Efficient Model Based Diagnosis with Maximum Satisfiability*

Joao Marques-Silva
INESC-ID, IST, Portugal
UCD CASL, Ireland
jpms@tecnico.ulisboa.pt

Mikoláš Janota
INESC-ID, IST
Lisbon, Portugal
mikolas.janota@gmail.com

Alexey Ignatiev
INESC-ID, IST
Lisbon, Portugal
aign@sat.inesc-id.pt

Antonio Morgado
INESC-ID, IST
Lisbon, Portugal
ajrmorgado@gmail.com

Abstract

Model-Based Diagnosis (MBD) finds a growing number of uses in different settings, which include software fault localization, debugging of spreadsheets, web services, and hardware designs, but also the analysis of biological systems, among many others. Motivated by these different uses, there have been significant improvements made to MBD algorithms in recent years. Nevertheless, the analysis of larger and more complex systems motivates further improvements to existing approaches. This paper proposes a novel encoding of MBD into maximum satisfiability (MaxSAT). The new encoding builds on recent work on using Propositional Satisfiability (SAT) for MBD, but identifies a number of key optimizations that are very effective in practice. The paper also proposes a new set of challenging MBD instances, which can be used for evaluating new MBD approaches. Experimental results obtained on existing and on the new MBD problem instances, show conclusive performance gains over the current state of the art.

1 Introduction

Model based diagnosis (MBD) is a well-known research topic in AI, with seminal work in the 80s [Reiter, 1987; de Kleer and Williams, 1987], and with a large body of work over the years. A number of alternative approaches have been proposed for MBD, including conflict-based [de Kleer *et al.*, 1992; de Kleer, 2009], constraint-based [Williams and Ragno, 2007; Feldman *et al.*, 2010; Metodi *et al.*, 2012; Nica and Wotawa, 2012; Metodi *et al.*, 2014], compilation-based [Darwiche, 2001; Siddiqi and Huang, 2007], and the use of duality in conflict-based approaches [Stern *et al.*, 2012]. Given a model of a system and an observation of the system's inputs and outputs, not consistent with the expected behavior, the goal of MBD is to identify a subset of the components (possibly minimal or of minimal cardinality), such that discarding those components makes the system consistent with the observation.

In recent years, a number of constraint-based approaches for MBD proposed the use SAT or MaxSAT [Smith *et al.*, 2005; Bauer, 2005; Safarpour *et al.*, 2007; Feldman *et al.*, 2010; Metodi *et al.*, 2012; Nica *et al.*, 2013; Metodi *et al.*, 2014], where minimal cardinality diagnoses are computed either with iterative SAT solving or with dedicated MaxSAT solvers. In addition, the importance of constraint-based MBD is demonstrated by applications in a number of different settings, which include software fault localization in C code with MaxSAT [Jose and Majumdar, 2011], spreadsheet debugging with CP [Hofer *et al.*, 2013], design debugging, both with SAT [Smith *et al.*, 2005] and MaxSAT [Safarpour *et al.*, 2007], among many others.

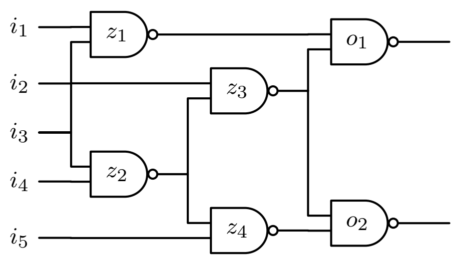
Recent work on compiling MBD to SAT showed clear performance gains over earlier approaches [Metodi *et al.*, 2012; 2014]. This paper builds on this recent work and develops a number of improvements to the compilation of MBD into SAT, which produce further categorical performance gains over what is arguably the currently best performing MBD approach [Metodi *et al.*, 2014]. Since the paper exploits *dominators* (in line with recent work [Siddiqi and Huang, 2007; Metodi *et al.*, 2014]), the new MBD to SAT compilation approach is referred to as *Dominator-Oriented Encoding* (DOE).

The paper is organized as follows. Section 2 introduces the definitions and notation used throughout. Section 3 develops the DOE approach for compiling MBD into SAT/MaxSAT. Section 4 compares the DOE approach with SCryptoDiagnoser [Metodi *et al.*, 2014], a state-of-the-art MBD approach. Section 5 concludes the paper.

2 Preliminaries

Well-known definitions from Propositional Satisfiability (SAT) and Maximum Satisfiability (MaxSAT) [Biere *et al.*, 2009] are assumed. Moreover, basic knowledge of recent MaxSAT algorithms, i.e. MaxSAT algorithms based on the iterative identification of unsatisfiable subformulas (or cores), is also assumed [Ansótegui *et al.*, 2013; Morgado *et al.*, 2013]. In addition, standard definitions of circuits are assumed [Biere *et al.*, 2009], including controlling (resp. non-controlling) input values of simple gates, concretely 0 (resp. 1) for AND/NAND and 1 (resp. 0) for OR/NOR).

*This work is partially supported by SFI grant BEACON (09/IN.1/I2618), by FCT grant POLARIS (PTDC/EIA-CCO/123051/2010), and by national funds through FCT with reference UID/CEC/50021/2013.



