Approximate Model-Based Diagnosis Using Greedy Stochastic Search

Alexander Feldman[1], Gregory Provan[2], and Arjan van Gemund[1]

[1] Delft University of Technology,
Faculty of Electrical Engineering, Mathematics and Computer Science,
Mekelweg 4, 2628 CD, Delft, The Netherlands,
E-mail: {a.b.feldman,a.j.c.vangemund}@tudelft.nl *
[2] University College Cork, Department of Computer Science,
College Road, Cork, Ireland,
E-mail: g.provan@cs.ucc.ie **

Abstract. Most algorithms for computing diagnoses within a model-based diagnosis framework are deterministic. Such algorithms guarantee soundness and completeness, but are NP-hard. To overcome this complexity problem, we propose a novel *approximation approach* for multiple-fault diagnosis, based on a greedy stochastic algorithm called SAFARI (StochAstic Fault diagnosis AlgoRIthm). SAFARI sacrifices guarantees of optimality, but for models in which component failure modes are defined solely in terms of a deviation from nominal behavior (known as weak fault models), it can compute 80-90% of all cardinality-minimal diagnoses, several orders of magnitude faster than state-of-the-art deterministic algorithms. We have applied this algorithm to the 74XXX and ISCAS-85 suites of benchmark combinatorial circuits, demonstrating order-of-magnitude speedup over a well-known deterministic algorithm, CDA*, for multiple-fault diagnoses.

1 Introduction

Model-Based Diagnosis (MBD) is an area of abductive or model-based inference in which a system model is used, together with observations about system behavior, to isolate sets of faulty components (diagnoses) that explain the observed behavior. The standard MBD formalization [1] 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.

However, 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 $\Sigma_1^{\mathbf{P}}$-complete [2, 3]. Since almost

* Supported by STW grant DES.7015.

** Supported by SFI grant 04/IN3/I524.

all proposed MBD algorithms have been complete and exact (with some authors proposing possible trade-offs between completeness and faster consistency checking by employing methods such as BCP [4]), the complexity problem remains a major challenge to MBD.

To overcome this complexity problem, we propose a novel *approximation approach* for multiple-fault diagnosis, based on stochastic algorithms. 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 the theoretical justification for the success of this algorithm, i.e., that minimal-cardinality diagnosis over weak fault models can be solved in poly-time (calling the incomplete SAT-solver BCP), and that more general frameworks are also amenable to this class of algorithm. (3) We apply this algorithm to a suite of benchmark combinatorial circuits, demonstrating order-of-magnitude speedup over a well-known deterministic algorithm, CDA*, 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.

2 Related Work

MBD is an instance of 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 deterministic, 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.

The most closely-related diagnostic approach is that of Vatan et al. [5], 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.

Stochastic algorithms have been discussed in the framework of constraint satisfaction [6] and Bayesian network inference [7]. The latter two approaches can be used for solving suitably translated MBD problems. It is often the case, though, that these encodings are more difficult for search than specialized ones.

3 Technical Background

The discussion starts by formalizing some basic notions in MBD, extending the notions proposed by de Kleer et al. [8]. A *model* of an artifact is represented as a propositional **Wff** over some set of variables V. Discerning disjoint subsets of them as failure-mode variables (*assumables*) and observable variables (*observables*) gives us a diagnostic system.

Definition 1 (Diagnostic System). *A diagnostic system* DS *is defined as the triple* DS $= \langle \mathrm{SD}, \mathrm{COMPS}, \mathrm{OBS} \rangle$*, where* SD *is a propositional theory describing the behavior of the system,* COMPS *is a set of assumable variables in* SD*, and* OBS *is a set of some observable variables in* SD*.*

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

3.1 A Running Example

The Boolean circuit shown in Fig. 1 provides some intuitions behind Safari. The subtractor consists of seven components: an inverter, two or-gates, two xor-gates, and two and-gates.

