Compiling Model-Based Diagnosis to Boolean Satisfaction*

Amit Metodi
Dept. of Computer Science
Ben-Gurion University of the Negev
Beer-Sheva, Israel 85104
amitmet@cs.bgu.ac.il

Roni Stern and Meir Kalech
Information Systems Engineering
Ben-Gurion University of the Negev
Beer-Sheva, Israel 85104
roni.stern@gmail.com, kalech@bgu.ac.il

Michael Codish
Dept. of Computer Science
Ben-Gurion University of the Negev
Beer-Sheva, Israel 85104
mcodish@cs.bgu.ac.il

Abstract

This paper introduces an encoding of Model Based Diagnosis (MBD) to Boolean Satisfaction (SAT) focusing on minimal cardinality diagnosis. The encoding is based on a combination of sophisticated MBD preprocessing algorithms and SAT compilation techniques which together provide concise CNF formula. Experimental evidence indicates that our approach is superior to all published algorithms for minimal cardinality MBD. In particular, we can determine, for the first time, minimal cardinality diagnoses for the entire standard ISCAS-85 benchmark. Our results open the way to improve the state-of-the-art on a range of similar MBD problems.

Introduction

Automated diagnosis is concerned with reasoning about the health of systems, including the identification of abnormal behavior, isolation of faulty components and prediction of system behavior under normal and abnormal conditions. As systems become large-scale and more complex, their automated diagnosis becomes more challenging. Model Based Diagnosis (MBD) is an artificial intelligence based approach that aims to cope with the diagnosis problem (Reiter 1987; de Kleer and Williams 1987). In MBD, a model of the system is first built. A diagnoser then observes the system to predict its behavior by the model. Discrepancies between the observation and the prediction are used as the input for a diagnosis algorithm which produces a set of possible faults that can explain the observation.

MBD is known to be a hard problem where algorithms have exponential runtime (in the size of the system). Moreover, the number of potential diagnoses that may explain an observation can be huge. Therefore, MBD algorithms typically focus on minimal diagnoses – that do not contain other diagnoses, and on minimal cardinality diagnoses – that are smallest in size. Computing the first minimal diagnosis is in P, but computing the next one is NP-hard (Bylander et al. 1991). Computing the minimal cardinality is NP-hard, even for the first diagnosis (Selman and Levesque 1990).

In case of large-scale systems, MBD is often impractical, especially for high-cardinality faults. For instance, in a system of 1000 components, to find a minimal cardinality diagnosis of size 5, a diagnosis enginemust first verify the absence of a diagnosis consisting of 4 components (there are more than 10^{11} such combinations). To overcome this problem we consider a novel encoding to SAT.

In recent years, Boolean SAT solving techniques have improved dramatically. Today's SAT solvers are considerably faster and able to manage larger instances than yesterday's. Moreover, encoding and modeling techniques are better understood and increasingly innovative. SAT is currently applied to solve a wide variety of hard and practical combinatorial problems, often outperforming dedicated algorithms. The general idea is to encode a (typically, NP) hard problem instance, μ, to a Boolean formula, φ_μ, such that the solutions of μ correspond to the satisfying assignments of φ_μ. Given the encoding, a SAT solver is then applied to solve μ.

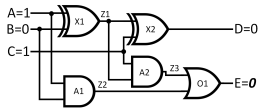
Previous attempts to apply SAT for MBD (see related work, below) indicate that SAT and MAX-SAT solvers perform poorly on the standard benchmarks. This paper introduces a novel SAT encoding based on a combination of sophisticated MBD preprocessing algorithms and SAT compilation techniques. Our approach results in concise CNF formulae. Experimental evidence indicates that are superior to those obtained in previous work.

We focus on minimal cardinality diagnosis evaluating our approach using two standard benchmarks: ISCAS-85 (Brglez et al. 1989) and 74XXX (Hansen, Yalcin, and Hayes 1999). We run two known sets of observations with minimal cardinalities between 1–30, and for the first time succeed to compute a minimal cardinality diagnosis for all observations in the benchmark. We compare our algorithm to HA* (Feldman and van Gemund 2006), CDA* (Williams and Ragno 2007), SAFARI (Feldman, Provan, and van Gemund 2010), HDIAG (Siddiqi and Huang 2007) and DCAS (Siddiqi 2011). Results are unequivocal. Our algorithm outperforms the others, often by orders of magnitude, in terms of runtime. This result is even more significant, as SAFARI is a stochastic algorithm, known as fast, which does not even aim to guarantee minimal cardinality. Our approach, on the other hand, guarantees a minimal cardinality diagnosis and runs faster than SAFARI.

Related work

SAT-based solutions for MBD have already been proposed. Smith *et al.* (Smith et al. 2005) encode a circuit, represent-

*The first and last authors acknowledge the support of the Frankel Center for Computer Science at Ben-Gurion University.



$COMPS = \{X_1, X_2, A_1, A_2, O_1\}$
$OBS = \{A, \neg B, C, \neg D, \neg E\}$

Figure 1: MBD: A full adder.

ing each component through its clauses and add constraints for cardinality. This is the basis for all the other SAT-based encodings, including the one we propose. Bauer (Bauer 2005) introduces a tailored SAT solver specifically designed to return many diagnoses. Stein *et al.* (Stein, Niggemann, and Lettmann 2006) address diagnosis of qualitative models of physical systems with multiple fault modes. Feldman *et al.* (Feldman et al. 2010) propose an encoding to MAX-SAT and demonstrate that off-the-shelf solvers require more consistency checks than the stochastic diagnosis algorithm SAFARI (Feldman, Provan, and van Gemund 2010).