(a) C17 circuit

Obs	$\langle i_1, i_2, i_3, i_4, i_5 \rangle$	$\langle o_1, o_2 \rangle$
19	$\langle 1, 0, 0, 0, 1 \rangle$	$\langle 1, 1 \rangle$
20	$\langle 0, 1, 1, 1, 1 \rangle$	$\langle 1, 1 \rangle$
51	$\langle 1, 0, 1, 1, 1 \rangle$	$\langle 0, 1 \rangle$
53	$\langle 1, 1, 1, 0, 1 \rangle$	$\langle 0, 0 \rangle$
63	$\langle 1, 0, 1, 0, 1 \rangle$	$\langle 0, 0 \rangle$

(b) Selected observations

$$
\begin{aligned}
\mathsf{Comps} &\triangleq \{z_1, z_2, z_3, z_4, o_1, o_2\} \\
\mathsf{SD} &\triangleq \textstyle\bigwedge_{c \in \mathsf{Comps}} (\mathsf{Ab}(c) \vee \mathcal{F}_c) \\
\mathcal{F}_{z_1} &\triangleq \mathsf{CNF}(z_1 \leftrightarrow \neg(i_1 \wedge i_3)) \\
\mathcal{F}_{z_2} &\triangleq \mathsf{CNF}(z_2 \leftrightarrow \neg(i_3 \wedge i_4)) \\
\mathcal{F}_{z_3} &\triangleq \mathsf{CNF}(z_3 \leftrightarrow \neg(i_2 \wedge z_2)) \\
\mathcal{F}_{z_4} &\triangleq \mathsf{CNF}(z_4 \leftrightarrow \neg(z_2 \wedge i_5)) \\
\mathcal{F}_{o_1} &\triangleq \mathsf{CNF}(o_1 \leftrightarrow \neg(z_1 \wedge z_3)) \\
\mathcal{F}_{o_2} &\triangleq \mathsf{CNF}(o_2 \leftrightarrow \neg(z_3 \wedge z_4))
\end{aligned}
$$

(c) MBD formulation

Figure 1: C17 circuit and selected observations from ISCAS85 scenarios

The paper assumes standard model-based diagnosis (MBD) definitions, following Reiter's seminal work [Reiter, 1987] and used in most modern references [Reiter, 1987; Siddiqi, 2011; Metodi *et al.*, 2012; Nica *et al.*, 2013; Metodi *et al.*, 2012]. As in recent work, the weak fault model (WFM) is assumed throughout. A system description SD is a set of first-order sentences [Reiter, 1987]. The system components, Comps, are a set of constants, $\mathsf{Comps} = \{c_1, \ldots, c_m\}$. Given a system description SD, composed of a set of components Comps, each component can be declared as *healthy* or *unhealthy*. For each component $c \in \mathsf{Comps}$, $\mathsf{Ab}(c) = 1$ if c is declared as unhealthy; otherwise $\mathsf{Ab}(c) = 0$. Similarly to earlier work [Feldman *et al.*, 2010; Metodi *et al.*, 2012], it is assumed that SD is represented as a CNF formula, namely:

$$\mathsf{SD} \triangleq \bigwedge_{c \in \mathsf{Comps}} (\mathsf{Ab}(c) \vee \mathcal{F}_c) \tag{1}$$

where $\mathcal{F}_c$ denotes the CNF encoding of component c.

Observations are used to represent situations where the behavior of the system is not the expected one. An observation Obs is defined as a finite set of first-order sentences [Reiter, 1987]. As with the system description, it is assumed that the observation can be encoded into CNF, as a set of unit clauses, and denoted Obs.

Definition 1 (Diagnosis Problem) *A system with description* SD *is faulty if its model is inconsistent with a given observation* Obs *when all components are declared healthy, i.e.:*

$$\mathsf{SD} \wedge \mathsf{Obs} \wedge \bigwedge_{c \in \mathsf{Comps}} \neg \mathsf{Ab}(c) \vDash \bot \tag{2}$$

The problem of diagnosis is to identify a set of components which, if declared unhealthy, make the system consistent with the observation. The problem of MBD is represented by the 3-tuple $\langle \mathsf{SD}, \mathsf{Comps}, \mathsf{Obs} \rangle$.

Definition 2 (Diagnosis) *Given an MBD problem* $\langle \mathsf{SD}, \mathsf{Comps}, \mathsf{Obs} \rangle$*, the set of components* $\Delta \subseteq \mathsf{Comps}$ *is a* diagnosis *if*

$$\mathsf{SD} \wedge \mathsf{Obs} \wedge \bigwedge_{c \in \Delta} \mathsf{Ab}(c) \wedge \bigwedge_{c \in \mathsf{Comps} \setminus \Delta} \neg \mathsf{Ab}(c) \nvDash \bot \tag{3}$$

A diagnosis Δ *is* minimal *if no proper subset* $\Delta' \subsetneq \Delta$ *is a diagnosis, and* Δ *is of* minimal cardinality *if there exists no other diagnosis* $\Delta' \subseteq \mathsf{Comps}$ *with* $|\Delta'| < |\Delta|$.