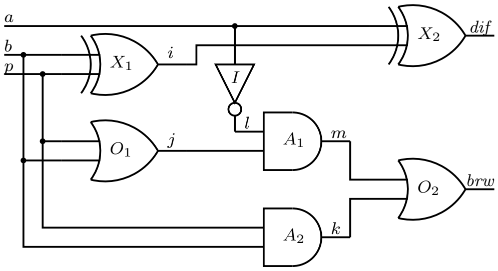

Fig. 1. A subtractor circuit

The expression $h \Rightarrow (o \Leftrightarrow \neg i)$ models 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 is $h \Rightarrow (o \Leftrightarrow i_1 \vee i_2)$. Finally, an xor-gate is specified as $h \Rightarrow (o \Leftrightarrow \neg(i_1 \Leftrightarrow i_2))$.

These 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 receiving a propositional formula for SD, given by $X_1 \Rightarrow (i \Leftrightarrow \neg(b \Leftrightarrow p)), X_2 \Rightarrow (dif \Leftrightarrow \neg(a \Leftrightarrow i)), O_1 \Rightarrow (j \Leftrightarrow b \vee p), O_2 \Rightarrow (brw \Leftrightarrow m \vee k), A_1 \Rightarrow (m \Leftrightarrow l \wedge j), A_2 \Rightarrow (k \Leftrightarrow b \wedge p), I \Rightarrow (a \Leftrightarrow \neg l)$. The set of component (assumable) variables is COMPS $= \{X_1, X_2, O_1, O_2, A_1, A_2, I\}$. The set of observable variables is OBS $= \{a, b, p, \mathit{dif}, \mathit{brw}\}$.

3.2 Minimal Diagnosis and Fault Modes

The traditional query in MBD results in finding terms of assumable variables which are explanations for the system description and an observation. The first definition of diagnosis uses a set notation.

Definition 2 (Diagnosis). *Given a system* DS $= \langle \mathrm{SD}, \mathrm{COMPS}, \mathrm{OBS} \rangle$ *and an observation term* α *over the variables in* OBS*, a diagnosis for* DS *and* α *is a set* $D \subseteq$ COMPS *such that:*

$$\mathrm{SD} \wedge \alpha \wedge \left[\bigwedge_{c \in D} \neg h_c\right] \wedge \left[\bigwedge_{c \in (\mathrm{COMPS} \setminus D)} h_c\right] \not\models \bot$$

In the MBD literature, a range of types of "preferred" diagnosis have been proposed. The first ordering we consider is a subset-ordering:

Definition 3 (Subset-Minimal Diagnosis). *A diagnosis* D *is subset-minimal, if no proper subset* $D' \subset D$ *exists such that* D' *is also a diagnosis.*

Throughout this paper we interchangeably use a propositional notation for expressing diagnoses. In this case we simply construct a conjunction of literals, each literal having a negative sign if its respective variable is in D and a positive sign otherwise. Consider the example from Fig. 1 and an observation $\alpha = a \wedge b \wedge p \wedge \neg \mathit{dif} \wedge \neg \mathit{brw}$. In this case $D_1 = \{A_1, A_2, X_1\}$ is a diagnosis and $D_2 = \{A_1, X_1\}$ is a minimal diagnosis (there are seven more minimal diagnoses for SD $\wedge\ \alpha$). Alternatively, instead of D_1 we may write $D_1' = \neg X_1 \wedge X_2 \wedge O_1 \wedge O_2 \wedge \neg A_1 \wedge \neg A_2 \wedge I$.

The *cardinality* of a diagnosis D is the size of D and is denoted as $|D|$. It represents the number of faulty components in COMPS given SD and α. Next to computing minimal diagnoses, it is of interest to MBD to compute some or all minimal-cardinality diagnoses, given a diagnostic problem.

Definition 4 (Cardinality-Minimal Diagnosis). *A diagnosis* D *is a cardinality-minimal diagnosis if it is a minimal diagnosis and no other diagnosis* D' *exists such that* $|D| < |D'|$.

The cardinality of a cardinality-minimal diagnosis computed from a system description SD and an observation α is denoted as $\mathit{MinCard}(\mathrm{SD} \wedge \alpha)$. For our example and the observation $\alpha = a \wedge b \wedge p \wedge \neg \mathit{dif} \wedge \neg \mathit{brw}$, it follows that $\mathit{MinCard}(\mathrm{SD} \wedge \alpha) = 2$. Note that in this case all minimal diagnoses are also cardinality-minimal diagnoses.

