# Solving Model-Based Diagnosis Problems with Max-SAT Solvers and Vice Versa

**Alexander Feldman[1,4], Gregory Provan[2], Johan de Kleer[3], Stephan Robert[4], and Arjan van Gemund[1]**

[1] *Delft University of Technology, Mekelweg 4, 2628 CD, Delft, The Netherlands*
*{a.b.feldman,a.j.c.vangemund}@tudelft.nl*
[2] *University College Cork, College Road, Cork, Ireland*
*g.provan@cs.ucc.ie*
[3] *Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, CA 94304, USA*
*dekleer@parc.com*
[4] *Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud*
*Route de Cheseaux 1, CH-1401 Yverdon-les-Bains, Switzerland*
*stephan.robert@heig-vd.ch*

## Abstract

In this paper we bring closer computation of consistency-based cardinality-minimal diagnosis and solving Max-SAT. We propose two algorithms for translating between those: (1) Diorama (DIagnOsis-based algoRithm for mAx-sat optiMizAtion) for translating cardinality-minimal consistency based diagnosis to Max-SAT and (2) Meridian (Max-sat-basEd algoRIthm for DIAgNosis) for the other way around. While the former approach has been studied, solving Max-SAT instances with a diagnostic solver is, to the best of our knowledge, novel. We configure Meridian with the Stochastic Local Search (SLS) solvers from the UBCSAT suite, perform extensive experimentation on fault-models of the 74XXX/ISCAS85 circuits and compare the resulting optimality to the one of the stochastic MBD algorithm Safari. The results show that the optimality of Safari is up to several-orders-of-magnitude better than that of the SLS-based Max-SAT solvers. We configure Diorama with Safari and experiment on instances from the Max-SAT competitions. While the performance of Diorama/Safari on crafted Max-SAT problems is slightly worse compared to UBCSAT, Diorama/Safari outperforms at least several-orders-of-magnitude all UBCSAT algorithms on small industrial Max-SAT instances.

## 1 INTRODUCTION

Model-Based Diagnosis (MBD) inference algorithms for propositional diagnosis models have, in general, been created specifically for the diagnosis problem, e.g., GDE (de Kleer and Williams, 1987), and more recently Safari (Feldman *et al.*, 2010). These algorithms have been improved over the years by taking advantage of properties of the diagnosis problem, e.g., by focusing on the most likely diagnoses (de Kleer, 1990), or using a notion of continuity of the diagnosis space (Feldman *et al.*, 2010).



With regard to domain-independent problems, significant progress has been made in developing powerful solvers for the *satisfiability* (SAT) problem, e.g., (Gomes *et al.*, 2007). This success has prompted the use of SAT-solvers for many other problems, such as planning (Castellini *et al.*, 2003) and circuit test-case generation (Iyer *et al.*, 2003). Using a SAT-solver for another (non-SAT) problem $P$ entails rewriting $P$ in SAT format; although this rewriting process can increase the size of the problem, the high efficiency of SAT solvers often makes the rewriting process worthwhile.

MBD is a version of propositional abduction, which is a more complex problem than SAT. Hence, although one can make calls to a SAT solver during the process of computing diagnoses (e.g., Safari uses an incomplete SAT solver), one cannot use a standard SAT solver directly for diagnostic inference. This article shows how one can use an extension of the SAT problem, called Max-SAT, to solve MBD problems.

Max-SAT is an optimization extension of SAT. Given a formula $\Phi$ in Conjunctive Normal Form (CNF), a Max-SAT (Hoos and Stützle, 2004) solution is a variable assignment that maximizes the number of satisfied clauses in $\Phi$ (in most cases of interest $\Phi$ is unsatisfiable, otherwise any variable assignment which satisfies $\Phi$ is also a Max-SAT solution). In partial Max-SAT, some of the clauses in $\Phi$ are designated as hard, the others are "soft". A solution to the partial Max-SAT problem should satisfy all "hard" clauses and maximize the number of satisfied "soft" clauses. Similarly, in weighted

Max-SAT a weight is assigned to each clause in $\Phi$ and a solution maximizes the sum of the weights of the satisfied clauses.

The contributions of this paper are as follows. (1) We are the first to cast the Max-SAT problem as an MBD problem, for which we propose an algorithm called DIORAMA (DIagnOsis-based algoRithm for mAx-sat optiMizAtion). (2) We show that DIORAMA/SAFARI outperforms the traditional UBCSAT (Tompkins and Hoos, 2005) Max-SAT algorithms by at least two-orders-of-magnitude on a class of industrial Max-SAT problems, even though it performs slightly worse than Stochastic Local Search (SLS) Max-SAT algorithms on crafted Max-SAT competition problems. (3) We propose an algorithm, called MERIDIAN (Max-sat-basEd algoRIthm for DIAgNosis), that translates an MBD problem to a Max-SAT problem. (4) We empirically show that MERIDIAN configured with traditional Max-SAT is less optimal than specialized MBD solvers such as SAFARI thereby revealing a large class of Max-SAT problems that expose continuous properties amenable to greedy algorithms like DIORAMA/SAFARI.

## 2 RELATED WORK

MBD has resemblance to Max-SAT (Hoos and Stützle, 2004) and we have conducted extensive experimentation with both complete Max-SAT (partial and weighted) and Max-SAT based on Stochastic Local Search (SLS). Empirical evidence shows that although Max-SAT can compute diagnoses in many of the cases, the performance of Max-SAT degrades when increasing the circuit size or the cardinality of the injected faults.

Fu and Malik (2006) construct a partial Max-SAT algorithm that uses UNSAT cores provided by SAT solvers. The algorithm of Fu and Malik iteratively relaxes UNSAT cores until the CNF input becomes satisfiable. The difference from their approach and DIORAMA is that they do not explicitly use a diagnostic algorithm to find a single minimal unsatisfiable core.

