Computing Minimal Diagnoses by Greedy Stochastic Search

Alexander Feldman
Delft University of Technology
Mekelweg 4, 2628 CD, Delft
The Netherlands
Tel.: +31 15 2781935
email: a.b.feldman@tudelft.nl

Gregory Provan
University College Cork
College Road, Cork, Ireland
Tel: +353 21 4901816
email: g.provan@cs.ucc.ie

Arjan van Gemund
Delft University of Technology
Mekelweg 4, 2628 CD, Delft
The Netherlands
Tel.: +31 15 2781935
email: a.j.c.vangemund@tudelft.nl

Abstract

Most algorithms for computing diagnoses within a model-based diagnosis framework are deterministic. Such algorithms guarantee soundness and completeness, but are $\Sigma_2^{\mathbf{P}}$-hard. To overcome this complexity problem, which prohibits the computation of high-cardinality diagnoses for large systems, we propose a novel *approximation approach* for multiple-fault diagnosis, based on a greedy stochastic algorithm called SAFARI (StochAstic Fault diagnosis AlgoRIthm). We prove that SAFARI can be configured to compute diagnoses which are of guaranteed minimality under subsumption. We analytically model SAFARI search as a Markov chain, and show a probabilistic bound on the minimality of its minimal diagnosis approximations. We have applied this algorithm to the 74XXX and ISCAS85 suites of benchmark combinatorial circuits, demonstrating order-of-magnitude speedups over two state-of-the-art deterministic algorithms, CDA* and HA*, for multiple-fault diagnoses.

Introduction

Model-Based Diagnosis (MBD) is an area of abductive inference that uses a system model, together with observations about system behavior, to isolate sets of faulty components (diagnoses) that explain the observed behavior, according to some minimality criterion. The standard MBD formalization (Reiter 1987) frames a diagnostic problem in terms of a set of logical clauses that include mode-variables describing the nominal and fault status of system components; from this the diagnostic status of the system can be computed given an observation (OBS) of the system's sensors. MBD provides a sound and complete approach to enumerating multiple-fault diagnoses, and exact algorithms can guarantee finding a diagnosis optimal with respect to the number of faulty components, probabilistic likelihood, etc.

The biggest challenge (and impediment to industrial deployment) is the computational complexity of the MBD problem. The MBD problem of isolating multiple-fault diagnoses is known to be $\Sigma_2^{\mathbf{P}}$-complete for arbitrary propositional models (Eiter and Gottlob 1995), and $\Sigma_1^{\mathbf{P}}$-complete for Horn propositional models (Bylander et al. 1991; Friedrich, Gottlob, and Nejdl 1990). Since almost all proposed MBD algorithms have been complete and exact, with some authors proposing possible trade-offs between completeness and faster consistency checking (Williams and Ragno 2007) by employing methods such as Binary Constraint Propagation (BCP), the complexity problem still remains a major challenge to MBD.

To overcome this complexity problem, we propose a novel *approximation approach* for multiple-fault diagnosis, based on a stochastic algorithm. SAFARI (StochAstic Fault diagnosis AlgoRIthm) sacrifices guarantees of optimality, but for diagnostic systems in which faults are described in terms of an arbitrary deviation from nominal behavior SAFARI can compute diagnoses several orders of magnitude faster than competing algorithms.

Our contributions are as follows. (1) This paper introduces an approximation algorithm for computing diagnoses within an MBD framework, based on a greedy stochastic algorithm. (2) We show that we can compute minimal-cardinality diagnoses for weak fault models in poly-time (calling the incomplete SAT-solver BCP), and that more general frameworks are also amenable to this class of algorithm. (3) We model SAFARI search as a Markov chain to show the performance and optimality trade-offs that the algorithm makes. (4) We apply this algorithm to a suite of benchmark combinatorial circuits, demonstrating order-of-magnitude speedup over two state-of-the-art deterministic algorithms, CDA* and HA*, for multiple-fault diagnoses. Moreover, whereas the search complexity for the deterministic algorithms tested increases exponentially with fault cardinality, the search complexity for this stochastic algorithm appears to be independent of fault cardinality. SAFARI is of great practical significance, as it can compute a large fraction of cardinality-minimal diagnoses for discrete systems too large or complex to be diagnosed by existing deterministic algorithms.

Related Work

MBD inference can be viewed as constraint optimization, with particular constraints over failure variables, as we will describe. MBD has developed algorithms to exploit these domain properties, and our proposed approach differs significantly with almost all MBD algorithms that appear in the literature. While most advanced MBD algorithms make use of preferences, e.g., fault-mode probabilities, to improve search efficiency, the algorithms themselves are determinis-

tic, and use the preferences to identify the most-preferred solutions. This contrasts with stochastic SAT algorithms, which rather than backtracking may randomly flip variable assignments to determine a satisfying assignment.

At first glance, it seems like MBD could be efficiently solved using an encoding as a SAT (Jin, Han, and Somenzi 2005), constraint satisfaction (Freuder et al. 1995) or Bayesian network (Kask and Dechter 1999) problem. However, one needs to take into account the increase in formula size (over a direct MBD encoding), in addition to the underconstrained nature of MBD problems. The most closely-related SAT encoding is that of Vatan et al. (Vatan et al. 2003), who map the diagnosis problem into the monotone SAT problem, and then propose to use efficient SAT algorithms for computing diagnoses. The approach of Vatan et al. has shown speedups in comparison with other diagnosis algorithms; the main drawback is the number of extra variables and clauses that must be added in the SAT encoding, which is even more significant for strong fault models and multi-valued variables. In contrast, our approach works directly on the given diagnosis model and requires no conversion to another representation. In the experimental analysis section, we show that a straightforward ALLSAT encoding (Jin, Han, and Somenzi 2005) compares very unfavourably with SAFARI, since standard SAT inference does not exploit particular MBD domain properties.