To model MBD with MaxSAT [Safarpour *et al.*, 2007; Feldman *et al.*, 2010], SD (see (1)) represents the *hard* clauses, whereas the *soft clauses* are unit clauses $(\neg \mathsf{Ab}(c))$, one for each component $c \in \mathsf{Comps}$. This is referred to as the *basic* MaxSAT encoding in this paper. Different MaxSAT solving approaches can then be applied. Alternatively, the soft clauses can be replaced by a cardinality constraint and solved iteratively with a SAT solver.

```
1 global: ⟨SD, Comps, Obs⟩
2 repeat
3 |  FindDominators()
4 |  FindBackboneComponents()
5 |  FindBlockedConnections()
6 |  if MaxNumberIterations() then break
7 until NoMoreChanges()
8 GenMaxsatModel()
```

Algorithm 1: MBD to MaxSAT compilation

Figure 1 summarizes the most often used SAT-based approach for MBD [Smith *et al.*, 2005; Feldman *et al.*, 2010; Nica *et al.*, 2013]. Recent work on SAT-based MBD [Metodi *et al.*, 2012; 2014] develops a more sophisticated model, by using logical equivalence between the unhealthy variable of a component and its associated CNF encoding, and also by exploiting structural properties of the system description, including graph dominators and sections.

3 Efficient MBD with MaxSAT

The novel approach for compiling MBD into MaxSAT is summarized in Algorithm 1. Lines 2 to 7 preprocess the system as described in the following sections. The objective is to generate a simpler MaxSAT instance than what would be achieved with the basic MaxSAT encoding (described in Section 2). The main loop is executed while additional changes are identified. Line 6 bounds the maximum number of iterations of the algorithm, so that the (quasi-quadratic) worst case running time is not observed. Line 8 generates the MaxSAT formula, using the basic encoding described above, but also taking into account the information generated by the preprocessing phase. The MBD to MaxSAT encoding proposed in this paper referred to as *Dominator-Oriented Encoding* (DOE), given the way dominators are exploited in the compilation process. The rest of this section details each preprocessing step.

3.1 Dominators, TLDs & Hard Components

A well-known technique to constrain the CNF encoding is to consider the immediate dominators of the circuit graph [Siddiqi and Huang, 2007; Metodi *et al.*, 2012].

Let O denote a special vertex to which every circuit output is connected. Vertex v is a dominator of vertex u if all paths from u to O include v [Lengauer and Tarjan, 1979]. Vertex v is an immediate dominator of u if every other dominator of u is also a dominator of v. A circuit gate is *dominated* if its immediate dominator is not O; otherwise it is *non-dominated*.

If a dominated gate is included in some minimal-cardinality diagnosis, then it can be replaced by its immediate (non-dominated) dominator gate [Siddiqi and Huang, 2007]. Thus, one can analyze solely the minimal cardinality diagnoses that do not involve dominated gates. These are referred to as *top-level diagnoses* (TLDs) [Siddiqi and Huang, 2007; Metodi *et al.*, 2012; 2014].

Definition 3 (Top-Level Diagnosis (TLD)) *A minimal cardinality diagnosis is a* top-level diagnosis *if it does not contain dominated gates.*

Recent work [Siddiqi and Huang, 2007; Siddiqi, 2011; Metodi *et al.*, 2012; 2014] focuses on computing TLDs. The other minimal cardinality diagnoses can be enumerated by iteratively replacing a non-dominated gate by a dominated gate, and then checking for consistency. This is in practice simpler than solving MaxSAT, since one call to a SAT solver on a simpler formula suffices. This approach is also assumed in this paper.

In the DOE MaxSAT problem formulation, immediate dominators are computed with a standard (quasi-)linear time algorithm [Lengauer and Tarjan, 1979]. In terms of the problem encoding, since dominated gates are not included in TLDs, these can be encoded as *hard* clauses, and referred to as *hard components*. Thus, unhealthy variables $\mathsf{Ab}(c)$ are only associated with non-dominated components, each of which is associated with a soft clause.

Proposition 1 *For any computed TLDs, dominated gates are declared healthy.*

Remark 1 *For computing TLDs (and minimal cardinality diagnoses), dominated gates are modeled as hard clauses.*

Remark 1 serves to simplify the MaxSAT problem instance, by reducing the number of soft clauses, since hard components *cannot* be declared unhealthy when computing TLDs.

3.2 Backbone Nodes & Blocked Edges

In practice, Remark 1 results in significant reduction in the number of used unhealthy variables if the number of dominated components is also significant[1]. More importantly, if the output value of a dominated component can be determined given its input values then, because the component *cannot* be declared unhealthy, the output value of the component can be encoded as a hard unit clause.

Definition 4 (Backbone Node (B-Node)) *A dominated component for which the output value is fixed for any TLD is a* backbone node *(B-Node).*

Example 1 *For the example in Figure 1, z_1 is always dominated. Thus, if $i_1 = i_3 = 1$, then $z_1 = 0$ for any TLD. In contrast, if $i_1 = 0$ (or $i_3 = 0$), then $z_1 = 1$ for any TLD.*

[1]Observe that components represent circuit gates. Moreover, each component can be viewed as a node in a directed graph.

The definition of backbone node mimics that of backbone literal [Monasson *et al.*, 1999; Slaney and Walsh, 2001]. Although there are recent practical algorithms for computing backbone literals [Janota *et al.*, 2015], in the DOE approach for MBD, backbone nodes are identified by value propagation through dominated components. In addition, the fact that dominated components can be declared backbone nodes allows the identification of connections that *cannot* be used for computing TLDs.

Definition 5 (Blocked Edge (B-Edge)) *A fanin edge e of a component g is* blocked *if the output value of g remains unchanged for any value assigned to the fanin edge e.*

Proposition 2 *For any computed TLD, if a fanin node of a component g is a B-Node and it is assigned the controlling value of g, then the other unassigned fanin edges of g are B-Edges.*

Example 2 *For the example in Figure 1, z_1 is dominated. If $i_1 = i_3 = 1$, then $z_1 = 0$. In this case, edge (z_3, o_1) is a B-edge. This information can lead to further simplifications, as illustrated below.*

3.3 Filtered Nodes and Edges

Blocked edges can now be used for determining nodes and edges which need not be added to the generated CNF formula.

