Delft University of Technology
Embedded Software Report Series

A Family of Model-Based Diagnosis Algorithms Based on Max-SAT

Alexander Feldman a.b.feldman@tudelft.nl
Gregory Provan g.provan@cs.ucc.ie
Arjan van Gemund a.j.c.vangemund@tudelft.nl

Report Number: ES-2009-02
ISSN: 1877-7805

embedded software

TU Delft Delft University of Technology

Published and produced by:
Embedded Software Section
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology
Mekelweg 4
2628 CD Delft
The Netherlands

Information about the Embedded Software Report Series:
info-es-ewi@tudelft.nl
Information about the Embedded Software Section:
http://www.es.ewi.tudelft.nl/

A Family of Model-Based Diagnosis Algorithms Based on Max-SAT

Alexander Feldman A.B.FELDMAN@TUDELFT.NL
Delft University of Technology,
Faculty of Electrical Engineering, Mathematics and Computer Science,
Mekelweg 4, 2628 CD, Delft, The Netherlands

Gregory Provan G.PROVAN@CS.UCC.IE
University College Cork, Department of Computer Science,
College Road, Cork, Ireland

Arjan van Gemund A.J.C.VANGEMUND@TUDELFT.NL
Delft University of Technology,
Faculty of Electrical Engineering, Mathematics and Computer Science,
Mekelweg 4, 2628 CD, Delft, The Netherlands

Abstract

Max-SAT is an extensively studied optimization problem and variations of Max-SAT (like Partial Max-SAT and Weighted Max-SAT) can be used for computing minimal diagnoses. In this report we (1) describe a family of algorithms for computing minimal diagnoses based on Max-SAT and (2) experimentally evaluate the performance and optimality of these algorithms on a benchmark consisting of 74XXX/ISCAS85 combinational circuits. For the experiments we use both complete (partial and weighted) and SLS-based Max-SAT. We have established that the performance of complete Max-SAT and the optimality of SLS-based Max-SAT degrade when increasing the circuit size or the cardinality of the faults.

1. Introduction

Given a formula Φ in Conjunctive Normal Form (CNF), a Max-SAT (Hoos & Stützle, 2004) solution is a variable assignment that maximizes the number of satisfied clauses in Φ (in most cases of interest Φ is unsatisfiable, otherwise any variable assignment which satisfies Φ is also a Max-SAT solution). In partial Max-SAT some of the clauses in Φ 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 Φ and a solution maximizes the sum of the weights of the satisfied clauses.

In Model-Based Diagnosis (MBD), a diagnosis ω satisfies a system description SD and an observation α (SD $\wedge\, \alpha$ is in CNF) such that SD $\wedge\, \alpha \wedge \omega \not\models \perp$ (Feldman, Provan, & van Gemund, 2009). A minimal diagnosis is a diagnosis which minimizes some criterion, for example, the number of negative literals in ω. It turns out that it is very easy to use Max-SAT for computing minimal diagnoses, something we study in this technical report.

In this technical report we (1) show how to use Max-SAT for computing minimal diagnoses, (2) study the performance of a family of Max-SAT-based diagnostic algorithms, and (3) empirically compare the performance and optimality of the Max-SAT-based algo-

rithms to a specialized MBD algorithm for computing diagnoses named Safari (Feldman et al., 2009). Our experiments are on the 74XXX/ISCAS85 benchmark of combinational circuits. The results show that although diagnosis based on Max-SAT performs better than other algorithms for computing minimal diagnosis, specialized stochastic solvers like Safari outperform Max-SAT at least an order-of-magnitude for the class of diagnostic problems we have considered. This is not surprising as specialized MBD algorithms exploit specific properties of the search space.

This technical report is organized as follows. Section 3 describes a Max-SAT based algorithm for MBD. Section 4 describes experimental results with complete Max-SAT algorithms. Section 5 describes experimental results with SLS-based Max-SAT algorithms.

2. Technical Background, Notation, and Problem Definition

This technical report uses the theory, notation, problem definition, and experimental setup of a paper describing an alternative approach to computing minimal diagnoses (Feldman et al., 2009).