Technical Background

Our discussion continues by formalizing some MBD notions. This paper uses the traditional diagnostic definitions (de Kleer and Williams 1987), except that we use propositional logic terms (conjunctions of literals) instead of sets of failing components.

Central to MBD, a *model* of an artifact is represented as a propositional **Wff** over some set of variables. Discerning two subsets of these variables as *assumable* and *observable*[1] variables gives us a diagnostic system.

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

Throughout this paper we assume that $\mathrm{OBS} \cap \mathrm{COMPS} = \emptyset$ and $\mathrm{SD} \not\models \bot$. Although SAFARI delivers good results for a larger class of diagnostic models, this paper focuses on the well-known weak-fault models.

Definition 2 (Weak-Fault Model)**.** A diagnostic system $\mathrm{DS} = \langle \mathrm{SD}, \mathrm{COMPS}, \mathrm{OBS} \rangle$ belongs to the class **WFM** iff SD is in the form $(h_1 \Rightarrow F_1) \wedge \ldots \wedge (h_n \Rightarrow F_n)$ such that $1 \leq i, j \leq n$, $\{h_i\} \subseteq \mathrm{COMPS}$, $F_j \in$ **Wff**, and none of h_i appears in F_j.

Note the conventional selection of the sign of the "health" variables $h_1, h_2, \ldots h_n$. Other authors use "ab" for abnormal or "ok" for healthy. Weak-fault models are sometimes referred to as models with *ignorance of abnormal behavior* (de Kleer, Mackworth, and Reiter 1992), or *implicit fault systems*. The traditional query in MBD computes terms of assumable variables which are explanations for the system description and an observation.

[1] In the MBD literature the assumable variables are also referred to as "component", "failure-mode", or "health" variables. Observable variables are also called "measurable", or "control" variables.

A Running Example

We will use the Boolean circuit shown in Fig. 1 as a running example for illustrating all the notions and algorithms in this paper. The subtractor, shown there, consists of seven components: an inverter, two or-gates, two xor-gates, and two and-gates. The expression $h \Rightarrow (o \Leftrightarrow \neg i)$ models the normative (healthy) behavior of an inverter, where the variables i, o, and h represent input, output and health respectively. Similarly, an and-gate is modeled as $h \Rightarrow (o \Leftrightarrow i_1 \wedge i_2)$ and an or-gate by $h \Rightarrow (o \Leftrightarrow i_1 \vee i_2)$. Finally, an xor-gate is specified as $h \Rightarrow [o \Leftrightarrow \neg (i_1 \Leftrightarrow i_2)]$.

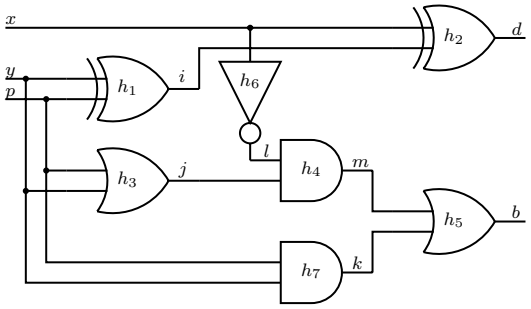

Figure 1: A subtractor circuit

The above propositional formulae are copied for each gate in Fig. 1 and their variables renamed in such a way as to properly connect the circuit and disambiguate the assumables, thus obtaining a propositional formula for the Boolean subtractor, given by:

$$\mathrm{SD} = \begin{cases} h_1 \Rightarrow [i \Leftrightarrow \neg (y \Leftrightarrow p)] \\ h_2 \Rightarrow [d \Leftrightarrow \neg (x \Leftrightarrow i)] \\ h_3 \Rightarrow (j \Leftrightarrow y \vee p) \\ h_4 \Rightarrow (m \Leftrightarrow l \wedge j) \\ h_5 \Rightarrow (b \Leftrightarrow m \vee k) \\ h_6 \Rightarrow (x \Leftrightarrow \neg l) \\ h_7 \Rightarrow (k \Leftrightarrow y \wedge p) \end{cases} \quad (1)$$

The set of assumables is $\mathrm{COMPS} = \{h_1, h_2, \ldots, h_7\}$ and the set of observable variables is $\mathrm{OBS} = \{x, y, p, d, b\}$.

Diagnosis and Minimal Diagnosis

The traditional query in MBD computes terms of assumable variables which are explanations for the system description and an observation.

Definition 3 (Health Assignment)**.** Given a diagnostic system $\mathrm{DS} = \langle \mathrm{SD}, \mathrm{COMPS}, \mathrm{OBS} \rangle$, an assignment HA to all variables in COMPS is defined as a health assignment.

A health assignment HA is a conjunction of propositional literals. In some cases it is convenient to use the set of negative or positive literals in HA. These two sets are denoted as $Lit^{-}(\mathrm{HA})$ and $Lit^{+}(\mathrm{HA})$, respectively.

In our example, the "all nominal" assignment is $\mathrm{HA}_1 = h_1 \wedge h_2 \wedge \ldots \wedge h_7$. The health assignment $\mathrm{HA}_2 = h_1 \wedge h_2 \wedge h_3 \wedge \neg h_4 \wedge h_5 \wedge h_6 \wedge \neg h_7$ means that the two and-gates from Fig. 1 are malfunctioning. What follows is a formal definition of consistency-based diagnosis.