Definition 6 (Filtered Edge) *An edge is* filtered *if it is blocked or if its fanout node is filtered.*

Definition 7 (Filtered Node) *A node is filtered if all of its fanout edges are filtered.*

Remark 2 *As the result of the preprocessing step of DOE compilation, filtered edges and nodes are* not *encoded into CNF.*

Let $\mathcal{F}$ denote the basic CNF encoding of the system (described in Section 2), consisting of CNF encoding the system and associated observation, respectively SD and Obs. Moreover, let $\mathcal{F}^f$ denote the CNF encoding of the system where filtered nodes are not encoded into CNF, respectively SD^f and Obs^f. Finally, let Δ be a computed minimal cardinality diagnosis using $\mathcal{F}$.

Proposition 3 *Δ is a TLD for SD given Obs iff Δ is a TLD for SD^f given Obs^f.*

Another important consequence of identifying filtered edges and nodes is that this enables detecting additional dominated components, which can lead to finding additional B-Nodes, B-Edges, and so additional filtered nodes and edges. As shown in Algorithm 1, this process is repeated while changes are observed. The immediate downside of this approach is that this results in a (quasi-)quadratic worst-case running time (due to computing dominators in quasi-linear time [Lengauer and Tarjan, 1979]). Observe that in the worst case: (i) the complexity of iteration of the loop is quasi-linear, due to the algorithm for finding dominators used [Lengauer and Tarjan, 1979]; and (ii) in each iteration most components in the system need to be analyzed. As a result, the proposed solution in Algorithm 1 is to bound the number of iterations by a constant.

Obs	Dominated	B-Nodes	B-Edges	Soft	CSs	$\lvert X\rvert$	$\lvert\mathcal{H}\rvert$	$\lvert\mathcal{S}\rvert$
19	$\{z_1, z_4, z_2\}$	$\{z_1\}$	$\{(z_2, z_3)\}$	$\{o_1, o_2, z_3\}$	$\{\langle z_3, z_4, o_1, o_2\rangle\}$	13	22	3
20	$\{z_1, z_4\}$	$\{z_1\}$	$\emptyset$	$\{o_1, o_2, z_2, z_3\}$	$\{\langle z_2, z_3, z_4, o_1, o_2\rangle\}$	14	22	4
51	$\{z_1, z_4, z_2, z_3\}$	$\{\bar{z}_1, \bar{z}_2, z_3, z_4\}$	$\{(z_2, z_3), (z_3, o_1)\}$	$\{o_1, o_2\}$	$\{\langle o_1\rangle, \langle o_2\rangle\}$	7	11	2
53	$\{z_1, z_4, z_3\}$	$\{\bar{z}_1\}$	$\{(z_3, o_1)\}$	$\{o_1, o_2, z_2\}$	$\{\langle o_1\rangle, \langle o_2, z_2, z_3, z_4\rangle\}$	13	22	3
63	$\{z_1, z_4, z_2, z_3\}$	$\{\bar{z}_1, z_2, z_3, \bar{z}_4\}$	$\{(z_2, z_3), (z_3, o_1), (z_3, o_2)\}$	$\{o_1, o_2\}$	$\{\langle o_1\rangle, \langle o_2\rangle\}$	7	11	2
					Basic encoding	17	25	6

Table 1: DOE for C17 example observations vs. basic encoding

Example 3 *For the case of Example 2, the fact that (z_3, o_1) is a B-edge, and so it is filtered, results in z_3 being declared dominated and so it is declared a hard component. Moreover, if $i_2 = 0$, then (z_2, z_3) becomes a B-edge. As a result, z_2 is now dominated, and so it is declared a hard component. Observe that given the assignments to i_1, i_2, i_3, and regardless of the assignments made to i_4, i_5, the compilation is able to declare z_1, z_2, z_3, z_4 as hard components. This leaves only o_1 and o_2 as the components that can be declared unhealthy for computing TLDs, given this concrete observation pattern.*

3.4 Induced Problem Decomposition

When computing TLDs, the identification of dominators, backbone nodes and blocked edges contribute to creating structural decompositions. Structural decompositions have been considered in MBD [Darwiche, 1998] but also in other domains [Bayardo Jr. and Pehoushek, 2000]. Depending on the observation, the DOE proposed in this paper can yield problem decompositions in the form of sets of connected components, which can be identified in linear time [Bayardo Jr. and Pehoushek, 2000]. If two connected components have (disjoint) unsatisfiable cores, these will represent *disjoint* unsatisfiable cores, which can be exploited by some recent core-guided MaxSAT algorithms [Ansótegui *et al.*, 2013; Morgado *et al.*, 2013].

3.5 Core-Guided MaxSAT Solving

As described above, the soft clauses of the MaxSAT formulation are unit and the complement of the unhealthy variable of each non-dominated component. Moreover, for many MBD problems, many of the components are *not* included in any minimal cardinality diagnosis, i.e. the value of $\mathsf{Ab}(c)$ is always 0. Core-guided MaxSAT algorithms can exploit this fact. Since these algorithms iteratively call a SAT solver using both the hard and the soft clauses, many of these irrelevant unhealthy variables will always be assigned value 0 and will not be included in computed cores. The experimental results (see Section 4) demonstrate that core-guided MaxSAT algorithms introduce significant performance gains over MaxSAT approaches based on iterative SAT solving [Metodi *et al.*, 2014].