An interesting approach to solving Max-SAT is proposed by de Givry, *et al* (2003) where they cast a Max-SAT problem as a weighted constraint satisfaction problem. On the other side, solving the diagnosis problem as a COP (Constraint Optimization Problem) is well-known from Williams and Ragno (2007).

On the side of solving diagnosis with Max-SAT, Kutsuna *et al.* (2009) use a partial Max-SAT algorithm to solve several diagnostic automotive problems. Similarly, Chen *et al.* (2009) use partial Max-SAT to solve problems of debugging sequential circuits. All these approaches differ from ours in that they solve specific diagnostic problems as opposed to empirically studying the general performance characteristics of Max-SAT and diagnostic algorithms.

## 3 TECHNICAL BACKGROUND

A *model* of an artifact is represented as a propositional formula over some set of variables. We discern subsets of these variables as *assumable* and *observable*.

**Definition 1** (Diagnostic System)**.** A diagnostic system DS is defined as the triple DS $= \langle$SD, COMPS, OBS$\rangle$, where SD is a propositional theory over a set of variables $V$, COMPS $\subseteq V$, OBS $\subseteq V$, COMPS is the set of assumables, and OBS is the set of observables.

Throughout this paper we assume that OBS $\cap$ COMPS $= \emptyset$ and SD $\not\models \perp$.

Not all propositional theories used as system descriptions are of interest to MBD. For example, models with ignorance of abnormal behavior are also known as weak-fault models.

**Definition 2** (Weak-Fault Model)**.** A diagnostic system DS $= \langle \mathrm{SD}, \mathrm{COMPS}, \mathrm{OBS} \rangle$ belongs to the class **WFM** iff for COMPS $= \{h_1, h_2, \ldots, h_n\}$, SD is equivalent to $(h_1 \Rightarrow F_1) \wedge (h_2 \Rightarrow F_2) \wedge \ldots \wedge (h_n \Rightarrow F_n)$ and COMPS$\cap V' = \emptyset$, where $V'$ is the set of all variables appearing in the propositional formulae $F_1, F_2, \ldots, F_n$.

Modeling of faults makes the problem of computing diagnoses more complex (de Kleer *et al.*, 1992), but can increase the precision of a diagnostic algorithm. Models that have knowledge of faults are formalized below.

**Definition 3** (Strong-Fault Model)**.** A diagnostic system DS $= \langle \mathrm{SD}, \mathrm{COMPS}, \mathrm{OBS} \rangle$ belongs to the class **SFM** iff SD is equivalent to $(h_1 \Rightarrow F_{1,1}) \wedge (\neg h_1 \Rightarrow F_{1,2}) \wedge \ldots \wedge (h_n \Rightarrow F_{n,1}) \wedge (\neg h_n \Rightarrow F_{n,2})$ such that $1 \leq i, j \leq n, k \in \{1, 2\}$, $\{h_i\} \subseteq$ COMPS, $F_{\{j,k\}}$ is a propositional formula, and none of $h_i$ appears in $F_{j,k}$.

In this paper, in addition to **WFM**, we experiment with stuck-at-zero (S-A-0) and stuck-at-one (S-A-1) models. S-A-0 and S-A-1 are subclasses of **SFM** (Feldman *et al.*, 2009) in which the output of a malfunctioning component is assumed either $\perp$ or $\top$.

**Definition 4** (Diagnosis)**.** Given a diagnostic system DS $= \langle \mathrm{SD}, \mathrm{COMPS}, \mathrm{OBS} \rangle$, an observation $\alpha$ over some variables in OBS, and a conjunction of literals $\omega$, $\omega$ is a diagnosis iff SD $\wedge\, \alpha \wedge \omega \not\models \perp$.

Given a conjunction of literals $\omega$ we denote the set of negative literals in $\omega$ as $Lit^-(\omega)$.

**Definition 5** (Subset-Minimal Diagnosis)**.** A diagnosis $\omega^\subseteq$ is defined as subset-minimal, if no other diagnosis $\tilde{\omega}^\subseteq$, $\tilde{\omega}^\subseteq \neq \omega^\subseteq$, exists such that $Lit^-(\tilde{\omega}^\subseteq) \subset Lit^-(\omega^\subseteq)$.

In the MBD literature, a range of types of "preferred" diagnosis has been proposed. In the following definition we consider the important from the practical perspective cardinality ordering.

**Definition 6** (Cardinality-Minimal Diagnosis)**.** A diagnosis $\tilde{\omega}^\leq$ is defined as cardinality-minimal if no other diagnosis $\tilde{\omega}^\leq$, $\tilde{\omega}^\leq \neq \tilde{\omega}^\leq$, exists such that $Lit^-(\tilde{\omega}^\leq) < Lit^-(\omega^\leq)$.

On the Max-SAT side we leave out all definitions as there is an agreement in the literature. Note that most SAT and Max-SAT solvers (as well as many diagnostic solvers) accept SD in CNF only. Any propositional formula can be converted to CNF taking into consideration a number of complexity and other issues (Feldman *et al.*, 2010).

## 4 MBD FRAMED AS MAX-SAT

In this section we demonstrate the use of Max-SAT for solving MBD problems.

### 4.1 A Max-SAT-Based MBD Algorithm

We propose an algorithm, called MERIDIAN (Max-sat-basEd algoRIthm for DIAgNosis), for computing cardinality-minimal diagnoses (see Def. 6). MERIDIAN uses the approach of Sang *et al.* (2007) for encoding Most Probable Explanation (MPE) as Max-SAT. Computing MPE is identical to computing a most-probable diagnosis in a more general framework. Algorithm 1 computes diagnoses by calling a Max-SAT oracle.