There are subset-minimal diagnoses which are not cardinality-minimal diagnoses. Consider, for example, the diagnostic system DS $= \langle \mathrm{SD}, \mathrm{COMPS}, \mathrm{OBS} \rangle$, where SD $= (h_1 \wedge h_2 \wedge x) \vee (h_4 \wedge x)$, COMPS $= \{h_1, h_2, h_3, h_4\}$, OBS $= \{x\}$, and $\alpha = x$. In this case, $D_1 = \{h_1, h_2, h_3\}$ is a non-subset-minimal diagnosis, $D_2 = \{h_1, h_2\}$ and $D_3 = \{h_4\}$ are subset-minimal diagnoses, but only D_3 is a cardinality-minimal diagnosis.

Another important diagnosis preference relation that we consider is one induced by a probability distribution over failure mode instantiations, γ, i.e., $Pr : \gamma \to [0, 1]$. If we assume that all components fail independently, then the probability of a multiple-fault F is just the product of the component probabilities, i.e., $Pr(F) = \prod_{F_i \in F} Pr(F_i)$.

Definition 5 (Probability-Minimal Diagnosis). *Given a non-trivial probability assignment to component failure modes, a probability-maximal diagnosis ω is a fault-mode such that $\nexists$ any other diagnosis ω' such that $Pr(\omega') > Pr(\omega)$.*

In the following, we will focus on subset-minimal and cardinality-minimal diagnoses; these two relationships mean that our results will also hold for a probability-minimal diagnosis.

MBD defines two broad classes of fault models, based on weak and strong modeling assumptions for abnormal behavior.

Weak-fault models define normative behavior of their components only, i.e., models which specify no fault-modes.

Definition 6 (Weak Fault Model). *A diagnostic system* DS *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 for $1 \leq i, j \leq n$, $\{h_i\} \subseteq$ COMPS, $F_j \in$* **Wff***, and none of h_i appears in F_j.*

In contrast, *strong fault models* specify the faulty behavior of their components. One way to define such a model is by partitioning the set of observable variables OBS into two subsets IN and OUT (denoting input and output variables respectively). Defining values to IN and COMPS then allows us to find a *unique assignment* to the values in OUT.

Definition 7 (Strong Fault Model). *Given a system* DS *and a partition* OBS $=$ IN $\cup$ OUT, DS $\in$ **SFM** *if for any instantiation ϕ of all variables in* IN $\cup$ COMPS*, it holds that there is exactly one term ψ such that $\phi \models$* SD $\wedge\, \psi$ *and ψ is an instantiation of all variables in* OUT*.*

In the following we show how our stochastic algorithm can compute subset- and cardinality-minimal diagnoses for SD such that SD $\in$ **WFM**, and indicate how the algorithm can be generalized to cover SD $\in$ **SFM**.

4 Stochastic MBD Algorithm

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

4.1 A Simple Example (Continued)

We now show what happens when we apply A* and stochastic search to our running example.

A deterministic A* search for the above diagnosis discovers it after expanding 127 nodes and performing 19 consistency checks. Even enabling conflict focusing may not result in a small number of generated nodes and consistency checks (this depends on the model, observation and conflict extraction mechanism), which shows how deterministic diagnosis search becomes impractical with bigger systems.

We will now show a two-step diagnostic process that requires fewer variable assignments and consistency checks. 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[3]. 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 $\alpha = a \wedge b \wedge p \wedge \neg dif \wedge \neg brw$, if $X_1 \wedge X_2$ is in the initial "guess" it will prove inconsistent with $\mathrm{SD} \wedge \alpha$ and another quintuple fault candidate will be guessed.