3.6 Compilation Examples

Table 1 summarizes the compilation with DOE for C17 using the selected observations from Figure 1. Columns $|X|$, $|\mathcal{H}|$ and $|\mathcal{S}|$ denote, respectively, the number of variables, the number of hard clauses, and the number of soft clauses. As can be observed, for some observations the reductions are significant, in the total number of soft clauses, total number of clauses and total number of variables.

The compilation process is illustrated for observation 51 (see Figure 1). In this case, z_1 and z_4 are initially dominated (other components may be declared dominated as preprocessing phase progresses). Thus, both z_1 and z_4 are hard components (and so encoded with hard clauses). Moreover, since its inputs are fixed and z_1 cannot be declared unhealthy, z_1 is a B-node, and assigned value 0. Observe that, although z_4 is a hard component, it cannot be declared a B-node at this stage; the value of z_4 cannot be decided since $i_5 = 1$ and z_2 may take any value. Given the value assigned to z_1, (z_3, o_1) becomes a B-edge. As a result, z_3 is now dominated, and so a hard component. Given the value of i_2, z_3 also becomes a B-node, with value 1. Moreover, (z_2, z_3) can now be declared a B-edge. The immediate consequence is that z_2 becomes dominated (and so a hard component), and is also declared a B-node, with value 0. The value of z_2 also implies that the value of z_4 is 1. Given the above, z_1, z_2, z_3, z_4 are hard components, each of which is declared a B-node. In addition, the only unassigned variables are o_1 and o_2, each of which is associated with a soft unit clause. As can be observed, the compilation process also induces a decomposition of the original problem, such that each unassigned variable is located in a separate connected subgraph. As observed earlier, modern core-guided MaxSAT solvers can exploit decomposition into connected subgraphs.

4 Experimental Results

This section compares the latest version of SATbD/SCryptoDiagnoser (scrypto) [Metodi *et al.*, 2012; 2014], recently shown to outperform most, if not all, of earlier MBD approaches[2], against the DOE MaxSAT model proposed in this paper. For the generated MaxSAT formulas, some of the best performing MaxSAT solvers on partial MaxSAT instances were selected[3]. Concretely, the MaxSAT solvers considered were eva500a (eva) [Narodytska and Bacchus, 2014] and open-wbo-inc (wboinc) [Martins *et al.*, 2014]. Both solvers are core-guided and so are better

[2]Recent results [Nica *et al.*, 2013] suggest that alternatives approaches could be moderately more efficient. However, the scenarios considered involve a small number of errors (1 to 3), and scrypto is expected to excel for larger numbers of errors.

[3]The solvers are selected given the results of the 2014 MaxSAT evaluation, `http://maxsat.ia.udl.cat`, concretely on the industrial categories. The two best performing, publicly available, MaxSAT solvers were selected.

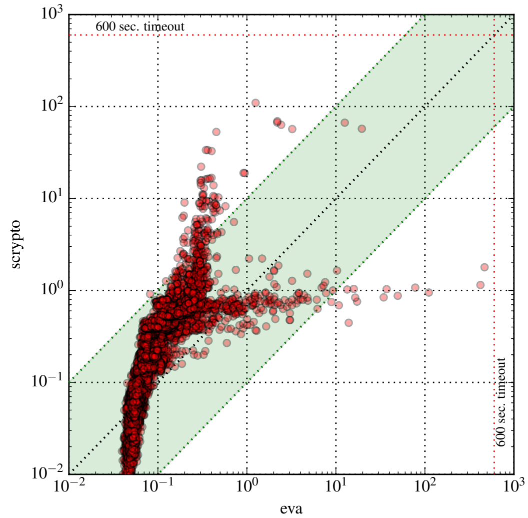


Figure 2: Scatter plot eva vs. scrypto for ISCAS85 suite

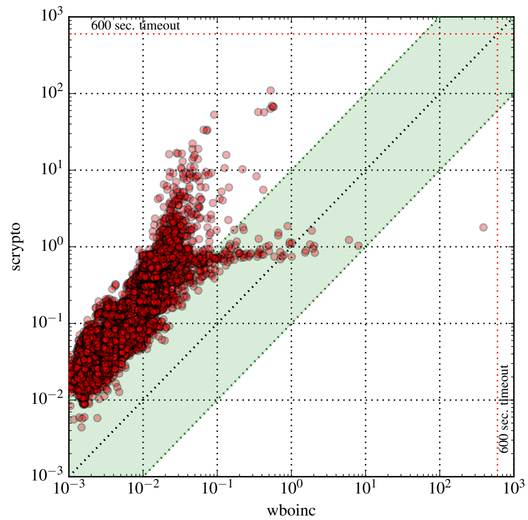


Figure 3: Scatter plot wboinc vs. scrypto for ISCAS85 suite

suited to exploit the encoding proposed in this paper (see Section 3.5). Because the iterative SAT solving approach used by scrypto can be considered to be solving MaxSAT, and it is not based on a portfolio of MaxSAT solvers, we opted not to use a portfolio MaxSAT solver [Ansótegui *et al.*, 2014].