Note that the diagnostic problems we solve in this paper can be translated to multiple optimization problems which can be solved with SAT-based methods (Giunchiglia and Maratea, 2006). The Maximum Satisfiable Subset (MSS) problem, for example, is dual to the Minimal Unsatisfiable Subset problem (Bailey and Stuckey, 2005) and the two can be solved with Max-SAT and Min-UNSAT solvers, respectively (Liffiton and Sakallah, 2005). From those, we have found preference in the research community towards Max-SAT, and for practical reasons we therefore compare SAFARI to Max-SAT.

**Algorithm 1** MERIDIAN: an algorithm for MBD based on weighted Max-SAT

1: **function** MERIDIAN(DS, $\alpha$) **returns** a set of diagnoses
  **inputs:** DS, diagnostic system
    DS $= \langle \text{SD}, \text{COMPS}, \text{OBS} \rangle$
    $\alpha$, term, observation
  **local variables:** $W$, set of weight and clause pairs
    $\Omega$, set of diagnoses
    $\omega$, diagnosis term
    $c_i$, clause
    $h_i$, variable
2: **for all** $c_i \in$ CLAUSES(SD) **do**
3: $W \leftarrow W \cup \langle \infty, c_i \rangle$
4: **end for**
5: **for all** $h_i \in$ COMPS **do**
6: $W \leftarrow W \cup \langle 1, h_i \rangle$
7: **end for**
8: **while** $\omega \leftarrow$ MAX-SAT($W$) **do**
9: $W \leftarrow W \cup \langle \infty, \neg\omega \rangle$
10: $\Omega \leftarrow \Omega \cup \omega$
11: **end while**
12: **return** $\Omega$
13: **end function**

Algorithm 1 adds a unit clause with weight 1 for each assumable (line 6). The weight of each input clause is set to a value greater than the number of all assumables (line 3). The loop in lines 8 – 11 computes a diagnosis with a call to Max-SAT and if a diagnosis exists, it is added to the result (line 10), and its negation is added to the original set of clauses (line 9) to prevent subsequent computation of the same diagnosis. Note that the negation of a term is, conveniently, a clause.

Depending on the implementation of the MAX-SAT call in line 8 of Alg. 1 we have a family of MAX-SAT algorithms for diagnosis: (1) if MAX-SAT is a partial Max-SAT solver, Alg. 1 computes diagnoses ordered by cardinality; (2) if MAX-SAT is a weighted Max-SAT solver, Alg. 1 computes diagnoses ordered by probability; and (3) if MAX-SAT is based on SLS, not every iteration of the main loop yields a diagnosis. We have run extensive experiments with all three Max-SAT variants, which we describe in the following sub-sections.

### 4.2 Experimental Results with SLS Max-SAT

In the experiments that follow we compare the optimality of MERIDIAN configured with a number of SLS-based Max-SAT algorithms from the UBCSAT suite (Tompkins and Hoos, 2005). We compare the results to SAFARI, a state-of-the-art stochastic MBD algorithm (Feldman *et al.*, 2010).

The following issues complicate the use of SLS Max-SAT in diagnostic algorithms:

- There is no simple termination criterion in diagnostic algorithms based on SLS Max-SAT, i.e., we keep the local diagnosis and restart SAFARI after a number of successive "unsuccessful" flips, while there is no notion of "unsuccessful" flip (from the viewpoint of diagnosis) in Max-SAT. As we will see from our experimentation, flipping a variable which decreases the weight (or number) of currently satisfied clauses may be necessary to escape plateaus and/or local optima, hence the accumulation of such flips cannot be used as a termination criterion;
- Diagnostic Max-SAT problems have two type of constraints: hard and soft. The hard constraints are the clauses of the original ("nominal") model, while the soft constraints are the unit clauses received from the assumable variables. An SLS Max-SAT algorithm does not distinguish between those hard and soft clauses; if such an algorithm guaranteed the satisfaction of the hard-constraints it would be classified as hybrid and not stochastic.

These reasons make the use of algorithms based on SLS Max-SAT problematic in practical diagnosis. Despite that, we have conducted extensive experimentation with UBCSAT in order to evaluate the potential of SLS Max-SAT in MBD.

To overcome the termination problems with SLS Max-SAT, for the following experiments, we have chosen observations leading to known single faults.

Table 1: Optimality of SLS-based Max-SAT MERIDIAN and SAFARI on 74XXX/ISCAS85 **WFMs** and the number of steps (in parentheses) in which this optimality has been achieved