Our approach focuses first on better modeling of MBD, and then on improving its encoding to SAT. Sophisticated preprocessing facilitates the introduction of constraints that enable to simplify the CNF encoding and also boost the search for minimal cardinality diagnosis. It is this "compiler-like" approach that enables us to significantly improve the subsequent runtime of the underlying SAT solver.

Compilation-based methods have also been proposed in the MBD context. Torasso and Torta apply BDDs to compile the model (Torasso and Torta 2006). Darwiche (Darwiche 2001) compiles a system description into Decomposable Negation Normal Form (DNNF) where a minimal cardinality diagnosis can be found in polynomial time. However, in systems with many components, the size of the DNNF becomes a bottleneck (Siddiqi and Huang 2007). Siddiqi (Siddiqi and Huang 2007) suggests to optimize MBD by identifying components that *dominate* others. We adopt this idea and apply it in our SAT-based approach.

Model-Based Diagnosis

Model Based Diagnosis problems arise when the normal behavior of a system is violated due to faulty components as indicated by certain observations. We focus on *weak fault models* (WFM), which ignore the mode of abnormal behavior of components. An MBD problem is specified as a triplet $\langle SD, COMPS, OBS\rangle$ where: SD is a system description, $COMPS$ is a set of components, and OBS is an observation. The system description takes into account that some components might be abnormal (faulty). This is specified by an unary predicate AB on components such that $AB(c)$ is true when component c is faulty. Denoting the correct behavior of c as a propositional formula, φ_c, SD is a conjunction of constraints of the form $(\neg AB(c) \Rightarrow \varphi_c)$. Namely, each component which is not faulty follows its correct behavior.

A diagnosis problem (DP) arises when, under the assumption that none of the components are faulty, there is an inconsistency between the system description and the observation (de Kleer and Williams 1987; Reiter 1987).

Definition 1 [Diagnosis Problem]. *Given an MBD,* $\langle SD, COMPS, OBS\rangle$*, a diagnosis problem arises when* $SD \cup \{\neg AB(c) \mid c \in COMPS\} \cup OBS \vdash \bot$.

For example, a diagnosis problem arises for the MBD of Figure 1 as normal behavior would give output $E = true$.

Once there is an inconsistency, a diagnosis algorithm tries to find a subset $\Delta \subseteq COMPS$ which, if assumed faulty, explains the observation.

Definition 2 [Diagnosis] *Given an MBD,* $\langle SD, COMPS, OBS\rangle$*,* $\Delta \subseteq COMPS$ *is a* diagnosis *if* $SD \cup \{AB(c) \mid c \in \Delta\} \cup \{\neg AB(c) \mid c \in COMPS - \Delta\} \cup OBS \nvdash \bot$ *. We say that* Δ *is a* minimal diagnosis *if no proper subset* $\Delta' \subset \Delta$ *is a diagnosis, and that* Δ *is a* minimal cardinality diagnosis *if for any other diagnosis* $\Delta' \subseteq COMPS$*,* $|\Delta| \leq |\Delta'|$.

For the MBD of Figure 1, Δ_1=$\{X_1, X_2\}$, Δ_2=$\{O_1\}$, Δ_3=$\{A_2\}$ are minimum diagnoses; Δ_2, Δ_3 are minimal cardinality diagnoses as there is no smaller diagnosis.

The Standard Approach to SAT-Based MBD

The standard encoding of MBD to Boolean Satisfiability (e.g. as introduced in (Smith et al. 2005)) associates each component $c \in COMPS$ with a corresponding Boolean "health" variable H_c. Viewing the observation as a propositional statement, an encoding is obtained by specifying

$$\varphi = OBS \ \wedge \ \wedge\{ H_c \Rightarrow \varphi_c \mid c \in COMPS \ \}$$

In a satisfying assignment for φ, the values assigned to the health variables determine a diagnosis Δ. To focus on minimal cardinality diagnosis we seek a satisfying assignment with a minimal number of those variables taking value $false$. This can be achieved using a MAX-SAT solver, as in (Feldman et al. 2010), or, as done in this paper, by introducing a cardinality constraint which constrains the sum of the negated health variables (viewing $true$ as 1 and $false$ as 0). Cardinality constraints are encoded to CNF using standard techniques, e.g. (Eén and Sörensson 2006).

$$\varphi_k = \varphi \wedge (\Sigma\{ \neg H_c \mid c \in COMPS \ \} \leq k)$$

For a constant k, φ_k is satisfied only if at most k health variables take the value $false$. More specifically, we seek a value k such that φ_k is satisfiable and φ_{k-1} is not satisfiable. This involves iterating over calls to the SAT solver.