Assume that the second candidate is $\omega'_2 = \neg X_1 \wedge \neg X_2 \wedge O_1 \wedge \neg O_2 \wedge \neg A_1 \wedge \neg A_2 \wedge I$. Clearly, $\mathrm{SD} \wedge \alpha \wedge \omega' \not\models \bot$. The search algorithm may next try to improve the diagnosis by "flipping" A_2. The candidate $\omega'_3 = \neg X_1 \wedge \neg X_2 \wedge O_1 \wedge \neg O_2 \wedge \neg A_1 \wedge A_2 \wedge I$ is a valid quadruple fault diagnosis and it can be improved twice more by "flipping" X_2 and O_2. This gives us the final double-fault $\omega'_6 = \neg X_1 \wedge X_2 \wedge O_1 \wedge O_2 \wedge \neg A_1 \wedge A_2 \wedge I$. 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.

4.2 A Greedy Stochastic Algorithm

The greedy stochastic algorithm, which we introduce next, finds multiple cardinality-minimal diagnoses (if such exist).

The stochastic algorithm presented in this paper uses the valuation function $Pr : \gamma \rightarrow [0, 1]$. In the analysis of our algorithm we use Pr to determine if an assignment to a health variable denotes a failure or a healthy mode. The complexity of the algorithm presented in this paper is not sensitive to Pr and the only use of the probabilities is to guide the search for a better efficiency.

The randomized search process performed by SAFARI has two parameters, M and N. There are N independent searches that start from randomly generated

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

candidates. After an initial candidate ω is found to be consistent with $\mathrm{SD} \wedge \alpha$, i.e., ω is a diagnosis, the algorithm tries to improve the cardinality of the diagnosis (while preserving its consistency) by randomly "flipping" fault literals.

Algorithm 1 Safari: A stochastic hill climbing algorithm for approximating a set of cardinality-minimal diagnoses.

```
function HillClimb($\mathrm{DS}, \alpha, M, N, Pr$) returns a trie
    inputs: $\mathrm{DS} = \langle \mathrm{SD}, \mathrm{COMPS}, \mathrm{OBS} \rangle$, a diagnostic system
            $\alpha$, term, observation
            $M$, integer, climb restart limit
            $N$, integer, number of tries
            $Pr$, a valuation function
    local variables: $m, n$, integers
                     $\omega, \omega'$, terms
                     $R$, a trie
    $n \leftarrow 0$
    while $n < N$ do
        $\omega \leftarrow$ RandomCandidate($Pr$)
        if $\mathrm{SD} \wedge \alpha \wedge \omega \not\models \bot$ then
            $m \leftarrow 0$
            while $m < M$ do
                $\omega' \leftarrow$ ImproveDiagnosis($Pr, \omega$)
                if $\mathrm{SD} \wedge \alpha \wedge \omega' \not\models \bot$ then
                    $\omega \leftarrow \omega'$
                    $m \leftarrow 0$
                else
                    $m \leftarrow m + 1$
                end if
            end while
            unless IsSubsumed($R, \omega$) then
                AddToTrie($R, \omega$)
                RemoveSubsumed($R, \omega$)
            end unless
            $n \leftarrow n + 1$
        end if
    end while
    return $R$
end function
```

Each attempt to find a cardinality-minimal diagnosis terminates after M unsuccessful attempts to change the value of a fault variable to healthy state. Thus, increasing M will lead to a better exploitation of the search space and possibly diagnoses of lower cardinality, while decreasing it will improve the overall speed of the algorithm.

Similar to deterministic methods for MBD, Safari uses a SAT-based procedure for checking the consistency of $\mathrm{SD} \wedge \alpha \wedge \omega$. Because SD and α do not change in consistency checks, using an LTMS [9] can improve search efficiency. The implementation of Safari combines a BCP-based LTMS to check for inconsistencies. If a candidate is consistent, a second DPLL-based check is invoked for completeness.

The RandomCandidate function generates a candidate diagnosis, used for "seeding" the diagnostic search. The valuation function can be modified to provide a more informed starting point, thus decreasing the number of "climbing" steps. The initial diagnosis ω should be of high cardinality, to increase the likelihood of $\text{SD} \wedge \alpha \wedge \omega \not\models \perp$. In order to do that, we generate an instantiation of ω by using Pr and *scaling* the a priori probabilities in Pr to bias the probability density function (pdf) from which we draw the initial candidates. Consider an example in which each component $h \in \text{COMPS}$ fails with a probability of 5%. The valuation function is $Pr(h = \textbf{False}) = 0.05$. We may use a scaling coefficient $k = 5$ which would lead to RandomCandidate returning a candidate with a quarter of the components failing.

