CMMO is summarized in algorithm 1. Let M be the final propositional logic formulas. We generate the first model M_1 by mapping each component and wire in the system into an integer number and encoding corresponding constants according to Obs_1 (to see line 1). Next, for other observations Obs_i ($i \in \{2, ..., m\}$), we generate other subsystem models (to see line 2-10). Indeed, using less integer number makes model simpler. It is critical to know when to map the $system\ variable$ into a new integer number and when to use the original integer numbers that the first observation used. In each iteration, algorithm 1 computes different inputs between $SysIn_1$ and $SysIn_i$ and propagates the value of DI_{1i}. Correspondingly, we find *P-Edges*(DI_{1i}) and *P-Nodes*(DI_{1i}). Algorithm 1 only encodes the components in *P-Nodes*(DI_{1i}) (see lines 5-8). In detail, two detailed information about encoded component need to be considered. The first is the gate type of encoded component(i.e. NAND, XOR, AND,...). The second is whether component's fan-in and fan-out are contained in *P-Edges*(DI_{1i}). If so, we use a new integer number with *VarMapping* process discussed above. For every $system\ inputs$ w.r.t. DI_{1i}, Algorithm 1 encodes a hard clause(see line 9).

In contrast to CMMO approach, the D-CMMO approach encodes dominated components into hard clauses and encodes remaining components into soft clauses (see lines 12-16). Finally, Algorithm 1 obtains an aggregated diagnosis by calling a MaxSAT solver (see line 20).

Denoting the set of components as *Comps* = $\{c_1, c_2, ..., c_n\}$, the time complexity of generating the final Boolean formula M is $O(m \cdot n)$ in the best case and $O(m^2 \cdot n)$ in the worst case. HSD and IHSD need many queries to an NP oracle, by contrast, Algorithm 1 requires an NP oracle call. In contrast to the One-SAT algorithm, CMMO and D-CMMO generate fewer clauses, which benefits fast solving from aggregated diagnosis.

Experimental Evaluation

To evaluate our algorithm for an aggregated diagnosis, we compare CMMO and its improvement, D-CMMO, with the state-of-the-art algorithms, namely HSD (Ignatiev et al. 2019), IHSD (Zhou et al. 2021), DC (Lamraoui and Nakajima 2016), DC* (Ignatiev et al. 2019) and One-SAT (Kalech, Stern, and Lazebnik 2021). We use RC2 (Ignatiev et al. 2019), a MaxSAT solver, which is also used in the HSD algorithm. In addition, we implement CMMO and D-CMMO in C++ and compile them by g++. Our experiments were conducted on Ubuntu 16.04 Linux with Intel Xeon E5-1607 @3.00G Hz, 16GB RAM.

In this paper, we evaluate algorithms when computing a cardinality-minimal aggregated diagnosis although computing more cardinality-minimal aggregated diagnoses and computing subset-minimal aggregated diagnoses are also supported. Computing more cardinality-minimal aggregated diagnosis is supported by the increased number of iterations using RC2, and computing subset-minimal diagnosis is supported by the use of the LBX algorithm (Mencía, Previti, and Marques-Silva 2015) in Line 20 of Algorithm 1.

We use the systems from the ISCAS-85 benchmark Boolean circuit, which is used in the literature (Ignatiev et al. 2019; Marques-Silva et al. 2015; Siddiqi 2011; Feldman et al. 2010; De Kleer 2009). To the best of our knowledge, there is no standard data sets for MBD problem with multiple observations. In this paper, test cases are generated by mimicking a faulty system as noted in (Ignatiev et al. 2019; Kalech, Stern, and Lazebnik 2021). For each circuit, we evaluate the experiment by establishing three fault models: the stuck-at-zero (S-A-0), the stuck-at-one (S-A-1), and flipped fault. For the first observation, we randomly generate a set of instantiations of $system\ inputs$ and randomly

300 instances	Flipped						
	DC	DC*	HSD	IHSD	One-SAT	CMMO	D-CMMO
# solved	0	0	185	209	230	260	274
# HSD wins	185	185	_	11	81	2	0
# IHSD wins	209	209	197	_	151	40	6
# One-SAT wins	230	230	150	87	_	13	6
# CMMO wins	260	260	256	218	253	_	7
# D-CMMO wins	273	273	272	266	270	269	_

Table 2: Experimental results on the ISCAS-85 benchmark using flipped model fault model.

300 instances	Stack-At-0						
	DC	DC*	HSD	IHSD	One-SAT	CMMO	D-CMMO
# solved	0	0	203	222	231	273	278
# HSD wins	203	203	_	12	97	8	1
# IHSD wins	222	222	209	_	173	47	6
# One-SAT wins	231	231	135	65	_	4	5
# CMMO wins	273	273	264	225	269	_	14
# D-CMMO wins	278	278	276	271	274	269	_

Table 3: Experimental results on the ISCAS-85 benchmark using S-A-0 fault model.