Our Approach to SAT-Based MBD

Our approach to encoding MBD to SAT builds on the classic one but is constraint-based. First, we model the SD similar to the classic approach but in terms of constraints. Second, we analyze the SD to introduce additional (redundant) constraints that will later boost the search for a minimal cardinality analysis. Third, we introduce constraints to model the given observation OBS and an additional constraint which is inferred by reasoning about SD and OBS and imposes a bound on the cardinality of the diagnosis (the number of unhealthy gates). This additional constraint eases the search for the minimal cardinality diagnosis when later solving the MBD problem. Given all of these constraints, we apply a constraint compiler to simplify and encode them to a corresponding CNF. Finally we apply a SAT solver to seek a suitable satisfying assignment and solve the problem. In the rest of this section we describe these phases in more detail.

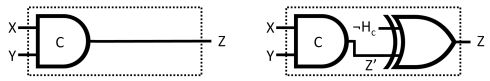


Figure 2: Modeling component c by composition with `xor`

The basic model for SD A well-known improvement for the WFM with a single observation is to observe that when searching for minimal diagnoses the behaviour of each component c can be assumed to be either healthy, in which case c is normal or else unhealthy in which case it produces an opposite output to normal. The basic model for SD consists of constraint

$$\boxed{\wedge \{\ \mathtt{comp(c, H_c)} \mid\ c \in \mathit{COMPS}\ \}} \tag{1}$$

which associates each component c with its health variable H_c. The formal specification of $\mathtt{comp(c, H_c)}$ is illustrated in Figure 2 where component c is modeled by its composition with a new `xor` component that receives as inputs the output from c and the negated health variable $\neg H_c$. If H_c is true then this composition is equivalent to the normal behavior of c, otherwise it is equivalent to component c with a negated output. For example, the system depicted as Figure 1 is modeled by the following constraints:

$$\begin{array}{lcl}
\mathtt{comp(X_1, H_{X_1})} = \mathtt{xor(A, B, Z'_1)} & \wedge & \mathtt{xor(\neg H_{X_1}, Z'_1, Z_1)} \\
\mathtt{comp(X_2, H_{X_2})} = \mathtt{xor(Z_1, C, D')} & \wedge & \mathtt{xor(\neg H_{X_2}, D', D)} \\
\mathtt{comp(A_1, H_{A_1})} = \mathtt{and(A, B, Z'_2)} & \wedge & \mathtt{xor(\neg H_{A_1}, Z'_2, Z_2)} \\
\mathtt{comp(A_2, H_{A_2})} = \mathtt{and(Z_1, C, Z'_3)} & \wedge & \mathtt{xor(\neg H_{A_2}, Z'_3, Z_3)} \\
\mathtt{comp(O_1, H_{O_1})} = \mathtt{or(Z_2, Z_3, E')} & \wedge & \mathtt{xor(\neg H_{O_1}, E', E)}
\end{array}$$

where $\mathtt{xor(A, B, C)}$ ($\mathtt{and(A, B, C)}$,$\mathtt{or(A, B, C)}$) represents the gate with inputs `A`, `B` and output `C`. Modeling $\mathtt{comp(c, H_c)}$ by introducing an `xor` gate for each component later motivates our choice of CryptoMiniSat (Soos 2010), which offers direct support for `xor` clauses, as the underlying SAT solver.

We introduce an additional constraint

$$\boxed{\mathtt{sum\_eq}(\{\ \neg \mathtt{H_c} \mid\ c \in \mathit{COMPS}\ \}, \mathtt{UHs})} \tag{2}$$

to specify that the sum of the negated health variables is the number represented by the Boolean variables `UHs`. Later we will require to satisfy the constraints of the model and also to minimize the number represented by `UHs`.

Introducing (redundant) Cardinality Constraints Reasoning about relations between the components in a system description SD enables to infer (cardinality) constraints about the number of unhealthy components in certain subsystems of SD. These constraints when compiled into the CNF, help boost the search, by the SAT solver, for a minimal cardinality diagnosis. A key point is the assertion in (de Kleer 2008) that the number of outputs in a system is an upper bound on its minimal cardinality diagnosis, for any observation. Another important concept we adopt is that of "gate domination" introduced in (Kirkland and Mercer 1987) and applied also by Siddiqi who further introduces the notion of a "cone". The following wording is taken from (Siddiqi and Huang 2007) in a setting where the system is a logical circuit and the components are its gates.

Definition 3 (Dominator and Cone) *A gate X in the fan-in region of gate G is dominated by G, and conversely G is a dominator of X, if any path from X to an output of the circuit contains G. The cone corresponding to a gate G is the set of gates dominated by G. A maximal cone is one that is either contained in no other cone or contained in exactly one other cone which is the entire circuit.*

Cones are single-output sub-circuits and as such, a minimal cardinality diagnosis will always, independent of the observation, indicate at most one unhealthy component per cone. Hence, given a partition to (maximal) cones, we can introduce cardinality constraints to state that each cone contains at most one unhealthy gate. These constraints, though redundant, improve considerably the search for minimal cardinality diagnosis. This because search can backtrack as soon as it indicates two unhealthy components in a cone.

Motivated by the utility of partitioning a system to cones, we seek a more general partitioning, which enables to apply similar cardinality constraints to larger subsystems of components. To this end we introduce the notion of a "section". We denote by $outputs(c)$ the set of system outputs which occur at the end of a path from a component c.