The biasing of Pr can improve the efficiency of Safari by exploiting knowledge about the likelihood of the cardinality of the cardinality-minimal diagnosis. In particular, when expecting cardinality-minimal diagnoses of high cardinality Pr should be configured to return an initial fault of higher cardinality. If the expected faults are of small cardinality, the search may start from a candidate with fewer faulty components, in which case more attempts (increased N) would be necessary to find local diagnoses close to the global optimum.

The ImproveDiagnosis function generates a candidate ω' of smaller cardinality than the diagnosis ω, supplied as an argument. This is done by flipping a random faulty literal in ω. The probability of flipping a faulty literal l in ω is inverse proportional to the a priori probability $Pr(l)$. Consider a diagnosis $\omega = \neg h_1 \wedge \neg h_2 \wedge \neg h_3 \wedge \neg h_4$, where $Pr(h_1 = \textbf{False}) = Pr(h_2 = \textbf{False}) = 0.1$ and $Pr(h_3 = \textbf{False}) = Pr(h_4 = \textbf{False}) = 0.025$. In this case ImproveDiagnosis would return $\omega' = h_1 \wedge \neg h_2 \wedge \neg h_3 \wedge \neg h_4$ or $\omega' = \neg h_1 \wedge h_2 \wedge \neg h_3 \wedge \neg h_4$, each of the two with probability of 0.4, and $\omega' = \neg h_1 \wedge \neg h_2 \wedge h_3 \wedge \neg h_4$ or $\omega' = \neg h_1 \wedge \neg h_2 \wedge \neg h_3 \wedge h_4$ the latter with probability 0.1.

There is no guarantee that two diagnostic searches, starting from a random diagnoses, would not lead to the same cardinality-minimal diagnosis. To prevent this, we store the generated diagnoses in a trie R [10], from which it is straightforwards 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.

5 Optimality and Complexity of Greedy Stochastic Algorithm

One of the key factors in the success of the proposed algorithm is the exploitation of the continuity of the search-space of weak fault models, where by continuity we mean that we can monotonically reduce the cardinality a non-minimal diagnosis. This section shows that our algorithm can guarantee finding minimal diagnoses in weak fault models in polynomial time (given a SAT oracle such as BCP), and trades off optimality in more general diagnostic frameworks, such as cardinality-minimal diagnostic inference or strong fault models.

5.1 Cardinality-Minimal Diagnosis in Weak-Fault Models

We will show some properties of minimal diagnoses. These properties are also true for fault-modes with a slight adaptation of the notation.

Hypothesis 1 (Minimal Diagnosis Hypothesis) *Let* SD *be a system description and D' a diagnosis. The Minimal Diagnosis Hypothesis (MDH) holds in* SD *if for any D such that $D \supset D'$ it holds that D is also a diagnosis.*

It has been shown in [8] that if a model $\mathrm{SD} \in \mathbf{WFM}$ (cf. Def. 6), then the Minimal Diagnosis Hypothesis (MDH) holds. There are other theories $\mathrm{SD}' \not\in \mathbf{WFM}$ for which MDH holds. Unfortunately, no necessary condition is known for MDH to hold in an arbitrary SD'.

Using MDH together with Def. 3, if $\mathrm{SD} \in \mathbf{WFM}$ then we immediately have the following lemma.