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

Traditionally, other authors (de Kleer and Williams 1987) arrive at minimal diagnosis by computing a minimal hitting set of the minimal conflicts (broadly, minimal health assignments incompatible with the system description and the observation), while this paper makes no use of conflicts, hence the equivalent direct definition above.

There is a total of 96 possible diagnoses given SD and an observation $\alpha_1 = x \wedge y \wedge p \wedge b \wedge \neg d$. Example diagnoses are $\omega_1 = \neg h_1 \wedge h_2 \wedge \ldots \wedge h_7$ and $\omega_2 = h_1 \wedge \neg h_2 \wedge h_3 \wedge \ldots \wedge h_7$. Trivially, given a weak-fault model, the "all faulty" health assignment (in our example $\mathrm{HA}_3 = \neg h_1 \wedge \neg h_2 \wedge \ldots \wedge \neg h_7$) is a diagnosis for any instantiation of the observable variables in OBS (cf. Def. 2).

In the MBD literature, a range of types of "preferred" diagnosis has been proposed. This turns the MBD problem into an optimization problem. In the following definition we consider the common subset-ordering.

Definition 5 (Minimal Diagnosis)**.** A diagnosis ω is minimal if no diagnosis ω' exists such that $Lit^{-}(\omega') \subset Lit^{-}(\omega)$.

For the weak-fault model SD of the circuit shown in Fig. 1 and an observation $\alpha_2 = \neg x \wedge y \wedge p \wedge \neg b \wedge d$ there are 8 minimal and 61 non-minimal diagnoses. In this example, two of the minimal diagnoses are $\omega_3 = \neg h_1 \wedge h_2 \wedge h_3 \wedge h_4 \wedge \neg h_5 \wedge h_6 \wedge h_7$ and $\omega_4 = \neg h_1 \wedge h_2 \wedge \ldots \wedge h_5 \wedge \neg h_6 \wedge \neg h_7$. The diagnosis $\omega_5 = \neg h_1 \wedge \neg h_2 \wedge h_3 \wedge h_4 \wedge \neg h_5 \wedge h_6 \wedge h_7$ is non-minimal as the negative literals in ω_3 form a subset of the negative literals in ω_5.

The set of all minimal diagnoses characterizes all diagnoses given a weak-fault model, but that does not hold in general (de Kleer, Mackworth, and Reiter 1992). With no restrictions on the model, faulty components may "exonerate" each other, resulting in a health assignment containing a proper superset of the negative literals of another diagnosis not to be a diagnosis.

Diagnosis cardinality gives us another partial ordering: a diagnosis is defined as *minimal cardinality* iff it minimizes the number of negative literals.

Definition 6 (Cardinality of a Diagnosis)**.** The cardinality of a diagnosis, denoted as $|\omega|$, is defined as the number of negative literals in ω.

A minimal cardinality diagnosis is a minimal diagnosis, but the opposite does not hold. There are minimal diagnoses which are not minimal cardinality diagnoses.

Stochastic MBD Algorithm

In this section we discuss an algorithm for computing multiple-fault diagnoses using stochastic search.

A Simple Example (Continued)

We will now show a two-step diagnostic process. Step 1 involves randomly choosing candidates. Step 2 attempts to minimize the fault cardinality in these candidates.

In step 1, the stochastic diagnostic search for the subtractor example will start from a random quintuple candidate[2]. In this particular version of our algorithm, once a component is marked as healthy, it cannot be changed back to faulty. To compensate for that, we perform multiple restarts from a random candidate. In our subtractor example and for $\alpha_3 = x \wedge y \wedge p \wedge \neg d \wedge \neg b$, if $h_1 \wedge h_2$ is in an initial "guessed" candidate, it will prove inconsistent with $\mathrm{SD} \wedge \alpha_3$ and another quintuple fault candidate will be guessed.

Assume that this second candidate is $\omega_6 = \neg h_1 \wedge \neg h_2 \wedge h_3 \wedge \neg h_4 \wedge \neg h_5 \wedge h_6 \wedge \neg h_7$. Clearly, $\mathrm{SD} \wedge \alpha_3 \wedge \omega_6 \not\models \bot$. The search algorithm may next try to improve the diagnosis by "flipping" the sign of h_7. The candidate $\omega_7 = \neg h_1 \wedge \neg h_2 \wedge h_3 \wedge \neg h_4 \wedge \neg h_5 \wedge h_6 \wedge h_7$ is a valid quadruple fault diagnosis and it can be improved twice more by "flipping" h_2 and h_5. This gives us the final double-fault $\omega_8 = \neg h_1 \wedge h_2 \wedge h_3 \wedge \neg h_4 \wedge h_5 \wedge h_6 \wedge h_7$. The actual algorithm is somewhat more involved as during the variable flipping it is normal to find inconsistencies. Instead of restarting, it will simply discard these inconsistent candidates until some termination criterion is satisfied.

Intuitively, from our example, due to the large number of double fault diagnoses explaining the same observation, it is not difficult to randomly guess sequences of variables which need to be false in order to explain the observation.

A Greedy Stochastic Algorithm

A number of utility functions are used in the pseudocode listed in this paper. The IMPROVEDIAGNOSIS subroutine takes a term as an argument and changes the sign of a random negative literal. If there are no negative literals, the function returns the original argument. The implementation of RANDOMDIAGNOSIS uses a modified DPLL solver (Davis, Logemann, and Loveland 1962) returning a random SAT solution of $\mathrm{SD} \wedge \alpha$ (cf. the experimental results section for implementation details).