Definition 4 (Section) *Given a system description SD with components $COMPS$ we define a disjoint partitioning $COMPS = S_1 \cup S_2 \cup \cdots \cup S_n$ such that for every $c_1, c_2 \in COMPS$, c_1 and c_2 are in the same section S_i if and only if $outputs(c_1) = outputs(c_2)$.*

Obtaining a partition to sections following Definition 4 is straightforward and can be computed in polynomial time complexity. Given the partitioning, we introduce the following three constraints which together further improve the encoding and hence the subsequent search for minimal cardinality diagnosis. First we express the sum of the negated health variables per individual section S_i:

$$\boxed{\mathtt{sum\_eq}(\{\ \neg \mathtt{H_c} \mid\ c \in S_i\ \}, \mathtt{UHs_i})} \tag{3}$$

This enables to decompose the total number of unhealthy components in terms of partial sums per section:

$$\boxed{\mathtt{UHs_1} + \cdots + \mathtt{UHs_n} = \mathtt{UHs}} \tag{4}$$

Constraint (4) improves on Constraint (2) which is consequently rendered redundant. Consider that any encoding of the total sum (number of unhealthy components) follows, by divide and conquer, some form of decomposition to partial sums. Constraint (4) guides the encoding to follow a specific decomposition for which the partial sums (number of unhealthy components per section) can be further constrained:

$$\boxed{\textstyle\bigwedge_{i=1}^{n} \mathtt{lessEq(UHs_i, b_i)}} \tag{5}$$

where b_i denotes the smaller of the following two bounds on the number of unhealthy components in section S_i: (a) viewing S_i as a system, following the assertion of de Kleer, the number of outputs from S_i is a bound, and (b) for any $c \in S_i$, $|outputs(c)|$ is a bound. Note that by Definition 4, this value is independent of c.

To illustrate the utility of sections, consider the system given as Figure 3 where cones are depicted in dotted lines and sections in dashed. The section labeled S_1 has 3 outputs, but each component $c \in S_1$ has only 2 corresponding system

outputs ($|outputs(c)| = 2$). Hence, 2 is an upper bound on the number of unhealthy gates in S_1. Note that if reasoning about cones instead of sections, the bound on the number of unhealthy components in S_1 is 3.

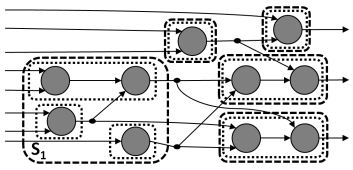


Figure 3: Partitioning a system to cones and sections.

Proposition 1 *Let S be a section and $c \in S$ a component. Then, the number of unhealthy gates (in a minimal cardinality diagnosis) in S is at most $|outputs(c)|$.*

Proof (idea) Let Δ be a minimal cardinality diagnosis with k unhealthy gates and let c be a component in a section S which contains t unhealthy gates. Assume for contradiction that $t > |outputs(c)|$. We construct a diagnostic Δ' with $k' < k$ unhealthy gates as follows: Mark the unhealthy gates from S as healthy and propagate the observed inputs to identify the "flipped" outputs which contradict the observed outputs. For each flipped output $o \in outputs(c)$, toggle the health value for the gate which outputs o to obtain Δ'. The construction marks t unhealthy gates as healthy and at most $|outputs(c)|$ gates as unhealthy. So for Δ' with k' unhealthy gates we have $k' \leq (k - t + |outputs(c)|) < k$. □

The reader might observe another benefit of partitioning to sections: if gate X is dominated by gate G, then $outputs(X)$=$outputs(G)$ implying that the gates of a cone are always in the same section. So, given a partition to sections, the identification of cones may be performed "per section" which is more efficient.

Introducing Constraints to Break Symmetry Symmetry breaking (e.g. (Crawford et al. 1996)) is about adding constraints to select a particular solution in case it is just as good as some other set of solutions. Consider a cone C in SD. Any minimal cardinality diagnosis of SD will indicate at most one unhealthy component in C. Without loss of generality, we may assume that all dominated components in C are healthy. This is correct because if X is unhealthy in some minimal cardinality diagnosis and dominated by G, then G must be healthy. So, there exists another minimal cardinality diagnosis where X is healthy and G is not. Based on this observation we can restrict the search for so-called "top-level" minimal cardinality diagnoses. The following is equivalent to the corresponding definition from (Siddiqi and Huang 2007).

Definition 5 (top-level diagnosis (TLD)) *We say that a minimal cardinality diagnosis is top-level if it does not contain any dominated gates.*

To restrict the search to top-level minimal cardinality diagnoses we add the following constraints where D denotes the set of dominated gates.

$$\boxed{\{\, \mathtt{H_c} = true \mid \; c \in D \;\}} \tag{6}$$

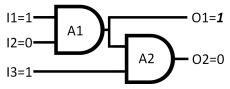


Figure 4: Minimal cardinality diagnosis is size 2 but propagating observed inputs leads to 1 contradiction to the observed outputs.

Introducing constraints to indicate healthy components reduces the number of (unassigned) health variables and hence boosts the search for minimal cardinality diagnosis.