| Name | RGSAT | Schöening | CDRW | URW | IRoTS | RoTS | G2WSAT | SAPS[a] | SAPS[b] | Adaptive Novelty$^+$ | Novelty$^+$ | Novelty | WalkSAT/TABU | WalkSAT | HSAT | GWSAT | GSAT | SAFARI |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 74182 | 1 | 1 | 1 | 9 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | (4331) | (163) | (501) | (29896) | (1265) | (19) | (40) | (40) | (76) | (42260) | (28) | (56) | (21) | (37) | (26) | (186) | (38) | (2) |
| 74L85 | 1 | 1 | 1 | – | 1 | 3 | 1 | 1 | 1 | 1 | 1 | 1 | 5 | 1 | 1 | 1 | 1 | 1 |
| | (65623) | (276) | (353) | (–) | (7236) | (27) | (39) | (199) | (143) | (80782) | (9815) | (1267) | (47) | (116) | (34) | (629) | (34) | (4) |
| 74283 | 1 | 1 | 1 | – | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 4 | 1 | 1 | 1 | 1 | 1 |
| | (63429) | (1134) | (573) | (–) | (3644) | (26) | (21806) | (166) | (112) | (25003) | (31900) | (435) | (35) | (1034) | (29) | (187) | (32) | (2) |
| 74181 | 4 | 1 | 1 | – | 1 | 13 | 1 | 1 | 1 | 1 | 1 | 1 | 10 | 1 | 3 | 1 | 3 | 1 |
| | (48395) | (4525) | (6542) | (–) | (49967) | (1498) | (1814) | (146) | (453) | (1045) | (81883) | (2060) | (75) | (2123) | (67) | (1139) | (54) | (3) |
| c432 | 28 | 1 | 1 | – | 4 | 36 | 4 | 1 | 1 | 7 | 1 | 8 | – | 1 | 13 | 1 | 16 | 1 |
| | (43176) | (1866) | (7452) | (–) | (82496) | (101) | (10262) | (806) | (564) | (20211) | (75488) | (16393) | (–) | (905) | (133) | (24164) | (167) | (2) |
| c499 | – | 1 | 1 | – | 29 | – | 6 | 1 | 1 | 7 | 5 | 13 | – | 1 | – | 1 | – | 1 |
| | (–) | (24203) | (95242) | (–) | (99999) | (–) | (98683) | (42534) | (66519) | (2112) | (87589) | (49516) | (–) | (47102) | (–) | (45465) | (–) | (1) |
| c880 | – | 1 | 1 | – | – | – | 26 | 1 | 1 | 35 | 23 | 30 | – | 1 | – | 1 | 46 | 1 |
| | (–) | (36269) | (8388) | (–) | (–) | (–) | (9561) | (4244) | (1551) | (5680) | (80926) | (34989) | (–) | (9825) | (–) | (27026) | (304) | (5) |
| c1355 | – | 2 | 1 | – | – | – | 44 | 1 | 1 | 52 | 51 | 53 | – | 1 | – | 1 | – | 1 |
| | (–) | (93549) | (34235) | (–) | (–) | (–) | (83944) | (9670) | (4975) | (84777) | (67454) | (85365) | (–) | (34786) | (–) | (93656) | (–) | (5) |
| c1908 | 240 | – | – | – | – | – | 91 | 1 | 1 | 94 | 98 | 101 | – | 1 | 125 | 1 | 139 | 1 |
| | (68649) | (–) | (–) | (–) | (–) | (–) | (97567) | (95942) | (19664) | (5142) | (79442) | (43541) | (–) | (50660) | (654) | (94695) | (643) | (6) |
| c2670 | – | – | – | – | – | – | 122 | 1 | 1 | 126 | 135 | 144 | – | 1 | 173 | 1 | 177 | 1 |
| | (–) | (–) | (–) | (–) | (–) | (–) | (93799) | (9547) | (6635) | (19000) | (67042) | (6532) | (–) | (83292) | (1008) | (85581) | (976) | (5) |
| c3540 | – | – | – | – | – | – | 167 | 1 | 1 | 170 | 177 | 203 | – | 1 | – | 16 | – | 1 |
| | (–) | (–) | (–) | (–) | (–) | (–) | (79188) | (16376) | (17776) | (88052) | (68342) | (68663) | (–) | (99209) | (–) | (93639) | (–) | (9) |
| c5315 | – | – | – | – | – | – | 261 | 1 | 1 | 270 | 275 | 329 | – | 14 | – | 70 | – | 1 |
| | (–) | (–) | (–) | (–) | (–) | (–) | (78093) | (79438) | (48388) | (64079) | (96973) | (48497) | (–) | (98557) | (–) | (89462) | (–) | (9) |
| c6288 | – | – | – | – | – | – | 329 | 4 | 8 | 360 | 372 | – | – | 58 | – | – | – | 1 |
| | (–) | (–) | (–) | (–) | (–) | (–) | (23191) | (79807) | (73752) | (36373) | (91636) | (–) | (–) | (94673) | (–) | (–) | (–) | (3) |
| c7552 | – | – | – | – | – | – | 403 | 1 | 1 | 412 | 440 | 449 | – | 115 | 492 | 181 | 507 | 1 |
| | (–) | (–) | (–) | (–) | (–) | (–) | (12905) | (77264) | (42284) | (39046) | (90485) | (32732) | (–) | (90449) | (2543) | (99340) | (2552) | (11) |

[a] Clause penalties are initialized to the clause weights and smoothed back to their initial values.
[b] Clause penalties are initialized to the clause weights.

For each **WFM** we have chosen 50 observations. We have configured the SLS Max-SAT search to terminate after 100 000 variable flips and we have modified Alg. 1 to terminate after 10 calls to Max-SAT, this 10 runs. The resulting optimality of algorithms based on SLS Max-SAT in computing single fault diagnoses is shown in Table 1. We have run experiments with all algorithms or algorithm variants implemented by the UBCSAT suite, covering the algorithms RGSAT, Schöening, CDRW, through GSAT.

The data in Table 1 show best cases. From each of the 500 Max-SAT invocations per algorithm/circuit (50 single faults, 10 runs per experiment) we have (1) ignored all results which do not satisfy all hard constraints, (2) recorded the best diagnostic cardinality achieved in the hill climbing (recall that these are single-faults hence the best result is 1) and (3) recorded the number of steps (bit flips) in which this best diagnostic cardinality was achieved (the number of bit-flips are given in parentheses below the optimality number in Table 1).