Lemma 1. *If D is a diagnosis of a diagnosis problem* DS *and MDH holds for* DS*, then there is a subset-minimal diagnosis D' that subsumes D, i.e., $D' \subseteq D$.*

Our greedy algorithm starts with a "seed" diagnosis and then randomly flips faulty component variables. We now use the MDH property to show that, starting with a non-minimal diagnosis D, the greedy stochastic diagnosis algorithm can monotonically reduce the size of "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.[4]

Theorem 1. *Given a a weak fault model* $\mathrm{SD} \in \mathbf{WFM}$ *and a non-subset-minimal diagnosis D with $|D| = \mu \leq n$ faulty components, the greedy stochastic diagnosis algorithm is guaranteed to compute a minimal diagnosis.*

We can now prove the following correctness result:

Theorem 2. *The greedy stochastic diagnosis algorithm is guaranteed to compute a subset-minimal diagnosis for a weak fault model with $|\mathrm{OBS}| = n$ in $O(\Theta n)$ time, where $O(\Theta)$ is the complexity of a consistency check.*

Since consistency-checking for this model class can be done in polynomial time (using BCP propagation), we have just demonstrated a polynomial-time algorithm for computing minimal diagnoses in a weak-fault-model system description (**WFM**). This result has been shown in the literature for a divide-and-conquer approach [12], but to our knowledge no current algorithm takes advantage of this approach.

[4] We omit proofs due to space limitations, but refer the reader to [11] for all proofs.

We now provide a brief intuition behind the performance of the Safari algorithm. A formal derivation of the performance/cost model is beyond the scope of this paper and appears in [11].

Assume we have an MBD problem with S optimal diagnoses of cardinality C. A key to the good performance of the algorithm is the fact that the pdf of the number of successful consecutive steps $j = 0, 1, \ldots$ toward the optimal solution ($j = |\mathrm{COMPS}| - C$), is generally an *increasing* function, rather than decreasing; i.e., the probability mass is concentrated near the optimum solution. This somewhat counter-intuitive fact is caused by two phenomena. First, the retry mechanism has a huge impact on the shape (slope) of the pdf. Whereas for $M = 0$ (no retries) the pdf is still close to a geometric distribution (i.e., a *decreasing* function), even for relatively small M (e.g., $M = 4$, as in our experiments) the retry process turns the slope into an increasing function. Second, the fact that there are usually multiple ($S > 1$) C-fault diagnoses implies that the attained j value is essentially the *maximum* of the j values per individual C-fault solution. The associated, order-statistical effect further increases the positive slope of the pdf. Thus, even for one run, the probability of reaching the optimum solution is quite high.

The effect of N (taking the maximum of the attained j over multiple runs) also translates into an order-statistical effect. However, this third effect is not nearly as powerful as the effect of M (and S). Hence, optimizing Safari's performance/cost ratio is primarily a matter of increasing M, rather than N.

Finally, note that the number of retries also guarantees optimality in those cases where the attained solution cardinality is less or equal to M. For instance, when a solution with $C = 4$ is found (where $M = 4$), it must be optimal, as the remaining of lower-cardinality solutions have all been (unsuccesfully) tried.

5.2 More General Diagnostic Frameworks

This section now addresses how the algorithm will perform in more general diagnostic frameworks, such as computing cardinality-minimal diagnoses or dealing with strong fault models. In generalizing beyond subset-minimal diagnosis computation, less restrictive definitions are computationally harder [3].

We can modify Safari to compute a wide variety of diagnoses, e.g., subset-minimal, non-minimal, cardinality-minimal, etc., by modifying the type of diagnosis computed together with the trie maintenance and subsumption testing of lines 16-18 of the pseudo-code.

The complexity of Safari is derived in a straightforward way.

Proposition 1. *The time complexity of Alg. 1 is $O(MN \log N\Theta)$, where Θ is the complexity of the consistency checking procedure.*

The $\log N$ factor comes from the trie maintenance, which contains a maximum number of N diagnoses with some ordering imposed on their literals. Note that the average case complexity of consistency checking, although exponential in the worst case, is low polynomial when incomplete methods like BCP are used [13] or when the model is highly-observable.

In this more general case we have no completeness guarantee, as we do for subset-minimal diagnoses. Note that this is effectively a polynomial-time algorithm that trades off some small amount of completeness and optimality for significant improvements in efficiency relative to deterministic diagnosis algorithms. In these more general frameworks, from a theoretical perspective one can only provide probabilistic arguments about the likelihood of finding particular classes of diagnosis; this is a topic of future work. The experimental results presented later in this article show that significant speedups over complete algorithms are possible while losing relatively little diagnostic completeness.