The experiments were performed on a cluster of Linux servers, each with two Intel Xeon 2.60GHz processors and 64 GByte of physical memory. For all experiments, the time limit was set to 600s and the memory limit to 4GByte. The experiments focus on the run time for computing one minimal cardinality diagnosis for each scenario. As a result, the preprocessing time of scrypto is not accounted for. Similarly, to guarantee that the MaxSAT encoding time is negligible, the number of iterations of Algorithm 1 is limited to 2.

Most recent papers on approaches for MBD have focused on the ISCAS85 [Brglez and Fujiwara, 1985] problem instances [Williams and Ragno, 2007; Siddiqi and Huang, 2007; de Kleer, 2009; Feldman *et al.*, 2010; Siddiqi, 2011; Metodi *et al.*, 2012; Stern *et al.*, 2012; Nica *et al.*, 2013; Metodi *et al.*, 2014]. Given the improvements made to MBD approaches in recent years, most of these problem instances are now easy to solve. As a result, this paper starts by showing results for the ISCAS85 problem instances, confirming that these are indeed easy, and then considers a new suite of far more challenging problem instances.

16174 instances	scrypto	eva	wboinc
% Solved	100.0	100.0	100.0
% scrypto wins	—	23.4	0.1
% eva wins	76.0	—	0.0
% wboinc wins	99.9	100.0	—

Table 2: Statistics for ISCAS85 suite

5903 instances	scrypto	eva	wboinc
% Solved	62.4	89.7	90.6
% scrypto wins	—	2.2	0.4
% eva wins	97.8	—	13.4
% wboinc wins	99.6	86.6	—

Table 3: Statistics for ITC99 suite

4.1 ISCAS85 Suite

The experimental results for the ISCAS85 suite are summarized in Figures 2 and 3, and in Table 2. As can be observed, modulo a small number of outliers, the MaxSAT model proposed in this paper enables observable performance gains over scrypto. For example, wboinc outperforms scrypto in 99.9% of the instances, with performance gains that most often range between 1 and 3 orders of magnitude. Among the two MaxSAT solvers, wboinc shows better performance than eva, in every of the 16174 instances. As can be observed, both scrypto, eva and wboinc are able to solve every problem instance within the time limit of 600s. Among the instances for which scrypto outperforms the MaxSAT approaches, there is one scenario for which both eva and wboinc take more than 300s. For this instance, the DOE proposed in this paper is not as effective as the encoding used by scrypto. It should be noted that the SAT solvers used by scrypto (CryptoMiniSat [Soos *et al.*, 2009]), and by eva and wboinc (Glucose [Audemard and Simon, 2009] for both), although different, are based on the MiniSAT SAT solver[4] and represent the current state-of-the-art.

4.2 ITC99 Suite

The results in the previous section demonstrate that any of the ISCAS85 circuits with any existing scenario can be solved very efficiently by state-of-the-art MBD approaches. This section proposes the use of the more challenging ITC99 benchmark suite [Corno *et al.*, 2000][5]. Scenarios for the

[4]`https://github.com/niklasso/minisat`.

[5]`http://www.cad.polito.it/tools/itc99.html`. All large ITC99 circuits were considered, namely b14, b15, b17, b18, b19, b20, b21 and b22. Each memory element was replaced by

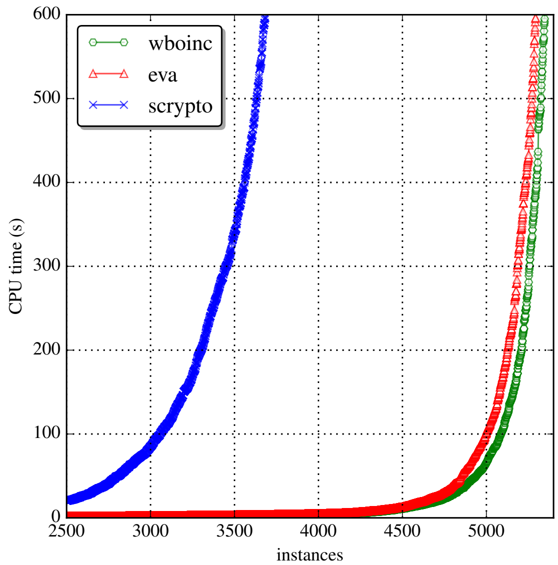


Figure 4: Cactus plot for ITC99 suite

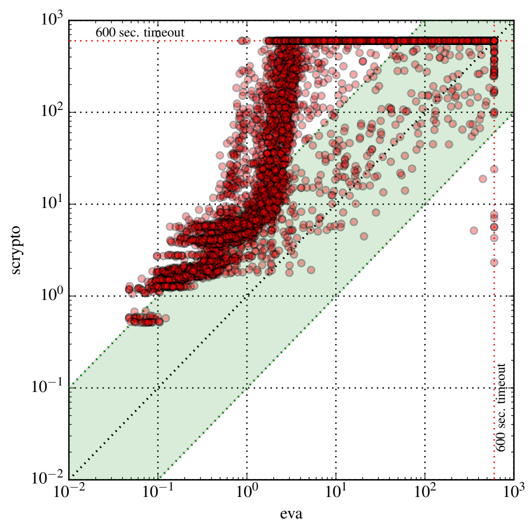


Figure 5: Scatter plot eva vs. scrypto for ITC99 suite

ITC99 benchmark suite were generated using a standard approach [Siddiqi, 2011], with the number of errors in the circuit's outputs ranging from 1 to 50. A total of 7903 scenarios were generated. However, scrypto is unable to preprocess two of the ITC99 circuits, b18 and b19, for which 2000 scenarios were generated. As a result, the experiments report only results for 5903 scenarios, although each of the MaxSAT formulas for the scenarios of both b18 and b19 can be solved in a matter of seconds by both MaxSAT solvers, eva and wboinc. Observe that the preprocessing time of scrypto for these circuits is far from negligible (ranging between a few tens and more than 100 seconds). This is far larger than the DOE compilation time. As a result, both times are not accounted for.