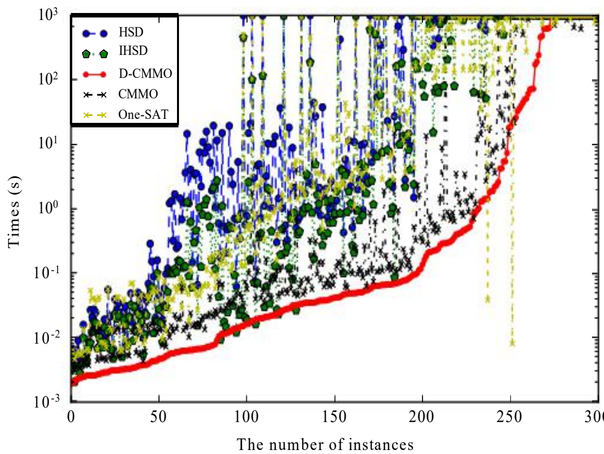

Figure 3: Runtime for the ISCAS-85 benchmark circuit using flipped fault model.

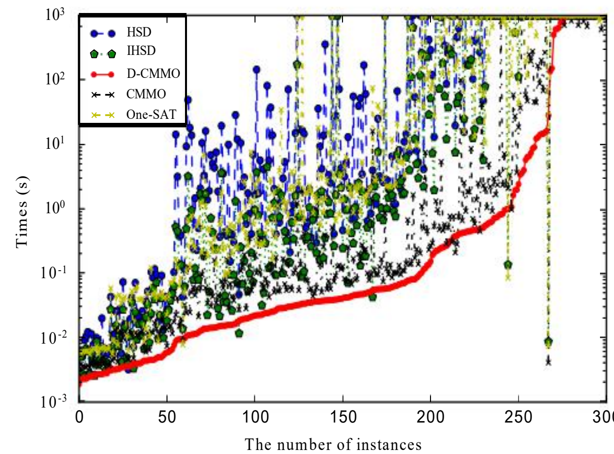

Figure 4: Runtime for the ISCAS-85 benchmark circuit using S-A-0 fault model.

set K components faulty, with K ranging from 20 to 50 (the minimal cardinality of the aggregated diagnosis is often less than K). $System\ outputs$ can be obtained by using SAT to compute a satisfying assignment according to the faulted behavioral model of the circuit. For the remainder of the observations, we generate the $system\ inputs$ by flipping one input of the first observation and generate the $system\ outputs$ by using SAT to compute a satisfying assignment according to the faulted behavioral model of the circuit. In this experiment, the range of the number of observations is 37-234. To illustrate the main points of the paper, this paper generates 300 benchmarks for each fault model.

The results are presented in Tables 2, 3 and 4. For each benchmark, we collect the execution runtime within 1000 s. We list the number of instances which all algorithms solved in the first row. Note that DC and DC* fail to find a diagnosis in almost all instances within 1000 s, so their detailed results are not reported in the tables. We calculate the averaged percent of instances that all algorithms solved with three fault models, which are as follows: 64.9% for HSD, 72.1% for IHSD, 77.6% for One-SAT, 89.6% for CMMO and 92.1% for D-CMMO. As observed, D-CMMO is able to solve more instances than all CMMO, HSD, IHSD, DC, DC* and One-SAT within a given time limit for the S-A-0, S-A-1 and flipped models. CMMO performs better than HSD, IHSD, DC and DC*. With three fault models, D-

300 instances	S-A-1						
	DC	DC*	HSD	IHSD	One-SAT	CMMO	D-CMMO
# solved	0	0	196	218	237	273	277
# HSD wins	196	196	–	15	102	29	1
# IHSD wins	218	218	202	–	173	57	12
# One-SAT wins	237	237	135	66	–	31	9
# CMMO wins	270	270	243	215	247	–	18
# D-CMMO wins	277	277	275	264	274	264	–

Table 4: Experimental results on the ISCAS-85 benchmark using S-A-1 fault model.

Circuit	#inst.	Flipped					S-A-0					S-A-1				
		HSD	IHSD	One-SAT	CMMO	D-CMMO	HSD	IHSD	One-SAT	CMMO	D-CMMO	HSD	IHSD	One-SAT	CMMO	D-CMMO
c1908	30	6	15	17	**30**	**30**	15	19	19	28	**30**	6	11	12	28	**30**
c3540	30	1	2	14	14	**29**	3	6	14	27	**30**	5	10	17	27	**30**
c5315	30	24	29	28	**30**	**30**	25	29	28	**30**	**30**	27	29	**30**	**30**	**30**
c6288	30	0	0	4	**5**	3	0	0	2	**7**	**7**	0	0	**7**	**7**	**7**
c7552	30	4	13	15	**30**	**30**	10	15	16	**30**	**30**	8	15	20	**30**	**30**

Table 5: Comparison on the number of instances solved by all algorithms on large scale circuits.