Table 1 shows the generally poor performance of SLS Max-SAT algorithms. In most of the cases the algorithm could either never satisfy all hard-constraints or achieved increasingly worse cardinality with the growth of the circuit. Exceptions are the two variants of SAPS (Hutter *et al.*, 2002) and we attribute this relatively good optimality of SAPS to its mechanism for assigning and updating weights to clauses based on the clause length. Recall that in our diagnostic problems clauses of assumable literals have unit weights while hard-constraints have weights greater than the number of assumable literals. Despite that, in the best case for c7552, SAPS needed 77 264 bit flips to find the optimal single-fault diagnosis. In comparison Safari performed 11 bit flips, and although an LTMS/SAT consistency check of Safari is strictly more expensive than the consistency checking of SLS Max-SAT (the former is worst-case NP-hard while the latter is in P), Safari is computationally more efficient on average.

Figure 1 illustrates the progress of two SLS Max-SAT invocations. The Conflict-Directed Random Walk (CDRW) (Papadimitriou, 1991) starts with a random variable assignment and flips the most profitable (for increasing the satisfied weight) variable. This often leads to violated hard-constraints (due to flipping of non-assumable variables), and the restarts which are needed for escaping those situations lead to the relatively noisy ascent of CDRW. Other SLS Max-SAT algorithms like HSAT (Gent and Walsh, 1993) avoid downward flips (flips which decrease the currently satisfied weight), quickly increasing the satisfied weight but ultimately get stuck in local optima. A close inspection of Fig. 1 reveals that HSAT oscillates forever short of satisfying all hard constraints.

## 5 MAX-SAT FRAMED AS MBD

In what follows we discuss the use of MBD for solving Max-SAT problems.

### 5.1 An MBD-Based Max-SAT Algorithm

Algorithm 2, called Diorama (DIagnOsis-based algoRithm for mAx-sat optiMizAtion), shows a very simple translation from a Max-SAT problem in CNF to a diagnostic problem.

**Algorithm 2** Diorama: an algorithm for Max-SAT optimization based on MBD

1: **function** Diorama($\Phi$) **returns** a term
  **inputs:** $\Phi$, set of clauses
  **local variables:** DS $= \langle \text{SD}, \text{COMPS}, \text{OBS} \rangle$, diagnostic system
    $c_i$, clause
    $h_i$, variable
2: **for all** $c_i \in \Phi$ **do**
3: SD $\leftarrow$ SD $\wedge \{h_i \Rightarrow c_i\}$
4: COMPS $\leftarrow$ COMPS $\cup\, h_i$
5: **end for**
6: **return** MBD(DS, $\top$)
7: **end function**

The loop in lines 2 - 4 of Alg. 2 modifies each clause in the input problem $\Phi$. Note that line 3 adds exactly one literal to each input clause $c_i$ as, given a clause $c = x_1 \vee x_2 \vee \cdots \vee x_n$, we have $h \Rightarrow (x_1 \vee x_2 \vee \cdots \vee x_n) \equiv \neg h \vee x_1 \vee x_2 \vee \cdots \vee x_n$ and the right-hand side of the last equivalence is also a clause. Line 4 adds a total of $|\Phi|$ assumable variables to $|\text{COMPS}|$ where $|\Phi|$ is the number of clauses in $\Phi$.

Algorithm 2 always creates a system description SD $\in$ **WFM** (cf. Def. 2). Note as well that Alg. 2 invokes the MBD oracle in line 6 with an empty observation (for any propositional formula $\Phi$ we have $\Phi \wedge \top \equiv \Phi$).

In a stricter paper one can formally show the correctness of Diorama, i.e., one can prove that Alg. 2 always computes an optimal Max-SAT solution if it is configured with an MBD oracle that computes at least one cardinality-minimal diagnosis. The complexity of Diorama is dominated by the complexity of the Max-SAT solver. The complexity of Alg. 2 is $O(|\Phi|) + \Psi$ where $\Psi$ is the complexity of the MBD oracle. We will, however, leave this discussion short in order to provide more extensive empirical evidence on the optimality of Diorama.

### 5.2 Experimental Results with a Stochastic MBD Oracle

In our first series of Max-SAT experiments we have configured Alg. 2 with the stochastic MBD oracle Safari (Feldman *et al.*, 2010). Safari is an approximation-based algorithm and we have configured it to compute guaranteed subset-minimal diagnoses (it cannot be configured to compute guaranteed cardinality-minimal diagnoses). These subset-minimal diagnoses are used as an approximation to cardinality-minimal diagnoses. The resulting algorithm Diorama/Safari is similar to SLS-based Max-SAT algorithms like the one discussed in Sec. 4.2.

Table 2: Optimality of UBCSAT algorithms and DIORAMA/SAFARI and the number of steps (in parentheses) in which this optimality has been achieved