3. A Max-SAT-Based MBD Algorithm

We have used the approach of Sang et al. (Sang, Beame, & Kautz, 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 technical report can be translated to multiple optimization problems which can be solved with SAT-based methods (Giunchiglia & Maratea, 2006). The Maximum Satisfiable Subset (MSS) problem, for example, is dual to the Minimal Unsatisfiable Subset problem (Bailey & Stuckey, 2005) and the two can be solved with Max-SAT and Min-UNSAT solvers, respectively (Liffiton & 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.

We next describe Alg. 1. 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) as 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.

Algorithm 1 Weighted Max-SAT based diagnosis algorithm.

```
function MaxSATDiagnose(DS, α) returns a set of diagnoses
    inputs: DS = ⟨SD, COMPS, OBS⟩, diagnostic system
            α, term, observation
    local variables: W, set of clauses; Ω, set of diagnoses
                     ω, diagnosis term; c, clause; h, variable
    for all c ∈ Clauses(SD) do
        W ← W ∪ ⟨∞, c⟩
    end for
    for all h ∈ COMPS do
        W ← W ∪ ⟨1, c⟩
    end for
    while ω ← Max-SAT(W) do
        W ← W ∪ ⟨∞, ¬ω⟩
        Ω ← Ω ∪ ω
    end while
    return Ω
end function
```

4. Experimental Results with Complete Max-SAT

We next show the performance of Alg. 1 with two partial Max-SAT algorithms (partial and weighted Max-SAT algorithms are always complete).

Table 1: Performance of W-MaxSatz and MiniMaxSat [% of tests solved]

	W-MaxSatz			MiniMaxSat		
Name	Weak	S-A-0	S-A-1	Weak	S-A-0	S-A-1
74182	100	100	100	100	100	100
74L85	100	100	100	100	100	100
74283	100	100	100	100	100	100
74181	100	100	100	100	100	100
c432	99.7	100	100	100	100	100
c499	2	100	65.5	99.9	43.8	100
c880	0	9.2	34.9	95.5	100	99.7
c1355	0	0	0	62	17.6	50.1
c1908	0	0	0	36.5	0.5	0
c2670	0	3.7	0	74.1	0	0
c3540	0	0	0	0.1	0	0
c5315	0	0	0	0.2	0	0
c6288	0	0	0	0	0	0
c7552	0	0	0	0	0	0

Table 1 shows performance results of the partial Max-SAT solver W-MaxSatz (Argelich, Li, & Manyá, 2007) and the weighted MiniMaxSat (Heras, Larrosa, & Oliveras, 2008). Note that the performance of W-MaxSatz in weighted mode is slightly worse than when solving the respective partial formulae. Both solvers are state-of-the-art and have been submitted to the Second Max-SAT Evaluation 2007.[1] The performance of the complete Max-SAT solvers degrades with increasing circuit size and the cardinality of the injected faults. The results of those algorithms are comparable to, or slightly better, than the one of CDA* and HA*. For comparison, Safari solves all test instances.

Our findings are consistent with the ones of Giunchiglia and Marathea (2006). In their paper Giunchiglia and Marathea report ISCAS85 times similar to the ones measured with state-of-the-art Max-SAT solvers where no solver computes a solution for a circuit larger than c3540. These experiments consider the plain ISCAS85 combinational circuits (i.e., their models do not have assumable variables) and it has been discussed by the same authors that adding assumable variables (or in their terminology increasing the number of preferences) increases the problem difficulty.

5. Experimental Results with SLS Max-SAT

A diagnostic algorithm based on SLS Max-SAT would be the best candidate for comparison to Safari due to the stochastic nature of the latter. Unfortunately, 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.

The two reasons above prevent us from using a diagnostic reasoner based on SLS Max-SAT in practical applications. Despite that, we have conducted extensive experimentation with UBCSAT (Tompkins & Hoos, 2005) 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. 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.

1. http://www.maxsat07.udl.es/

The resulting optimality of algorithms based on SLS Max-SAT in computing single fault diagnoses is shown in Table 2. We have run experiments with all algorithms or algorithm variants implemented by the UBCSAT suite.

The data in Table 2 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 2).

Table 2 shows the generally bad optimality 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, Tompkins, & Hoos, 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 & 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.