Similar to deterministic methods for MBD, SAFARI uses a SAT-based procedure for checking the consistency of $\mathrm{SD} \wedge \alpha \wedge \omega$. Because $\mathrm{SD} \wedge \alpha$ does not change during the search, the incremental nature of the Logic-Based Truth Maintenance System (LTMS) assumption checking (McAllester 1990) greatly improves the search efficiency. The implementation of SAFARI combines a BCP-based LTMS to check for inconsistencies. If a candidate is consistent, a subsequent DPLL-based check is invoked for completeness.

The randomized search process performed by SAFARI has two parameters, M and N. There are N independent searches that start from randomly generated starting points. The algorithm tries to improve the cardinality of the initial diagnoses (while preserving their consistency) by randomly "flipping" fault literals. The change of a sign of literal is done in one direction only: from faulty to healthy.

[2] In the formal description of the algorithm we describe a method for determining the initial candidates.

Algorithm 1 SAFARI: A greedy stochastic hill climbing algorithm for approximating the set of minimal diagnoses.

```
1: function HILLCLIMB(DS, α, M, N) returns a trie
       inputs: DS = ⟨SD, COMPS, OBS⟩, diag. system
               α, term, observation
               M, integer, climb restart limit
               N, integer, number of tries
2:     n ← 0
3:     while n < N do
4:         ω ← RANDOMDIAGNOSIS(SD, α)
5:         m ← 0
6:         while m < M do
7:             ω′ ← IMPROVEDIAGNOSIS(ω, ρ)
8:             if SD ∧ α ∧ ω′ ⊭⊥ then
9:                 ω ← ω′
10:                m ← 0
11:            else
12:                m ← m + 1
13:            end if
14:        end while
15:        unless ISSUBSUMED(R, ω) then
16:            ADDTOTRIE(R, ω)
17:            REMOVESUBSUMED(R, ω)
18:        end unless
19:        n ← n + 1
20:    end while
21:    return R
22: end function
```

Each attempt to find a minimal diagnosis terminates after M unsuccessful attempts to "improve" the current diagnosis stored in ω. Thus, increasing M will lead to a better exploitation of the search space and, possibly, to diagnoses of lower cardinality, while decreasing it will improve the overall speed of the algorithm.

It is possible two diagnostic searches to result in the same minimal diagnosis. To prevent this, we store the generated diagnoses in a trie R (Forbus and de Kleer 1993), from which it is straightforward to extract the resulting diagnoses by recursively visiting its nodes. A diagnosis ω is added to the trie R by the function ADDTOTRIE, iff no subsuming diagnosis is contained in R (the ISSUBSUMED subroutine checks on that condition). After adding a diagnosis ω to the resulting trie R, all diagnoses contained in R and subsumed by ω are removed by a call to REMOVESUBSUMED.

Optimality Analysis

This section shows that our algorithm can be configured to guarantee finding a minimal diagnosis in weak fault models in polynomial time (given a SAT oracle such as BCP). We also show that SAFARI trades off optimality for speed or for more general diagnostic framework, such as strong-fault models.

Optimality Guarantee

The hypothesis which comes next is well-studied in prior work (de Kleer, Mackworth, and Reiter 1992) as it determines the conditions in which minimal diagnoses represent all diagnoses of a model and an observation. This paper is interested in the hypothesis from the computational viewpoint: it defines a class of models for which it is possible to establish a theoretic bound on the optimality and performance of SAFARI.

Hypothesis 1 (Minimal Diagnosis Hypothesis)**.** Let $\mathrm{DS} = \langle \mathrm{SD}, \mathrm{COMPS}, \mathrm{OBS} \rangle$ be a diagnostic system and ω' a diagnosis for an arbitrary observation α. The Minimal Diagnosis Hypothesis (MDH) holds in DS iff for any health assignment ω such that $Lit^{-}(\omega) \supset Lit^{-}(\omega')$, it holds that ω is also a diagnosis.

It is easy to show that MDH holds for all weak-fault models. There are other theories $\mathrm{SD} \notin \mathbf{WFM}$ for which MDH holds (e.g., one can directly construct a theory as a conjunction of terms for which MDH to hold). Unfortunately, no necessary condition is known for MDH to hold in an arbitrary SD. The lemma which comes next is a direct consequence of MDH and weak-fault models.

Lemma 1. *Given a system* $\mathrm{DS} = \langle \mathrm{SD}, \mathrm{COMPS}, \mathrm{OBS} \rangle$, $\mathrm{SD} \in \mathbf{WFM}$*, and a diagnosis* ω *for some observation* α*, it follows that* ω *is non-minimal iff another diagnosis* ω' *can be obtained by changing the sign of exactly one negative literal in* ω*.*

Proof (Sketch). From Def. 2 and $\mathrm{SD} \in \mathbf{WFM}$, it follows that if ω is a minimal diagnosis, any diagnosis ω' obtained by flipping one positive literal in ω is also a diagnosis. Applying the argument in the other direction gives us the above statement. □

Our greedy algorithm starts with an initial diagnosis and then randomly flips faulty assumable variables. We now use the MDH property to show that, starting with a non-minimal diagnosis ω, the greedy stochastic diagnosis algorithm can monotonically reduce the size of the "seed" diagnosis to obtain a minimal diagnosis through appropriately flipping a fault variable from faulty to healthy; if we view this flipping as search, then this search is continuous in the diagnosis space.