The experimental results for the ITC99 suite are summarized in Figures 4, 5, and 6, and in Table 3. The results confirm that the new scenarios are far more challenging for existing MBD approaches, with scrypto being able to solve only 62.4% of the problem instances within the time limit. The DOE proposed in this paper enables MaxSAT solvers to perform significantly better than scrypto, with eva and wboinc being able to solve, respectively 89.7% and 90.6% of the instances. More importantly, the MaxSAT solvers are most often able to outperform scrypto with gains than range between 1 and more than 3 orders of magnitude. As before, wboinc outperforms eva for most instances, and consistently outperforms scrypto. In contrast, scrypto outperforms wboinc for only 0.4% of the instances (among those solved by the two solvers). The results also suggest that a portfolio of approaches is unlikely to outperform the DOE into MaxSAT.

an input/output pair, as is standard for example in testing.

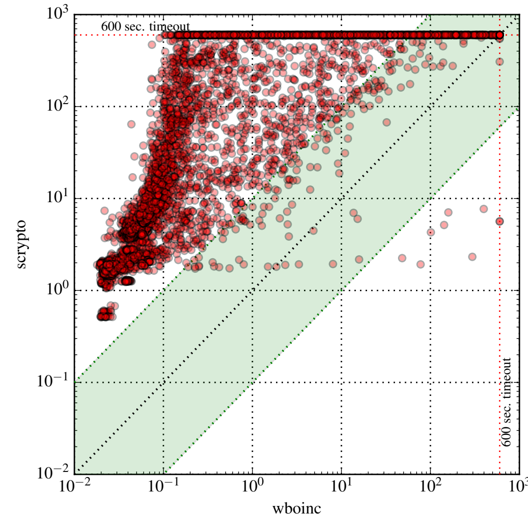


Figure 6: Scatter plot wboinc vs. scrypto for ITC99 suite

5 Conclusions

This paper proposes a novel approach for compiling MBD into MaxSAT. This novel approach addresses the computation of TLDs, and emphasizes techniques for effectively constraining the resulting MaxSAT formulation. By building on dominators, the proposed encoding introduces several new concepts, including hard and backbone nodes, blocked edges, and filtered nodes and edges. Experimental results obtained on standard MBD scenarios show gains that most often exceed one order of magnitude over SATbD/SCryptoDiagnoser, one of the best performing MBD approaches [Metodi *et al.*, 2012; 2014]. Given that these standard scenarios turn out to be fairly simple for the new DOE approach, the paper also develops a new MBD suite using more challenging circuits [Corno *et al.*, 2000]. The experimental results on the new MBD suite show conclusive gains over SATbD/SCryptoDiagnoser. These results are even more significant, since both SATbD/SCryptoDiagnoser and the two core-guided MaxSAT solvers are SAT-based, and use state-of-the-art SAT solvers. Also, SATbD/SCryptoDiagnoser uses BEE, a state-of-the-art finite domain constraint compiler [Metodi and Codish, 2012].

Future work will explore further optimizations for the DOE into MaxSAT compilation approach. One line of research will be to integrate restricted versions of the optimizations proposed in SATbD/SCryptoDiagnoser.

References

[Ansótegui *et al.*, 2013] Carlos Ansótegui, Maria Luisa Bonet, and Jordi Levy. SAT-based MaxSAT algorithms. *Artif. Intell.*, 196:77–105, 2013.

[Ansótegui *et al.*, 2014] Carlos Ansótegui, Yuri Malitsky, and Meinolf Sellmann. Maxsat by improved instance-specific algorithm configuration. In *AAAI*, pages 2594–2600, 2014.

[Audemard and Simon, 2009] Gilles Audemard and Laurent Simon. Predicting learnt clauses quality in modern SAT solvers. In *IJCAI*, pages 399–404, 2009.

[Bauer, 2005] Andreas Bauer. Simplifying diagnosis using LSAT: A propositional approach to reasoning from first principles. In *CPAIOR*, pages 49–63, 2005.

[Bayardo Jr. and Pehoushek, 2000] Roberto J. Bayardo Jr. and Joseph Daniel Pehoushek. Counting models using connected components. In *AAAI/IAAI*, pages 157–162, 2000.

[Biere *et al.*, 2009] Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors. *Handbook of Satisfiability*. IOS Press, 2009.

[Brglez and Fujiwara, 1985] F. Brglez and H. Fujiwara. A neutral list of 10 combinational benchmark circuits and a target translator in FORTRAN. In *ISCAS*, pages 695–698, Jun. 1985.

[Corno *et al.*, 2000] Fulvio Corno, Matteo Sonza Reorda, and Giovanni Squillero. RT-level ITC'99 benchmarks and first ATPG results. *IEEE Design & Test of Computers*, 17(3):44–53, 2000.

[Darwiche, 1998] Adnan Darwiche. Model-based diagnosis using structured system descriptions. *J. Artif. Intell. Res. (JAIR)*, 8:165–222, 1998.