Modeling the Observation and Further Boosting Search Let t and f denote the sets of variables assigned true and false in OBS respectively. Then, to model the observation we add the obvious constraints.

$$\boxed{\{\, \mathtt{X} = true \mid \; \mathtt{X} \in t \;\} \cup \{\, \mathtt{X} = false \mid \; \mathtt{X} \in f \;\}} \tag{7}$$

To improve the search for a minimal cardinality diagnosis one can introduce an upper bound on the minimal cardinality, for example using the assertion from (de Kleer 2008) that the number of outputs in a system is an upper bound on the minimal cardinality. Siddiqi proposes (Siddiqi 2011) to improve this bound for a given observation by propagating the input values through the system, and then taking as an upper bound the number of contradictions between the observed and the propagated outputs. While this idea is intuitively appealing, the circuit depicted as Figure 4 illustrates that it is not always correct. Consider $OBS = \{\; I_1, \neg I_2, I_3, O_1, \neg O_2 \;\}$. Propagating the inputs through the circuit assigns 0 to both outputs and hence counts a single contradiction with the observation (on O_1). However, the only (minimal cardinality) diagnosis for this example has cardinality 2. This example is not contrived: 83 of the 350 observations for system 74181 of the 74XXX benchmark, exhibit a minimal cardinality diagnosis larger than the bound obtained when counting conflicts between propagated and observed outputs.

We propose a fix to obtain a correct upper bound on the size of a minimal cardinality diagnosis.

Procedure 1 *Given a system and an observation, repeatedly propagate the observed inputs through the system as follows: if the inputs of a component c are known then evaluate its normal output v. If the output from c is not in contradiction with the given observation, then mark c as healthy and maintain its output as v. Otherwise, if the observation indicates the output from c as $\neg v$, then mark c as unhealthy and maintain its output as $\neg v$.*

For the circuit in Figure 4, propagating the inputs of gate A_1 gives the output 0 in contradiction to the observation on O_1. Hence, we mark A_1 as unhealthy and propagate the observation $O_1 = 1$ as an input to A_2 together with $I_3 = 1$. This results in an additional contradiction to the observation $O_2 = 0$ and so we mark A_2 as unhealthy too, and report the value 2 as an upper bound for the minimal cardinality. Procedure 1 provides both an upper-bound as well as a diagnosis with the corresponding number of unhealthy com-

ponents. Note that this procedure is correct also when given probes (observed values on the outputs from internal gates).

Proposition 2 *Application of Procedure 1 indicates an upper-bound on the minimal cardinality diagnosis.*

Proof (idea) It follows from the construction that the gates marked unhealthy, call them Δ, are those whose output contradicts the observation. So flipping the outputs from Δ gates gives a behaviour consistent with the observation. Hence, Δ is a diagnosis and so $|\Delta|$ is an upper-bound. □

Let k_{UB} be the bound found by application of Procedure 1 and UHs be the Boolean variables representing the number of the unhealthy components in the system. We introduce a constraint:

$$\boxed{\texttt{lessEq}(\texttt{UHs}, \texttt{k}_{\texttt{UB}})} \quad (8)$$

To appreciate the impact of Procedure 1 we note that, for the ISCAS-85 and 74XXX benchmarks, there are many instances for which Procedure 1 provides a bound identical to the actual minimal cardinality. For such observations, our SAT-based approach still needs to validate that there is no diagnosis of smaller size.

Compiling Constraints to CNF In (Metodi et al. 2011) the authors describe a compiler that encodes finite domain constraints to CNF. Besides facilitating the encoding process, this compiler also applies partial evaluation and other optimizations to simplify the constraints before encoding them to CNF. In particular, it applies "equi-propagation" which is the process of identifying equalities between literals (and constants) implied by a given constraint. If $X{=}L$ is implied by a constraint (where X is a variable and L is a literal or a Boolean constant), then all occurrences of X are replaced by L, reducing the number of variables in the subsequent CNF encoding. For example, consider a constraint $\texttt{comp}(\texttt{c}, \texttt{H}_\texttt{c}) = \texttt{and}(\texttt{X}, \texttt{Y}, \texttt{Z}') \wedge \texttt{xor}(\neg\texttt{H}_\texttt{c}, \texttt{Z}', \texttt{Z})$ modeling the component of Figure 2, in a context where $\texttt{X}{=}1$ is a system input and $\texttt{H}_\texttt{c}{=}1$ is determined by symmetry breaking. Equi-propagation gives the additional equations $\texttt{Y}{=}\texttt{Z}'$, and $\texttt{Z}{=}\texttt{Z}'$ which are applied to all other constraints. Then, $\texttt{comp}(\texttt{c}, \texttt{H}_\texttt{c})$ is removed from the model and not encoded to CNF.

Diagnosis Process and Implementation

We summarize the four phases of the diagnosis process in our approach. Let $\mu = \langle SD, COMPS, OBS \rangle$ be an MBD problem. In the first two phases we construct a constraint model. First, focusing on SD, to introduce constraints which are independent of the observation, and then "per observation" to introduce further constraints. In the third phase we encode the constraint model to a CNF, φ_μ, and finally in the fourth, solve φ_μ using a SAT solver to compute (one or all) minimal cardinality diagnoses.

Modeling the system ("offline"): We first model SD by Constraint (1). We add Constraints (3–5) to bound the number of unhealthy components per section and represent the total number of unhealthy components as an integer variable represented by the Boolean variables UHs. Finally, we add symmetry breaking Constraint (6) which states that, without loss of generality, we may assume that dominated components are healthy. All of the system preprocessing is performed "offline", once per system.