Proposition 1 (Minimal Diagnosis Guarantee)**.** *Given a diagnostic system* $\mathrm{DS} = \langle \mathrm{SD}, \mathrm{COMPS}, \mathrm{OBS} \rangle$*, an observation* α*, and* $\mathrm{SD} \in \mathbf{WFM}$*, the greedy stochastic algorithm can be configured to compute a minimal diagnosis.*

Proof. Let us configure Alg. 1 with $M = |\mathrm{COMPS}|$. The diagnosis improvement loop starts, in the worst case, from a health assignment ω which is a conjunction of negative literals only. Necessarily, in this case, ω is a diagnosis as $\mathrm{SD} \in \mathbf{WFM}$. A diagnosis ω' that is subsumed by ω would be found with at most M consistency checks (provided that ω' exists) as M is set to be equal to the number of literals in ω and there are no repetitions in randomly choosing of which literal to flip next. If, after trying all the negative literals in ω, there is no diagnosis, then from Lemma 1 it follows that ω is a minimal diagnosis.

Through a simple inductive argument, we can continue this process until we obtain a minimal diagnosis. □

From Proposition 1 it follows that there is a an upper bound of $|\text{COMPS}|^2$ consistency checks for finding a single minimal diagnosis. In most of the practical cases, however, we are interested in finding an approximation to *all* minimal diagnoses. As a result the complexity of the optimally configured SAFARI algorithm becomes $O(|\text{COMPS}|^2 S)$, where S is the number of minimal diagnoses for the given observation. The number of assumable variables in a system of practical significance may exceed thousands, rendering an optimally configured SAFARI computationally too expensive. In the next section we show that while it is more computationally efficient to configure $M < |\text{COMPS}|$, it is still possible to find a minimal diagnosis with high probability.

Performance and Optimality Trade-Offs

In contrast to deterministic algorithms, in the SAFARI algorithm there is no absolute guarantee that the optimum solution (minimal diagnosis) is found. Below we will provide an intuition behind the performance of the SAFARI algorithm by means of an approximate, analytical model that estimates the probability of reaching a diagnostic solution of specific minimality for weak-fault models. We will start by considering a single run of the algorithm without retries where we will assume the existence of only one minimal diagnosis. Next, we will extend the model by considering retries. Finally, we take into account the fact that there usually is a large number of minimal diagnoses.

Basic Model Consider $\text{DS} = \langle \text{SD}, \text{COMPS}, \text{OBS} \rangle$ such that $\text{SD} \in \textbf{WFM}$ and an observation α such that α manifests only one minimal diagnosis ω. For the argument that follows we will configure SAFARI with $M = 1$ and we will assume that the starting solution is the trivial "all faulty" diagnosis.

When SAFARI randomly chooses a faulty variable and flips it, we will be saying that it is a "success" if the new candidate is a diagnosis and, a "failure" otherwise. Let k denote the number of steps that the algorithm successfully traverses in the direction of the minimal diagnosis of cardinality $|\omega|$. Thus k also measures the number of variables whose values are flipped from faulty to healthy in the process of climbing.

Let $f(k)$ denote the pdf of k. In the following we derive the probability $p(k)$ of successfully making a transition from k to $k+1$. A diagnosis at step k has k positive literals and still $|\text{COMPS}| - k$ negative literals. The probability of the next variable flip being successful equals the probability that the next negative to positive flip out of the $|\text{COMPS}| - k$ negative literals does not conflict with a negative literal belonging to a diagnosis solution ω. Consequently, of the $|\omega| - k$ literals only $\text{COMPS}| - |\omega| - k$ literals are allowed to flip, and therefore the success probability equals:

$$p\,(k) = \frac{|\text{COMPS}| - |\omega| - k}{|\text{COMPS}| - k} = 1 - \frac{|\omega|}{|\text{COMPS}| - k} \qquad (2)$$

The search process can be modeled in terms of the Markov chain depicted in Fig. 2, where k equals the state of the algorithm. Running into an inconsistency is modeled by the transitions to the state denoted "fail".

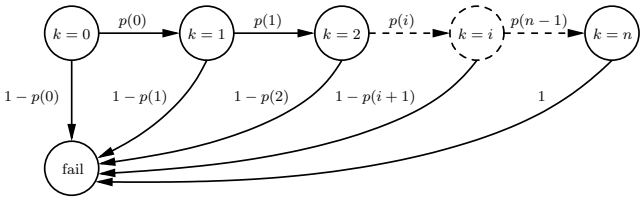

Figure 2: Model of a SAFARI run for $M = 1$ and a single diagnosis ω ($n = |\text{COMPS}| - |\omega|$)

The probability of exactly attaining step k (and subsequently failing) is given by:

$$f(k) = (1 - p(k+1)) \prod_{i=0}^{k} p(i) \qquad (3)$$

After substituting (2) in (3) we receive the pdf of k:

$$f(k) = \frac{|\omega|}{|\text{COMPS}| - k + 1} \prod_{i=0}^{k} \left[1 - \frac{|\omega|}{|\text{COMPS}| - i} \right] \qquad (4)$$

At the optimum goal state $k = |\text{COMPS}| - |\omega|$ the failure probability term in (4) is correct as it equals unity.

If p were independent of k, f would be according to a geometric distribution, which implies that chances of reaching the goal state $k = |\text{COMPS}| - |\omega|$ are slim. However, the fact that p decreases with k moves probability mass to the tail of the distribution, which works in favor of reaching higher-k solutions. For instance, for single-fault solutions ($|\omega| = 1$) the distribution becomes uniform. Fig. 3 shows the pdf for problem instances with $|\text{COMPS}| = 100$ for an increasing fault cardinality $|\omega|$.