6. Conclusion

We have shown a family of Max-SAT-based algorithms for computing minimal diagnoses. Depending on the type of the Max-SAT oracle, these algorithms compute minimal-cardinality diagnoses (by using partial Max-SAT) or probability-minimal diagnoses (by using weighted Max-SAT). We have also discussed the implications of using an SLS-based Max-SAT algorithm and studied the optimality of the computed diagnoses with a number of SLS-based Max-SAT algorithms.

We have experimented with the 74XXX/ISCAS85 combinational circuits and a number of observation vectors. The results showed that although Max-SAT could compute diagnoses in many of the cases, the performance of Max-SAT degraded when increasing the circuit size or the cardinality of the injected faults.

Table 2: Optimality of SLS-based Max-SAT algorithms and Safari (**WFM**)

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.

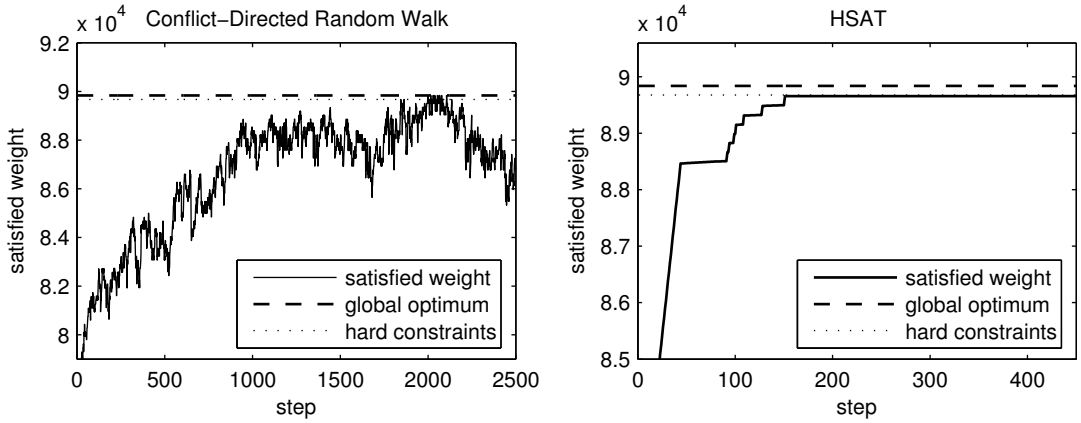

Figure 1: Progress of two SLS Max-SAT algorithms in a weak-fault model of c432, single fault observation

References

Argelich, J., Li, C. M., & Manyá, F. (2007). An improved exact solver for partial Max-SAT. In *Proc. NCP'07*, pp. 230–231.

Bailey, J., & Stuckey, P. J. (2005). Discovery of minimal unsatisfiable subsets of constraints using hitting set dualization. In *Proc. PADL'05*, pp. 174–186.

Feldman, A., Provan, G., & van Gemund, A. (2009). Approximate model-based diagnosis using greedy stochastic search. Submitted for review to JAIR.

Gent, I. P., & Walsh, T. (1993). Towards an understanding of hill-climbing procedures for SAT. In *Proc. AAAI'93*, pp. 28–33.

Giunchiglia, E., & Maratea, M. (2006). Solving optimization problems with dll. In *Proc. ECAI'06*, pp. 377–381.

Heras, F., Larrosa, J., & Oliveras, A. (2008). MiniMaxSAT: An efficient weighted Max-SAT solver. *Journal of Artificial Intelligence Research*, *31*, 1–32.

Hoos, H., & Stützle, T. (2004). *Stochastic Local Search: Foundations and Applications*. Morgan Kaufmann Publishers Inc.

Hutter, F., Tompkins, D. A. D., & Hoos, H. H. (2002). Scaling and probabilistic smoothing: Efficient dynamic local search for sat. In *Proc. CP'02*, pp. 233–248.

Liffiton, M. H., & Sakallah, K. A. (2005). On finding all minimally unsatisfiable subformulas. In *Proc. SAT'05*, pp. 173–186.

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

Sang, T., Beame, P., & Kautz, H. A. (2007). A dynamic approach for MPE and weighted MAX-SAT. In *Proc. IJCAI'07*, pp. 173–179.

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