Modeling the observation ("online"): OBS is modeled by introducing Constraint (7) and an additional (cardinality) Constraint (8) is introduced to further bound the number of unhealthy components (by k_{UB}) based on Procedure 1. The simplification of the entire constraint system is performed "online", for each observation.

Encoding: The constraint model is encoded to CNF, φ_μ, applying the optimizing CNF compiler (Metodi et al. 2011).

Solving: To compute a diagnosis, Δ, we seek a satisfying assignment for the encoding, φ_μ, by applying the CryptoMiniSat solver (Soos 2010). Δ is then the set of health variables assigned *false* by this assignment. Denoting $|\Delta| = k$, we again seek a satisfying assignment, but this time for $\varphi \wedge (\texttt{UHs} < k)$. If a satisfying assignment is found, it indicates a smaller diagnosis, Δ'. Otherwise, Δ is of minimal cardinality. This process is invoked repeatedly, each time finding a smaller diagnosis, until for some k' the formula $\varphi \wedge (\texttt{UHs} < k')$ is not satisfiable. Then, the diagnosis found in the previous iteration is of minimal cardinality.

To facilitate the search for a minimal cardinality diagnosis, we apply the SAT solver wrapper, SCryptoMiniSat (Metodi 2012). SCryptoMiniSat takes as input a CNF formula (φ_μ) and the Boolean variables representing a number (UHs). It provides a satisfying assignment which minimizes the given number.

After finding the first minimal cardinality diagnosis with k' faulty components (as described above), we seek to enumerate the set of all top-level minimal cardinality diagnoses. To this end, we apply an additional functionality of SCryptoMiniSat which allows to enumerate (possibly with a specified time-out) all, or a specified number of, satisfying assignments for a given CNF. We apply this option to enumerate all satisfying assignments for $\varphi_\mu \wedge (\texttt{UHs} = k')$.

To compute all minimal cardinality diagnoses, we first observe that any minimal cardinality diagnosis, Δ', which is not top-level can be obtained by expanding another one, Δ, which is top-level and is determined by replacing each dominated component from Δ' by its corresponding dominator. For the converse, given a minimal cardinality TLD, the set of minimal cardinality diagnoses it expands to is given by the following procedure.

Procedure 2 *Let $\Delta = \{G_1, \ldots, G_k\}$ be a minimal cardinality TLD consisting of k dominator gates from corresponding cones $C_1, \ldots, C_k$. First, propagate the inputs through the system, flipping the outputs for unhealthy gates (those in Δ), thus annotating for each cone C_i its inputs and outputs. Then, per cone C_i, determine the set of single components $G \in C_i$ that can replace G_i in Δ such that this replacement is still a diagnosis: $\mathcal{X}_i = \{G \in C_i \mid \Delta \setminus \{G_i\} \cup \{G\}$ is a diagnosis$\}$. Finally, the elements of $\mathcal{X}_1 \times \cdots \times \mathcal{X}_k$ are the minimal cardinality diagnoses that Δ expands to.*

We omit, for lack of space, the formal proof that Procedure 2 provides a representation of all minimal cardinality diagnoses in time $O(|COMPS|^2)$ per TLD.

System					Find a Single Minimal Cardinality Diagnosis (30 sec. timeout)								Find All Minimal Cardinality Diagnoses (1800 sec. timeout)						
					HA*		CDA*		SAFARI		SAT		HDIAG		DCAS		SAT		
Name	comp	in	out	offline Secs.	Succ. rate%	Time Secs.	Succ. rate%	Time Secs.	Succ. rate% (minimal%)	Time Secs.	Succ. rate%	Time Secs.	Succ. rate%	Time Secs.	Succ. rate%	Time Secs.	Succ. rate%	TLD Secs.	All Secs.
74181	65	14	8	0.02	68.3	3.15	46.3	4.51	100.0 (44)	0.00	100.0	0.02				N/A			
74182	19	9	5	0.01	100.0	0.00	100.0	0.01	100.0 (91)	0.00	100.0	0.01				N/A			
74283	36	9	5	0.01	100.0	0.04	100.0	1.45	100.0 (57)	0.00	100.0	0.02				N/A			
c432	160	36	7	0.03	78.1	3.63	38.2	5.15	100.0 (28)	0.03	100.0	0.03	100.0	0.21	100.0	0.31	100.0	0.07	0.09
c499	202	41	32	0.08	24.1	5.45	10.1	1.22	100.0 (7)	0.05	100.0	0.04	100.0	0.12	100.0	0.20	100.0	0.08	0.10
c880	383	60	26	0.06	11.9	3.76	6.3	6.66	100.0 (48)	0.18	100.0	0.05	99.0	0.07	99.0	0.12	100.0	0.08	0.11
c1355	546	41	32	0.24	11.4	3.90	0.0	-	100.0 (5)	0.37	100.0	0.07	99.5	0.16	99.5	0.15	100.0	0.13	0.16
c1908	880	33	25	0.37	6.4	1.75	0.0	-	100.0 (17)	1.08	100.0	0.14	90.5	368.13	76.5	82.25	100.0	0.25	0.30
c2670	1193	233	140	0.29	12.3	4.83	0.0	-	100.0 (14)	2.71	100.0	0.15	90.0	176.17	100.0	3.15	100.0	0.23	0.29
c3540	1669	50	22	0.71	3.7	4.30	0.0	-	100.0 (9)	5.25	100.0	0.27				N/A			
c5315	2307	178	123	1.50	2.7	11.94	0.0	-	100.0 (9)	13.34	100.0	0.42	0.0	-	97.5	52.34	100.0	0.58	0.67
c6288	2416	32	32	1.48	13.6	7.87	0.0	-	53.5 (25)	16.18	100.0	0.56	0.0	-	27.5	305.10	50.0	104.58	105.14
c7552	3512	207	108	1.73	4.2	1.06	0.0	-	0.0	-	99.3	1.07	0.0	-	87.5	260.93	100.0	1.01	1.12
c7552	(with 80 sec timeout)				7.3	20.77	0.0	0.0	99.5 (13)	43.50	100.0	1.49							