| Set Name | RGSAT | Schöening | CDRW | URW | SAMD | IRoTS | RoTS | G2WSAT Novelty+ | G2WSAT | Adaptive Novelty+ | Novelty+ | Novelty | WalkSAT/TABU | WalkSAT | HWSAT | HSAT | GSAT/TABU | GWSAT | GSAT | DIORAMA/SAFARI |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MAX3SAT/40 | 30 | 39 | 39 | 48 | 30 | 30 | 30 | 30 | 30 | 30 | 31 | 31 | 34 | 32 | 31 | 33 | 30 | 30 | 33 | 58 |
| | (801) | (4586) | (4425) | (4515) | (183) | (109) | (108) | (247) | (240) | (1846) | (2852) | (3141) | (4272) | (3699) | (1505) | (29) | (231) | (747) | (32) | (–) |
| MAX3SAT/50 | 26 | 37 | 37 | 48 | 26 | 26 | 26 | 26 | 26 | 26 | 27 | 27 | 31 | 29 | 27 | 29 | 26 | 26 | 29 | 54 |
| | (2192) | (4486) | (4659) | (4546) | (406) | (241) | (255) | (654) | (695) | (2388) | (3374) | (3484) | (3999) | (3892) | (1522) | (47) | (384) | (1513) | (77) | (–) |
| MAX3SAT/60 | 22 | 34 | 34 | 48 | 21 | 21 | 21 | 21 | 21 | 22 | 23 | 23 | 26 | 25 | 23 | 24 | 21 | 21 | 24 | 48 |
| | (2947) | (4667) | (4622) | (4755) | (328) | (289) | (271) | (755) | (867) | (2161) | (3094) | (3209) | (3868) | (3633) | (1696) | (82) | (310) | (1488) | (108) | (–) |
| MAX3SAT/70 | 19 | 32 | 31 | 48 | 18 | 18 | 18 | 18 | 18 | 19 | 20 | 20 | 23 | 22 | 20 | 21 | 18 | 18 | 21 | 43 |
| | (3735) | (4480) | (4591) | (4976) | (847) | (631) | (622) | (1527) | (1697) | (2516) | (3205) | (3093) | (3648) | (3464) | (1604) | (147) | (770) | (2316) | (279) | (–) |
| MAX3SAT/40 | 31 | 39 | 39 | 48 | 31 | 31 | 31 | 31 | 31 | 31 | 32 | 32 | 34 | 33 | 32 | 34 | 31 | 31 | 34 | 58 |
| | (812) | (4411) | (4398) | (4540) | (209) | (130) | (126) | (277) | (296) | (2007) | (2841) | (2819) | (3654) | (3371) | (1386) | (40) | (220) | (829) | (41) | (–) |
| MAX3SAT/60 | 23 | 34 | 34 | 49 | 22 | 22 | 22 | 22 | 22 | 23 | 24 | 24 | 27 | 26 | 24 | 25 | 22 | 22 | 25 | 49 |
| | (2768) | (3962) | (3957) | (4324) | (337) | (234) | (247) | (735) | (803) | (2176) | (3022) | (3178) | (3367) | (3245) | (1259) | (128) | (363) | (1442) | (142) | (–) |
| MAX3SAT/80 | 18 | 31 | 31 | 49 | 17 | 17 | 17 | 17 | 17 | 18 | 19 | 19 | 21 | 21 | 18 | 20 | 17 | 17 | 20 | 40 |
| | (3376) | (3711) | (3722) | (4380) | (534) | (534) | (458) | (1727) | (1689) | (1972) | (2581) | (2769) | (3356) | (3049) | (1427) | (148) | (593) | (2109) | (235) | (–) |
| MAX2SAT/60 | 93 | 110 | 110 | 132 | 93 | 93 | 93 | 93 | 93 | 97 | 102 | 102 | 104 | 102 | 93 | 93 | 93 | 93 | 93 | 143 |
| | (794) | (4754) | (4673) | (4765) | (88) | (63) | (64) | (2204) | (2136) | (3774) | (4348) | (4445) | (4442) | (4346) | (444) | (58) | (95) | (387) | (84) | (–) |
| MAX2SAT/100 | 81 | 105 | 105 | 138 | 80 | 80 | 80 | 82 | 83 | 87 | 96 | 95 | 95 | 95 | 81 | 81 | 80 | 80 | 81 | 131 |
| | (3320) | (4611) | (4659) | (4710) | (246) | (238) | (221) | (2930) | (3080) | (3651) | (4178) | (4237) | (4444) | (4184) | (574) | (233) | (248) | (1227) | (357) | (–) |
| MAX2SAT/140 | 68 | 93 | 93 | 139 | 66 | 66 | 66 | 70 | 70 | 73 | 83 | 83 | 80 | 81 | 66 | 66 | 66 | 66 | 67 | 113 |
| | (4080) | (4404) | (4384) | (4752) | (554) | (522) | (689) | (3056) | (2915) | (3533) | (3975) | (3966) | (3960) | (4001) | (764) | (574) | (542) | (1824) | (672) | (–) |
| SPINGLASS | 64 | 86 | 87 | 113 | 58 | 58 | 58 | 63 | 63 | 66 | 75 | 74 | 70 | 69 | 60 | 60 | 58 | 59 | 60 | 105 |
| | (3628) | (4249) | (4484) | (4492) | (692) | (431) | (792) | (3488) | (3771) | (3368) | (4418) | (4345) | (4100) | (4369) | (914) | (587) | (964) | (2739) | (712) | (–) |
| RAMSEY | 11 | 16 | 16 | 21 | 10 | 9 | 9 | 10 | 10 | 10 | 11 | 11 | 14 | 14 | 10 | 11 | 10 | 9 | 11 | 673 |
| | (1243) | (1721) | (1625) | (2596) | (189) | (368) | (261) | (818) | (724) | (672) | (1063) | (1132) | (1115) | (1214) | (657) | (40) | (198) | (687) | (45) | (–) |
| DIMACS | 233 | 248 | 248 | 252 | 233 | 233 | 233 | 233 | 233 | 238 | 242 | 242 | 243 | 242 | 234 | 235 | 233 | 233 | 235 | 267 |
| | (601) | (3878) | (3975) | (4153) | (89) | (59) | (67) | (391) | (367) | (3365) | (3885) | (3680) | (3974) | (3785) | (863) | (22) | (111) | (510) | (25) | (–) |
| RANDOM | 182 | 212 | 212 | 223 | 182 | 182 | 182 | 182 | 182 | 193 | 201 | 201 | 202 | 202 | 185 | 188 | 182 | 182 | 187 | 239 |
| | (2459) | (4710) | (4896) | (4534) | (303) | (205) | (204) | (2638) | (2698) | (4299) | (4544) | (4818) | (4718) | (4865) | (1298) | (33) | (344) | (1931) | (37) | (–) |
| SPINGLASS | 113 | 156 | 155 | 199 | 98 | 96 | 97 | 112 | 112 | 116 | 136 | 136 | 118 | 126 | 99 | 99 | 98 | 99 | 100 | 172 |
| | (4558) | (4008) | (4370) | (4887) | (899) | (1981) | (2289) | (3246) | (3724) | (3673) | (3931) | (3615) | (3556) | (4543) | (1826) | (1153) | (1052) | (3553) | (1202) | (–) |