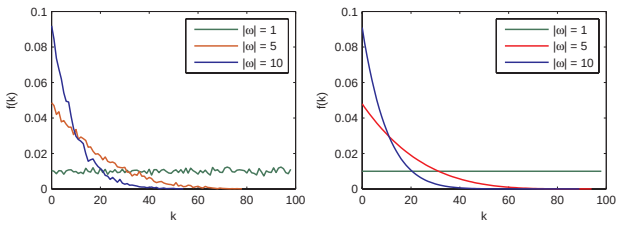

Figure 3: Empirical (left) and analytic (right) $f(k)$ for no retries and a single diagnosis

In the next section we show that *retries* will further move probability mass towards the optimum, providing the increasing distribution tail, needed for (almost always) reaching optimality.

Modeling Retries In this section we extend the model to account for retries, which has a profound effect on the resulting pdf of f. Again, consider the transition between step k and $k+1$ where the algorithm can spend up to $m = 1, \ldots, M$ retries before bailing out. As can be seen by the algorithm (cf. Alg. 1), when a variable flip produces an inconsistency a retry is executed while m is incremented.

From basic combinatorics we can compute the probability of having a diagnosis after flipping any of M different negative literals at step k. Similar to (2), at stage k there are $|\mathrm{COMPS}| - k$ faulty literals from which M are chosen (as variable "flips" leading to inconsistency are recorded and not attempted again, there is no difference between choosing in advance or one after another the M variables). The probability of advancing from stage k to stage $k+1$ becomes:

$$p(k) = 1 - \frac{\binom{|\omega|}{M}}{\binom{|\mathrm{COMPS}|-k}{M}} \tag{5}$$

The progress of SAFARI can be modeled for values of $M > 1$ as a Markov chain, similar to the one shown in Fig. 2 with the transition probability of p replaced by p'. The resulting pdf of the number of successful steps becomes:

$$f(k) = \frac{\binom{|\omega|}{M}}{\binom{|\mathrm{COMPS}|-k+1}{M}} \prod_{i=0}^{k} \left[1 - \frac{\binom{|\omega|}{M}}{\binom{|\mathrm{COMPS}|-i}{M}} \right] \tag{6}$$

It can be seen that (4) is a private case of (6) for $M = 1$. The retry effect on the shape of the pdf is profound. Whereas for single-fault solutions the shape for $M = 1$ is uniform, for $M = 2$ most of the probability mass is already located at the optimum $k = |\mathrm{COMPS}| - |\omega|$. Fig. 4 plots f for a number of problem instances with increasing M. As expected, the effect of M is extremely significant. Note that in case of the real system, for $M = |\mathrm{COMPS}|$ the pdf would consist of a single, unit probability spike at $|\mathrm{COMPS}| - |\omega|$.

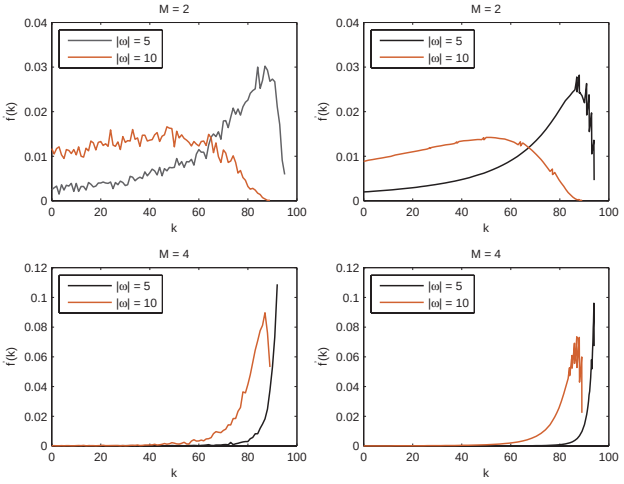

Figure 4: Empirical (left) and analytic (right) $f(k)$ for multiple retries and a single diagnosis

Although the above transition model is not amenable to analytic treatment, the graphs immediately shows that for large M the probability of moving to $k = |\mathrm{COMPS}| - |\omega|$ is very large indeed. Hence, for reasonable values of M relative to $|\mathrm{COMPS}|$ the pdf has a considerable probability mass located at $k = |\mathrm{COMPS}| - |\omega|$.

Thus far, we have only considered a single solution. In general, however, depending on the observation, there are many minimal diagnoses, with all of them or a fraction being of minimal cardinality.

Experimental Results

This section discusses empirical results measured from an implementation of SAFARI. For the experiments, we have performed a total of 131 184 diagnostic computations on 64 dual-CPU nodes belonging to a cluster. Each node contains two 2.4 GHz AMD Opteron DP 250 processors and 4 Gb of RAM.

In all experiments, SAFARI was configured with $M = 8$ and $N = 4$, that is, maximum number of 8 retries before giving up the climb, and a total of 4 attempts. To provide more precise average run-time performance data for this randomized algorithm, we average the SAFARI run-times over 10 runs on each model and observation vector.

Implementation Notes and Test Set Description

We have implemented SAFARI in approximately 1 000 lines of C code (excluding the LTMS, interface, and DPLL code) and it is a part of the LYDIA package.[3]

For implementing RANDOMDIAGNOSIS (cf. Alg. 1) we have modified the POSIT satisfiability solver (Freeman 1995). The modification ensured a random initial assignment for the greedy stochastic search. For the consistency check performed in line 8, we have used LTMS which, while incomplete in general, is complete in our experimentation setup (it can be shown that given a full health assignment, values of all inputs and outputs, well-formedness of the circuit, etc., LTMS is complete).