Table 1: Systems description **(left)**, and results for Feldman's scenario set **(middle)** and Siddiqi's scenario set **(right)**.

Experimental Results

Table 1 summarizes an evaluation of our SAT-based algorithm on the common benchmarks ISCAS-85 (Brglez et al. 1989), and 74XXX (Hansen, Yalcin, and Hayes 1999). The **left** part presents the systems: names, and numbers of components, inputs and outputs. It also indicates the preprocessing time per system (for the SAT-based approach) which includes all actions performed "once per system": constructing the constraint model, decomposing the system to sections, computing the bounds per section, etc. Experimentation involves two sets of scenarios (observations) with multiple faulty components. The **middle** part of the table presents results for scenarios generated by Feldman *et al.* (Feldman, Provan, and van Gemund 2010). This is the larger set and includes hard scenarios with the highest minimal cardinality. In the **right** part, results for scenarios generated by Siddiqi (Siddiqi 2011) where minimal cardinality is bounded by 8.

Feldman's scenario set: In Table 1 (**middle**) we compare our algorithm with HA* (Feldman and van Gemund 2006), CDA* (Williams and Ragno 2007) and SAFARI (Feldman, Provan, and van Gemund 2010). All experiments are run on the same machine.[1] The table reports on the search for the **first** minimal cardinality diagnosis, for each algorithm, indicating the percentage of observations solved within 30 seconds (succ. rate%) and the average search time, excluding timeouts (time). SAFARI, applies a stochastic approach which does not guarantee minimal cardinality. Feldman *et al.* report (Feldman, Provan, and van Gemund 2010) that even for single and double fault cardinalities, SAFARI does not always find the minimal cardinality. We run SAFARI in a configuration which instead guarantees a minimal subset diagnosis. So, SAFARI is often faster, comparing to HA* and CDA*, at the expense of minimality. For SAFARI, the table indicates in brackets (minimal%) also the percentage of observations, excluding timeouts, where a minimal cardinality diagnosis was found. Our algorithm is presented in the last column (SAT). It clearly outperforms the other three algorithms. There are 11 observations for system `c7552` which we do not solve within the 30 second timeout. These are solved within 80 seconds each, as indicated by the last row in the table. This row reports results when considering an 80 seconds timeout for the 1557 observations for system `c7552`. One may observe that while our SAT based approach now solves all observations, the results for HA* and CDA* are basically the same as with the 30 sec. timeout. SAFARI now solves almost all observations but provides a minimal cardinality diagnosis for only 13%. So, we succeed to compute and verify minimal cardinality diagnosis even for scenarios with a minimal cardinality of 30. To the best of our knowledge, no algorithm before succeeded to compute minimal cardinality diagnosis for such hard scenarios.

Siddiqi's scenario set: In Table 1 (**right**) we compare our algorithm with HDIAG (Siddiqi and Huang 2007) and DCAS (Siddiqi 2011) for which we present results from (Siddiqi 2011) where experiments are reported for a 2.4GHz IntelXeon X3220 with 2Gb RAM (empty rows in the table indicate that these numbers where not reported). Both algorithms verify minimal cardinality diagnosis. The first and third colums report the percentage of scenarios for which HDIAG and DCAS find **all** minimal cardinality diagnoses within 1800 seconds. The second and fourth columns report the average runtimes for the same sets of scenarios. In column 5 we present the success rate (same success criteria) of our algorithm (SAT) running on our machine[1], in column six we report the average runtime to compute the top level diagnoses, and in column seven, the average runtime to compute all minimal cardinality diagnoses. We observe that the difference between finding top level diagnoses and all minimal cardinality diagnoses is small (supporting the fact that it is performed in polynomial time per top-level diagnosis as described above). Our SAT-based algorithm clearly outperforms HDIAG and DCAS. It succeeds to compute all minimal cardinality diagnoses for all observations for all the systems

[1]Intel Core 2 Duo (E8400 3.00GHz CPU, 4GB memory) under Linux (Ubuntu lucid, kernel 2.6.32-24-generic).

except c6288 where it succeeds on 50% of the 40 observations compared to 26.5% for DCAS. Note that because of the higher success rate, the average runtimes of our SAT algorithm involve harder instances not solved by DCAS.