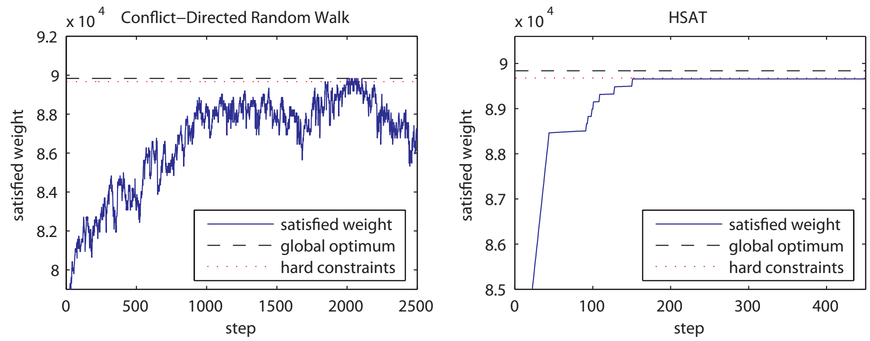


Figure 1: Progress of two SLS Max-SAT algorithms in a weak-fault model of c432, single fault observation

Table 3: Optimality of the UBC SLS-based Max-SAT algorithms and SAFARI/DIORAMA on small industrial Max-SAT 2009 instances

| | c3 | c5315 | c6288 | c7552 | mot_comb1 | mot_comb2 | mot_comb3 | s15850 |
|---|---|---|---|---|---|---|---|---|
| RGSAT | 1 215 | 526 | 440 | 647 | 531 | 1362 | 1 758 | 3 483 |
| Schöening | 1 556 | 146 | 410 | 262 | 1 | 805 | 2 534 | 6 055 |
| CDRW | 1 532 | 145 | 423 | 255 | 2 | 832 | 2 533 | 6 012 |
| URW | 4 936 | 1 098 | 1 939 | 1 520 | 1 432 | 3 654 | 6 838 | 12 093 |
| SAMD | 342 | 500 | 132 | 695 | 9 | 86 | 598 | 2 175 |
| IRoTS | 261 | 78 | 86 | 86 | 24 | 83 | 436 | 1 972 |
| RoTS | 347 | 129 | 130 | 122 | 5 | 108 | 565 | 2 170 |
| G2WSAT Novelty$^+$ | 237 | 13 | 99 | 54 | 1 | 4 | 544 | 2 124 |
| G2WSAT | 206 | 16 | 101 | 60 | 1 | 3 | 484 | 2 084 |
| Adaptive Novelty$^+$ | 238 | 38 | 106 | 64 | 1 | 184 | 522 | 2 504 |
| Novelty$^+$ | 339 | 13 | 111 | 66 | 1 | 49 | 800 | 3 367 |
| Novelty | 314 | 16 | 121 | 63 | 1 | 46 | 769 | 3 335 |
| WalkSAT/TABU | 368 | 24 | 129 | 84 | 1 | 676 | 810 | 2 819 |
| WalkSAT | 555 | 21 | 177 | 94 | 1 | 27 | 1 088 | 3 281 |
| HWSAT | 351 | 417 | 123 | 547 | 4 | 66 | 579 | 2 187 |
| HSAT | 354 | 498 | 130 | 798 | 18 | 116 | 576 | 2 145 |
| GSAT/TABU | 354 | 98 | 126 | 98 | 7 | 147 | 589 | 2 144 |
| GWSAT | 446 | 371 | 117 | 474 | 1 | 229 | 839 | 2 629 |
| GSAT | 372 | 392 | 130 | 347 | 18 | 149 | 596 | 2 191 |
| DIORAMA/SAFARI | 1 | 2 | 3 | 1 | 2 | 2 | 2 | 1 |

Table 2 compares the optimality of DIORAMA/SAFARI to the algorithms from the UBCSAT suite. The experiments are on the problems from the Second Max-SAT Evaluation 2007. The majority of those problems (680 out of a total of 815) are random 2-SAT[1] and 3-SAT. We have configured UBCSAT to terminate after 100 000 steps and we have run it 10 times for each experiment. We can see in Table 2 that the optimality of DIORAMA/SAFARI is slightly worse but comparable to the optimality of the UBCSAT algorithms. In general, the optimality, of all UBCSAT algorithms and DIORAMA/SAFARI is similar which means that there are either (1) continuous diagnostic subspaces in the Max-SAT instances [2] or (2) the Max-SAT algorithms and DIORAMA/SAFARI cannot climb after the initial variable assignment.

Table 3 shows the optimality of the UBCSAT Max-SAT algorithms and DIORAMA/SAFARI on the eight smallest instances of the Max-SAT 2009 industrial benchmark. The c3, c5315, c6288, c7552, mot_comb1, mot_comb2, mot_comb3, and s15850 columns in Table 3 correspond to the c3_DD_s3_f1_e1_v1-bug-onevec-gate-0, c5315-bug-gate-0, c6288-bug-gate-0, c7552-bug-gate-0, mot_comb1._red-gate-0, mot_comb2._red-gate-0, mot_comb3._red-gate-0, and s15850-bug-onevec-gate-0 instances in the Max-SAT benchmark. We can see that DIORAMA/SAFARI outperforms the traditional SLS-based algorithms by two to three orders-of-magnitude. This is not surprising as the c5315, c6288, c7552 instances come from the ISCAS85 benchmark and we have seen the

[1]Recall that although the 2-SAT decision problem is easy, the optimization Max-2-SAT problem is already *NP*-hard.