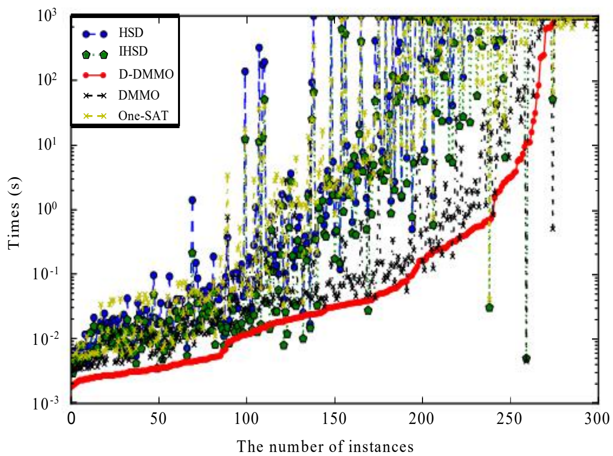

Figure 5: Runtime for the ISCAS-85 benchmark circuit using S-A-1 fault model.

CMMO outperforms HSD, IHSD, One-SAT and CMMO in the 91.4%, 88.1%, 89.0% and 89.1% of the instances respectively on average and CMMO outperforms HSD, IHSD and One-SAT in the 84.8%, 73.11%, and 85.4% of the instances respectively on average.

In detail, Figures 3, 4 and 5 present these experimental results with different fault models. The y-axis shows the differences between the algorithms in runtime which are presented on a logarithmic scale. The x-axis is the number of instances for the ISCAS-85 suite. D-CMMO exhibits improved performance compared with the CMMO, HSD, IHSD, DC, DC* and One-SAT for the solved instances. For almost all circuit system, D-CMMO proposed in this paper outperforms both HSD and IHSD with performance gains that most often range between 1 and 5 orders of magnitude. As noted, the improvement proposed in Section 3 enables the D-CMMO algorithm offer improved performance compared with the CMMO by no more than an order of magnitude in almost all instances.

In addition, for all fault models, the runtime for all the algorithms increases as the number of faults increases. When more faults are injected, almost all the algorithms can not return a cardinality-minimal aggregated diagnosis within 1000 s for large scale circuit. Except DC and DC*, all algorithms can solve instances for small scale circuits within 1000 s. We summarize the experimental results on large scale circuits in Table 5. In bold we present the best results for each circuit. With three groups of experiments, most of the instances for which D-CMMO cannot be solved within a given time frame are derived from c6288.

Conclusions

In this paper, we address the MBD problem when there are multiple observations. We propose three ideas for improving previous approaches. First, we consider diagnosis by building the subsystem for each observation and compute a diagnosis that enables the unions of subsystems to be consistent with all of the observations. In particular, we compact the subsystem using a novel variable mapping process named *VarMapping*. This approach is not used in previous studies. Second, we encode dominated components into hard clauses to compute CM-TLD, which reduces the solution space for computing a diagnosis. Finally, we explore the aggregated diagnosis using three fault models. The experimental results on the ISCAS-85 benchmark show notable gains compared with the state-of-the-art algorithms, namely DC, DC*, HSD, IHSD, and One-SAT.

Diagnosis with multiple observations is more challenging to solve than diagnosis with a single observation. In future work, we will further improve the approach for diagnosis with multiple observations.

Acknowledgments

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

References

Alur, R.; Courcoubetis, C.; and Yannakakis, M. 1995. Distinguishing tests for nondeterministic and probabilistic machines. In *Proceedings of the Twenty-Seventh annual ACM symposium on Theory of computing*, 363–372.

Ardissono, L.; Console, L.; Goy, A.; Petrone, G.; Picardi, C.; Segnan, M.; and Dupré, D. T. 2005. Cooperative model-based diagnosis of web services. In *Proceedings of the Sixteenth International Workshop on Principles of Diagnosis*, 125–130.

Brglzz, F.; and Fujtwara, H. 1985. A neutral netlist of 10 combinational benchmark circuits and a target translator in fortran. In *Proceedings of International Symposium on Circuits and Systems*, 695–698.

Cai, S.; and Lei, Z. 2020. Old techniques in new ways: Clause weighting, unit propagation and hybridization for maximum satisfiability. *Artif. Intell.*, 287: 103354.

Console, L.; Dupré, D. T.; and Torasso, P. 1991. On the relationship between abduction and deduction. *J. Log. Comput.*, 1(5): 661–690.