[Darwiche, 2001] Adnan Darwiche. Decomposable negation normal form. *J. ACM*, 48(4):608–647, 2001.

[de Kleer and Williams, 1987] Johan de Kleer and Brian C. Williams. Diagnosing multiple faults. *Artif. Intell.*, 32(1):97–130, 1987.

[de Kleer *et al.*, 1992] Johan de Kleer, Alan K. Mackworth, and Raymond Reiter. Characterizing diagnoses and systems. *Artif. Intell.*, 56(2-3):197–222, 1992.

[de Kleer, 2009] Johan de Kleer. Minimum cardinality candidate generation. In *DX*, pages 397–402, 2009.

[Feldman *et al.*, 2010] A. Feldman, G. Provan, J. de Kleer, S. Robert, and A. van Gemund. Solving model-based diagnosis problems with Max-SAT solvers and vice versa. In *DX*, pages 185–192, 2010.

[Hofer *et al.*, 2013] Birgit Hofer, André Riboira, Franz Wotawa, Rui Abreu, and Elisabeth Getzner. On the empirical evaluation of fault localization techniques for spreadsheets. In *FASE*, pages 68–82, 2013.

[Janota *et al.*, 2015] Mikolás Janota, Inês Lynce, and Joao Marques-Silva. Algorithms for computing backbones of propositional formulae. *AI Commun.*, 28(2):161–177, 2015.

[Jose and Majumdar, 2011] Manu Jose and Rupak Majumdar. Cause clue clauses: error localization using maximum satisfiability. In *PLDI*, pages 437–446, 2011.

[Lengauer and Tarjan, 1979] Thomas Lengauer and Robert Endre Tarjan. A fast algorithm for finding dominators in a flowgraph. *ACM Trans. Program. Lang. Syst.*, 1(1):121–141, 1979.

[Martins *et al.*, 2014] Ruben Martins, Saurabh Joshi, Vasco M. Manquinho, and Inês Lynce. Incremental cardinality constraints for MaxSAT. In *CP*, pages 531–548, 2014.

[Metodi and Codish, 2012] Amit Metodi and Michael Codish. Compiling finite domain constraints to SAT with BEE. *TPLP*, 12(4-5):465–483, 2012.

[Metodi *et al.*, 2012] Amit Metodi, Roni Stern, Meir Kalech, and Michael Codish. Compiling model-based diagnosis to Boolean satisfaction. In *AAAI*, 2012.

[Metodi *et al.*, 2014] Amit Metodi, R. Stern, Meir Kalech, and Michael Codish. A novel sat-based approach to model based diagnosis. *J. Artif. Intell. Res. (JAIR)*, 51:377–411, 2014.

[Monasson *et al.*, 1999] Rémi Monasson, Riccardo Zecchina, Scott Kirkpatrick, Bart Selman, and Lidror Troyansk. Determining computational complexity from characteristic 'phase transitions'. *Nature*, 400:133–137, July 1999.

[Morgado *et al.*, 2013] António Morgado, Federico Heras, Mark H. Liffiton, Jordi Planes, and Joao Marques-Silva. Iterative and core-guided MaxSAT solving: A survey and assessment. *Constraints*, 18(4):478–534, 2013.

[Narodytska and Bacchus, 2014] Nina Narodytska and Fahiem Bacchus. Maximum satisfiability using core-guided maxsat resolution. In *AAAI*, pages 2717–2723, 2014.

[Nica and Wotawa, 2012] Iulia Nica and Franz Wotawa. ConDiag - computing minimal diagnoses using a constraint solver. In *DX*, pages 185–192, 2012.

[Nica *et al.*, 2013] Iulia Nica, Ingo Pill, Thomas Quaritsch, and Franz Wotawa. The route to success - a performance comparison of diagnosis algorithms. In *IJCAI*, 2013.

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

[Safarpour *et al.*, 2007] Sean Safarpour, Hratch Mangassarian, Andreas G. Veneris, Mark H. Liffiton, and Karem A. Sakallah. Improved design debugging using maximum satisfiability. In *FMCAD*, pages 13–19, 2007.

[Siddiqi and Huang, 2007] Sajjad Ahmed Siddiqi and Jinbo Huang. Hierarchical diagnosis of multiple faults. In *IJCAI*, pages 581–586, 2007.

[Siddiqi, 2011] Sajjad Ahmed Siddiqi. Computing minimum-cardinality diagnoses by model relaxation. In *IJCAI*, pages 1087–1092, 2011.

[Slaney and Walsh, 2001] John K. Slaney and Toby Walsh. Backbones in optimization and approximation. In *IJCAI*, pages 254–259, 2001.

[Smith *et al.*, 2005] Alexander Smith, Andreas G. Veneris, Moayad Fahim Ali, and Anastasios Viglas. Fault diagnosis and logic debugging using Boolean satisfiability. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 24(10):1606–1621, 2005.

[Soos *et al.*, 2009] Mate Soos, Karsten Nohl, and Claude Castelluccia. Extending SAT solvers to cryptographic problems. In *SAT*, pages 244–257, 2009.

[Stern *et al.*, 2012] Roni Tzvi Stern, Meir Kalech, Alexander Feldman, and Gregory M. Provan. Exploring the duality in conflict-directed model-based diagnosis. In *AAAI*, 2012.

[Williams and Ragno, 2007] Brian C. Williams and Robert J. Ragno. Conflict-directed A^* and its role in model-based embedded systems. *Discrete Applied Mathematics*, 155(12):1562–1595, 2007.