Conclusion

This paper presents a novel encoding of MBD to SAT which enables to determine, for the first time, minimal cardinality diagnoses for the entire standard ISCAS-85 benchmark. The power behind our approach comes from a combination of sophisticated system preprocessing, improved modeling, and the application of SAT compilation techniques. Our Procedure 1 improves on existing techniques to provide an initial bound on minimal cardinality. Our Procedure 2 improves on existing techniques to extend a TLD to the set of all minimal cardinality diagnoses it represents. Experimental evaluation considers the ISCAS-85 and 74XXX benchmarks with large sets of scenarios involving observations with minimal cardinalities of up to 30. We compare our SAT-based algorithm to HA*, CDA*, SAFARI, HDIAG and DCAS. Results are unequivocal. Our algorithm outperforms the others, often by orders of magnitude, in terms of runtime. We succeed to find and verify a minimal cardinality diagnosis for all but 11 of the scenarios in under 30 seconds per scenario, and for the remaining 11 in under 80 seconds. Further details regarding the experimental evaluation as well as a prototype implementation of our SAT-based MBD tool can be found at (Metodi et al. 2012).

References

Bauer, A. 2005. Simplifying diagnosis using LSAT: a propositional approach to reasoning from first principles. In Barták, R., and Milano, M., eds., *Proceedings of the 2005 International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems (CP-AI-OR)*, volume 3524 of *Lecture Notes in Computer Science*, 49–63. Berlin, Heidelberg: Springer-Verlag.

Brglez, F.; Bryan, D.; Kozminski, K.; and of North Carolina, M. C. 1989. *Combinatorial profiles of sequential benchmark circuits*. MCNC technical report.

Bylander, T.; Allemang, D.; Tanner, M. C.; and Josephson, J. R. 1991. The computational complexity of abduction. *Artif. Intell.* 49(1-3):25–60.

Crawford, J. M.; Ginsberg, M. L.; Luks, E. M.; and Roy, A. 1996. Symmetry-breaking predicates for search problems. In *KR*, 148–159.

Darwiche, A. 2001. Decomposable negation normal form. *Journal of the ACM* 48(4):608–647.

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

de Kleer, J. 2008. An improved approach for generating max-fault min-cardinality diagnoses. In *19th International Workshop on Principles of Diagnosis (DX-08)*.

Eén, N., and Sörensson, N. 2006. Translating pseudo-Boolean constraints into sat. *JSAT* 2(1-4):1–26.

Feldman, A., and van Gemund, A. J. C. 2006. A two-step hierarchical algorithm for model-based diagnosis. In *AAAI*.

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 21th International Workshop on Principles of Diagnosis (DX-10)*, 185–192.

Feldman, A.; Provan, G.; and van Gemund, A. 2010. Approximate model-based diagnosis using greedy stochastic search. *J. Artif. Int. Res.* 38:371–413.

Hansen, M. C.; Yalcin, H.; and Hayes, J. P. 1999. Unveiling the iscas-85 benchmarks: A case study in reverse engineering. *IEEE Des. Test* 16:72–80.

Kirkland, T., and Mercer, M. R. 1987. A topological search algorithm for atpg. In *Proceedings of the 24th ACM/IEEE Design Automation Conference*, DAC '87, 502–508. New York, NY, USA: ACM.

Metodi, A.; Codish, M.; Lagoon, V.; and Stuckey, P. J. 2011. Boolean equi-propagation for optimized sat encoding. In Lee, J. H.-M., ed., *CP*, volume 6876 of *Lecture Notes in Computer Science*, 621–636. Springer.

Metodi, A.; Stern, R.; Kalech, M.; and Codish, M. 2012. Compiling model-based diagnosis to Boolean satisfaction: Detailed experimental results and prototype implementation. `http://www.cs.bgu.ac.il/~mcodish/Papers/Pages/aaai-2012.html`.

Metodi, A. 2012. SCryptominisat. `http://amit.metodi.me/research/scrypto`.

Reiter, R. 1987. A theory of diagnosis from first principles. *Artificial Intelligence* 32(1):57–96.

Selman, B., and Levesque, H. J. 1990. Abductive and default reasoning: A computational core. In *AAAI*, 343–348.

Siddiqi, S. A., and Huang, J. 2007. Hierarchical diagnosis of multiple faults. In *IJCAI*, 581–586.

Siddiqi, S. A. 2011. Computing minimum-cardinality diagnoses by model relaxation. In *IJCAI*, 1087–1092.

Smith, E.; Veneris, A.; Member, S.; Ali, M. F.; Member, S.; Member, S.; and Viglas, A. 2005. Fault diagnosis and logic debugging using Boolean satisfiability. *IEEE Trans. on CAD* 24:1606–1621.

Soos, M. 2010. Cryptominisat, v2.5.1. `http://www.msoos.org/cryptominisat2`.

Stein, B.; Niggemann, O.; and Lettmann, T. 2006. Speeding up model-based diagnosis by a heuristic approach to solving sat. In *Proceedings of the 24th IASTED international conference on Artificial intelligence and applications*, 273–278. Anaheim, CA, USA: ACTA Press.

Torasso, P., and Torta, G. 2006. Model-based diagnosis through obdd compilation: A complexity analysis. In *Reasoning, Action and Interaction in AI Theories and Systems*, 287–305.

Williams, B. C., and Ragno, R. J. 2007. Conflict-directed A* and its role in model-based embedded systems. *Discrete Appl. Math.* 155(12):1562–1595.