De Kleer, J. 2009. Mininimum cardinality candidate generation. In *Proceedings of the twentieth International Workshop on the Principles of Diagnosis*, 397–402.

de Kleer, J.; and Williams, B. C. 1987. Diagnosing multiple faults. *Artif. Intell.*, 32(1): 97–130.

Feldman, A.; Pill, I.; Wotawa, F.; Matei, I.; and de Kleer, J. 2020. Efficient model-based diagnosis of sequential circuits. In *Proceedings of the Thirty-Forth AAAI Conference on Artificial Intelligence*, 2814–2821.

Feldman, A.; Provan, G.; De Kleer, J.; Robert, S.; and van Gemund, A. 2010. Solving model-based diagnosis problems with Max-SAT solvers and vice versa. In *Proceedings of the Twenty-First International Workshop on the Principles of Diagnosis*, 1–8.

Friedrich, G.; Gottlob, G.; and Nejdl, W. 1990. Hypothesis classification, abductive diagnosis and therapy. In *International Workshop on Expert Systems in Engineering*, 69–78.

Gu, J.; Purdom, P. W.; Franco, J.; and Wah, B. W. 1999. Algorithms for the satisfiability (sat) problem. *Handbook of Combinatorial Optimization*, 379–572.

Ignatiev, A.; Morgado, A.; Weissenbacher, G.; and Marques-Silva, J. 2019. Model-based diagnosis with multiple observations. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, 1108–1115.

Jannach, D.; and Schmitz, T. 2016. Model-based diagnosis of spreadsheet programs: a constraint-based debugging approach. *Autom. Softw. Eng.*, 23(1): 105–144.

Kalech, M.; Stern, R.; and Lazebnik, E. 2021. Minimal cardinality diagnosis in problems with multiple observations. *Diagnostics*, 11(5): 780.

Keren, B.; Kalech, M.; and Rokach, L. 2011. Model-based diagnosis with multi-label classification. In *Proceedings of the Twenty-Second International Workshop on the Principles of Diagnosis*.

Lamraoui, S.; and Nakajima, S. 2014. A formula-based approach for automatic fault localization of imperative programs. In *Proceedings of the Sixteenth International Conference on Formal Engineering Methods*, 251–266.

Lamraoui, S.; and Nakajima, S. 2016. A formula-based approach for atomatic fault localization of multi-fault programs. *J. Inf. Process.*, 24(1): 88–98.

Marques-Silva, J.; Heras, F.; Janota, M.; Previti, A.; and Belov, A. 2013. On computing minimal correction subsets. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, 615–622.

Marques-Silva, J.; Janota, M.; Ignatiev, A.; and Morgado, A. 2015. Efficient model based diagnosis with Maximum Satisfiability. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, 1966–1972.

Mencía, C.; Previti, A.; and Marques-Silva, J. 2015. Literal-based MCS extraction. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, 1973–1979.

Metodi, A.; Stern, R.; Kalech, M.; and Codish, M. 2014. A novel SAT-based approach to model based diagnosis. *J. Artif. Intell. Res.*, 51: 377–411.

Narasimhan, S.; and Biswas, G. 2007. Model-based diagnosis of hybrid systems. *IEEE Trans. Syst. Man Cybern. Part A*, 37(3): 348–361.

Narodytska, N.; and Bacchus, F. 2014. Maximum satisfiability using core-guided MaxSAT resolution. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2717–2723.

Nica, I.; Pill, I.; Quaritsch, T.; and Wotawa, F. 2013. The route to successA performance comparison of diagnosis algorithms. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*.

Pencolé, Y.; and Cordier, M. 2005. A formal framework for the decentralised diagnosis of large scale discrete event systems and its application to telecommunication networks. *Artif. Intell.*, 164(1-2): 121–170.

Pill, I.; and Quaritsch, T. 2013. Behavioral diagnosis of LTL specifications at operator level. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, 1053–1059.

Previti, A.; Mencía, C.; Järvisalo, M.; and Marques-Silva, J. 2018. Premise set caching for enumerating minimal correction subsets. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, 6633–6640.

Reiter, R. 1987. A theory of diagnosis from first principles. *Artif. Intell.*, 32(1): 57–95.

Siddiqi, S. A. 2011. Computing minimum-cardinality diagnoses by model relaxation. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, 1087–1092.

Stern, R.; and Juba, B. 2019. Safe partial diagnosis from normal observations. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, 3084–3091.

Struss, P.; and Price, C. J. 2004. Model-based systems in the automotive industry. *AI Mag.*, 24(4): 17–34.

Torlak, E.; Chang, F. S.; and Jackson, D. 2008. Finding minimal unsatisfiable cores of declarative specifications. In *Proceedings of the Fifteenth International Symposium on Formal Methods*, 326–341.

Zhou, H.; Ouyang, D.; Zhang, L.; and Tian, N. 2021. Model-based diagnosis with improved implicit hitting set dualization. Forthcoming.