Traditionally, MBD algorithms have been tested on diagnostic models of digital circuits like the ones included in the ISCAS85 benchmark suite (Brglez and Fujiwara 1985). As models derived from ISCAS85 are large from the diagnostic perspective, we have also considered four medium-sized circuits from the 74XXX family (Hansen, Yalcin, and Hayes 1999).

Since the performance of diagnostic algorithms depends on the observation vectors, we have averaged our experimental results over a number of different observations for each model. These observations lead to diagnoses of different minimal-cardinality values, varying from 1 to nearly the maximum for the respective circuits (for the 74XXX models it is the maximum). The experiments omit nominal scenarios as they are trivial from the viewpoint of MBD.

Table 1 provides an overview of the fault diagnosis benchmark used for our experiments. The third and fourth columns show the number of observable and assumable variables, which characterize the size of the circuits. The last column shows the number of observation vectors with which we have tested the models.

Comparison to ALLSAT and Model Counting

We have compared the performance of SAFARI to that of a pure SAT-based approach, which uses blocking clauses for avoiding duplicate diagnoses (Jin, Han, and Somenzi 2005).

[3] LYDIA, SAFARI, and the diagnostic benchmark can be downloaded from `http://fdir.org/lydia/`.

Name	Description	|OBS|	|COMPS|	Tests
74182	4-bit CLA	14	19	250
74L85	4-bit comparator	14	33	150
74283	4-bit adder	14	36	202
74181	4-bit ALU	22	65	350
c432	27-channel interrupt controller	43	160	301
c499	32-bit SEC circuit	73	202	835
c880	8-bit ALU	86	383	1 182
c1355	32-bit SEC circuit	73	546	836
c1908	16-bit SEC/DEC	58	880	846
c2670	12-bit ALU	373	1 193	1 162
c3540	8-bit ALU	72	1 669	756
c5315	9-bit ALU	301	2 307	2 038
c6288	32-bit multiplier	64	2 416	404
c7552	32-bit adder	315	3 512	1 557

Table 1: An overview of the 74XXX/ISCAS85 benchmark circuits

Although SAT encodings have worked efficiently on a variety of other domains, such as planning, the health modeling makes the diagnostic problem so underconstrained that an uninformed ALLSAT strategy (i.e., a search not exploiting the continuity imposed by the weak-fault modeling) is quite inefficient, even for small models.

To substantiate our claim, we have experimented with the state-of-the-art satisfiability solver RELSAT, version 2.02 (Bayardo and Pehoushek 2000). Instead of enumerating all solutions and filtering the minimal diagnoses only, we have performed model-counting, whose relation to MBD has been extensively studied (Kumar 2002). While it was possible to solve the two smallest circuits, the solver did not terminate for any of the larger models within the predetermined time of 1 h. The results are shown in Table 2.

Name	models	time [s]
74182	3.9896×10^7	1
74L85	8.3861×10^{14}	340
74283	$> 1.0326 \times 10^{15}$	$> 3\,600$
74181	$> 5.6283 \times 10^{15}$	$> 3\,600$

Table 2: Model count and time for counting

The second column of Table 2 shows the model count returned by RELSAT, with sample observations from our benchmark. The rightmost column reports the time for model counting. This slow performance on relatively small diagnostic instances leads us to the conclusion that specialized solvers like SAFARI are better suited for finding minimal diagnoses than off-the-shelf ALLSAT (model counting) implementations that do not encode inference properties similar to those encoded in SAFARI.

Comparison to Complete Methods and Absolute Performance of the Algorithm

Table 3 shows the results from comparing SAFARI to implementations of two state-of-the-art complete and deterministic diagnostic algorithms: an implementation of CDA* (Williams and Ragno 2007) that has been modified to be complete, and HA* (Feldman and van Gemund 2006).

	CDA*	HA*	SAFARI		
Name	% of tests	% of tests	% of tests	$t_{\min}$ [ms]	$t_{\max}$ [ms]
74182	100	100	100	0.41	1.25
74L85	100	100	100	0.78	7.47
74283	100	100	100	0.92	4.84
74181	79.1	100	100	2.04	6.94
c432	74.1	71.1	100	8.65	38.94
c499	29	24.1	100	14.19	31.78
c880	11.6	12.4	100	48.08	88.87
c1355	3.8	10.8	100	95.03	141.59
c1908	0	6.1	100	237.77	349.96
c2670	0	5	100	500.54	801.12
c3540	0	1.1	100	984.31	1 300.98
c5315	0	1.1	100	1 950.12	2 635.71
c6288	0	3.5	100	2 105.28	2 688.34
c7552	0	3.9	100	4 557.4	6 545.21

Table 3: Comparison of CDA*, HA*, and SAFARI [% of tests solved] and absolute performance of SAFARI [ms]

Table 3 shows, for each model/algorithm combination, the percentage of all tests for which a diagnosis could be computed within 1 minute. As it is visible from the third column, SAFARI could find diagnoses for all observation vectors, while the performance of the two deterministic algorithms degraded as model size increased. Furthermore, we have observed a degradation of the performance of CDA* and HA* with increased cardinality of the minimal-cardinality diagnoses, while, as we will see below, the performance of SAFARI remained unaffected.

The two rightmost columns of Table 3 show the absolute performance of SAFARI. This varies from under a millisecond for the small models, to approx. 7 s. These fast absolute times show that SAFARI is suitable for on-line reasoning tasks, where autonomy depends on speedy diagnosis computation. For each model, the minimum and maximum time for computing a diagnosis has been computed. These values are shown under columns t_{min} and t_{max}, respectively. The small range of $t_{max} - t_{min}$ confirms our theoretical predictions that SAFARI is insensitive to the fault cardinalities of the diagnoses it computes. The performance of CDA* and HA*, on the other hand, is dependent on the fault cardinality and quickly degrades with increasing cardinality.