[2]See (Feldman *et al.*, 2010) for defining continuity in MBD.

good performance of SAFARI on these instances in Sec. 4. What is more interesting is that these results hold for other benchmark instances from formal verification. s15850, for example, comes from ISCAS89 and has 534 D-type flip-flops. Note that all these problem instances result in solutions of very small cardinality.

## 6 CONCLUSION

This paper offers extensive empirical research on the use of consistency-based diagnosis for solving Max-SAT problems and vice-versa. The main contribution of this paper is solving more than 800 Max-SAT instances with DIORAMA/SAFARI and UBCSAT and more than 700 74XXX/ISCAS85 problems with MERIDIAN/UBCSAT and SAFARI. We have experimented with 20 algorithms for SLS-based Max-SAT. The good result of DIORAMA/SAFARI on small industrial instances show that many Max-SAT problems of real-world importance can be optimally and efficiently solved with greedy stochastic algorithms.

## REFERENCES

(Bailey and Stuckey, 2005) James Bailey and Peter J. Stuckey. Discovery of minimal unsatisfiable subsets of constraints using hitting set dualization. In *Proc. PADL'05*, pages 174–186, 2005.

(Castellini *et al.*, 2003) Claudio Castellini, Enrico Giunchiglia, and Armando Tacchella. SAT-based planning in complex domains: Concurrency, constraints and nondeterminism. *Artificial Intelligence*, 147(1-2):85–117, 2003.

(Chen *et al.*, 2009) Yibin Chen, Sean Safarpour, Andreas G. Veneris, and João P. Marques Silva. Spatial and temporal design debug using partial MaxSAT. In *ACM Great Lakes Symposium on VLSI*, pages 345–350, 2009.

(de Givry *et al.*, 2003) Simon de Givry, Javier Larrosa, Pedro Meseguer, and Thomas Schiex. Solving Max-SAT as weighted CSP. In *CP*, pages 363–376, 2003.

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

(de Kleer *et al.*, 1992) Johan de Kleer, Alan Mackworth, and Raymond Reiter. Characterizing diagnoses and systems. *Artificial Intelligence*, 56(2-3):197–222, 1992.

(de Kleer, 1990) Johan de Kleer. Using crude probability estimates to guide diagnosis. *Artificial Intelligence*, 45(3):381–291, 1990.

(Feldman *et al.*, 2009) Alexander Feldman, Gregory Provan, and Arjan van Gemund. Solving strong-fault diagnostic models by model relaxation. In *Proc. IJCAI'09*, 2009.

(Feldman *et al.*, 2010) Alexander Feldman, Gregory Provan, and Arjan van Gemund. Approximate model-based diagnosis using greedy stochastic search. *JAIR*, 2010.

(Fu and Malik, 2006) Zhaohui Fu and Sharad Malik. On solving the partial max-sat problem. In *SAT*, pages 252–265, 2006.

(Gent and Walsh, 1993) Ian P. Gent and Toby Walsh. Towards an understanding of hill-climbing procedures for SAT. In *Proc. AAAI'93*, pages 28–33, 1993.

(Giunchiglia and Maratea, 2006) Enrico Giunchiglia and Marco Maratea. Solving optimization problems with DLL. In *Proc. ECAI'06*, pages 377–381, 2006.

(Gomes *et al.*, 2007) Carla P. Gomes, Henry Kautz, Ashish Sabharwal, and Bart Selman. Satisfiability solvers. *Handbook of Knowledge Representation*, 2007.

(Hoos and Stützle, 2004) Holger Hoos and Thomas Stützle. *Stochastic Local Search: Foundations and Applications*. Morgan Kaufmann Publishers Inc., 2004.

(Hutter *et al.*, 2002) Frank Hutter, Dave A. D. Tompkins, and Holger H. Hoos. Scaling and probabilistic smoothing: Efficient dynamic local search for SAT. In *Proc. CP'02*, pages 233–248, 2002.

(Iyer *et al.*, 2003) Madhu K. Iyer, Ganapathy Parthasarathy, and Kwang Ting Ting Cheng. SATORI - a fast sequential SAT engine for circuits. In *Proc. ICCAD'03*, pages 320–325, 2003.

(Kutsuna *et al.*, 2009) Takuro Kutsuna, Shuichi Sato, and Naoya Chujo. Diagnosing automotive control systems using abstract model-based diagnosis. In *Proc. DX'09*, pages 99–105, 2009.

(Liffiton and Sakallah, 2005) Mark H. Liffiton and Karem A. Sakallah. On finding all minimally unsatisfiable subformulas. In *Proc. SAT'05*, pages 173–186, 2005.

(Papadimitriou, 1991) Christos Papadimitriou. On selecting a satisfying truth assignment. In *Proc. FOCS'91*, pages 163–169, 1991.

(Sang *et al.*, 2007) Tian Sang, Paul Beame, and Henry A. Kautz. A dynamic approach for MPE and weighted MAX-SAT. In *Proc. IJCAI'07*, pages 173–179, 2007.

(Tompkins and Hoos, 2005) Dave A. D. Tompkins and Holger H. Hoos. UBCSAT: An implementation and experimentation environment for SLS algorithms for SAT and MAX-SAT. In *Proc. SAT'04*, pages 306–320, 2005.

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