Optimality of the Greedy Stochastic Search

From the result produced by the complete diagnostic methods (CDA* and HA*), we know the exact cardinalities of the minimal-cardinality diagnoses for some of the observations. By considering the observations that lead to single and double faults, we have evaluated the average optimality of SAFARI. Table 4 shows these optimality results for the greedy search. The second column shows the number of observations leading to single faults for each weak-fault model. The

third column shows the average cardinality of SAFARI. The second and third columns are repeated for double faults.

	Single Faults		Double Faults	
Name	# of tests	Cardinality	# of tests	Cardinality
74182	50	1	50	2
74L85	50	1.04	50	2.12
74283	50	1.08	50	2.2
74181	50	1.19	50	2.25
c432	58	1.19	82	2.46
c499	84	1.49	115	3.27
c880	50	1	50	2.01
c1355	84	1.66	6	2.15
c1908	52	1.05	–	–
c2670	29	1.03	13	2.12
c3540	8	1.01	–	–
c5315	14	1	7	2
c6288	13	1	1	2
c7552	27	1.01	16	2

Table 4: Optimality of SAFARI [average cardinality]

Table 4 shows that, for weak fault models, the average cardinality returned by SAFARI is very close to the optimal values for both single and double faults. The c1355 model shows the worst-case results for the single-fault observations, while c499 is the most difficult weak-fault model for computing a double-fault diagnosis. These results can be easily improved by increasing M and N, as discussed in the previous section. It is beyond the scope of this article to define the relationship between diagnosis minimality and the parameters M and N; we leave it as future work to study this relationship.

Conclusion

We have described a greedy stochastic algorithm for computing diagnoses within a model-based diagnosis framework. We have shown that subset-minimal diagnoses can be computed optimally in weak fault models, and analyzed the optimality of the computed diagnoses for other configurations of SAFARI.

We have applied SAFARI to a suite of benchmark models, and shown significant performance improvements for multiple-fault diagnoses, compared to two state-of-the-art deterministic algorithms, CDA* and HA*. Our results indicate at least an order-of-magnitude speedup over these algorithms for multiple-fault diagnoses. Moreover, whereas the search complexity for the deterministic algorithms tested increases with fault cardinality, the search complexity for SAFARI is virtually independent of fault cardinality.

We argue that SAFARI can be of broad practical significance, as it can compute a significant fraction of cardinality-minimal diagnoses for systems too large or complex to be diagnosed by existing deterministic algorithms.

References

Bayardo, R. J., and Pehoushek, J. D. 2000. Counting models using connected components. In *Proc. AAAI'00*, 157–162.

Brglez, F., and Fujiwara, H. 1985. A neutral netlist of 10 combinational benchmark circuits and a target translator in fortran. In *Proc. ISCAS'85*, 695–698.

Bylander, T.; Allemang, D.; Tanner, M.; and Josephson, J. 1991. The computational complexity of abduction. *Artificial Intelligence* 49:25–60.

Davis, M.; Logemann, G.; and Loveland, D. 1962. A machine program for theorem-proving. *Communications of the ACM* 5(7):394–397.

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

de Kleer, J.; Mackworth, A.; and Reiter, R. 1992. Characterizing diagnoses and systems. *Artificial Intelligence* 56(2-3):197–222.

Eiter, T., and Gottlob, G. 1995. The complexity of logic-based abduction. *Journal of the ACM* 42(1):3–42.

Feldman, A., and van Gemund, A. 2006. A two-step hierarchical algorithm for model-based diagnosis. In *Proc. AAAI'06*, 827–833.

Feldman, A.; Provan, G.; and van Gemund, A. 2008. Computing observation vectors for max-fault min-cardinality diagnoses. In *Proc. AAAI'08*.

Forbus, K., and de Kleer, J. 1993. *Building Problem Solvers*. MIT Press.

Freeman, J. W. 1995. *Improvements to Propositional Satisfiability Search Algorithms*. Ph.D. Dissertation, University of Pennsylvania.

Freuder, E. C.; Dechter, R.; Ginsberg, B.; Selman, B.; and Tsang, E. P. K. 1995. Systematic versus stochastic constraint satisfaction. In *Proc. IJCAI 95*, volume 2, 2027–2032.

Friedrich, G.; Gottlob, G.; and Nejdl, W. 1990. Physical impossibility instead of fault models. In *Proc. AAAI*, 331–336.

Hansen, M.; Yalcin, H.; and Hayes, J. 1999. Unveiling the ISCAS-85 benchmarks: A case study in reverse engineering. *IEEE Design & Test* 16(3):72–80.

Jin, H.; Han, H.; and Somenzi, F. 2005. Efficient conflict analysis for finding all satisfying assignments of a boolean circuit. In *Proc. TACAS'05*, 287–300.

Kask, K., and Dechter, R. 1999. Stochastic local search for Bayesian networks. In *Proc. AISTAT'99*, 113–122.

Kumar, T. K. S. 2002. A model counting characterization of diagnoses. In *Proc. DX'02*, 70–76.

McAllester, D. 1990. Truth maintenance. In *Proc. AAAI'90*, volume 2, 1109–1116.

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

Vatan, F.; Barrett, A.; James, M.; Williams, C.; and Mackey, R. 2003. A novel model-based diagnosis engine: Theory and applications. In *IEEE Aerospace Conf.*, volume 2, 901–910.

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