In generalizing from weak to strong fault models, the key difference is the increased difficulty in "guessing" the initial diagnosis for Safari. For a weak fault model, we are guaranteed to find a subset-minimal diagnosis by choosing an initial diagnosis with all components faulty,[5] but this guess is not guaranteed to be consistent in a strong fault model. Developing a robust diagnosis initialization algorithm for strong fault models is a topic for future research.

6 Experimental Results

Next, we discuss some empirical results measured from an implementation of Safari. In the following, for any models SD, it holds that SD $\in$ **WFM**.

We have implemented Safari in approximately 700 lines of C code (excluding LTMS and DPLL) and it is a part of the Lydia[6] package.

Table 1. Test model sizes

Name	Description	H	V	C_w	O
74182	4-bit CLA	19	47	75	14
74283	4-bit adder	40	89	130	14
74L85	4-bit comparator	41	93	134	14
74181	4-bit ALU	62	138	216	22
c432	27-channel interrupt controller	160	356	514	43
c499	32-bit SEC circuit	202	445	714	73
c880	8-bit ALU	383	826	1 112	86
c1355	32-bit SEC circuit	546	1 133	1 610	73
c1908	16-bit SEC/DEC	880	1 793	2 378	58
c2670	12-bit ALU	1 193	2 543	3 269	221
c3540	8-bit ALU	1 669	3 388	4 608	72
c5315	9-bit ALU	2 307	4 792	6 693	301
c6288	32-bit multiplier	2 416	4 864	7 216	64
c7552	32-bit adder	3 512	7 230	9 656	313

[5] We start with half the components faulty and use the parameter M to restart if our initial guess is incorrect.

[6] Lydia can be downloaded from `http://fdir.org/lydia/`.

Table 1 summarizes the benchmark suite. All models are derived from the 74XXX and ISCAS85 family of benchmark circuits. We have added an assumable variable to each gate in each model. The benchmark implements weak fault models for each component, in a way similar to the example. The same valuation function Pr has been used in all the experiments. In particular, $Pr(h = \textbf{False}) = 0.01$, and $Pr(h = \textbf{True}) = 0.99$.

The number of assumable variables is denoted as H. The total number of variables is denoted V and the number of clauses in the CNF representation is denoted as C_w. The number of observable variables is denoted as O.

All the experiments described in this paper are performed on a host with 1.86 GHz Pentium M CPU and 2 Gb of RAM.

6.1 Comparison to HA* and Multiple-Fault Scalability

We compare the efficiency of Safari to HA* [14] with Max-Fault Min-Cardinality (MFMC) observation vectors [15]. MFMC observation vectors maximize the number of faults in the cardinality-minimal diagnoses consistent with a model. From the deterministic algorithms, HA* performs better than CDA* [4] in finding cardinality-minimal diagnoses of high cardinality. On the other hand, CDA* is very fast in finding faults of small cardinality (single and double faults) and we will then compare Safari to CDA*.

First, we use small models with observations maximizing the number of faults in a cardinality-minimal diagnosis. The results, shown in Table 2, illustrate an advantage of Safari: its performance does not degrade when the fault cardinality increases. We have run two groups of experiments: finding a single cardinality-minimal diagnosis and finding all cardinality-minimal diagnoses.

Table 2. Times [ms] for diagnosing MFMC faults by Alg. 1 and HDA*

		Single Diagnosis				Multiple Diagnoses				
Name	MFMC	T_h	T	C	K	T'_h	T'	K'	N'	M'
74182	5	38	2	5	300	171	143	242	800	4
74283	5	4 708	4	5.5	814	27 606	5759	665	10 000	4
74L85	3	143	4	3	100	1 281	184	90	400	4
74181	7	106 386	9	7	3 817	634 739	31 377	3 236.7	20 000	8

Next, we describe the notation in the column headings of Table 2. MFMC is the number of faults in the cardinality-minimal diagnosis consistent with the MFMC observation. T_h is the time for finding a single diagnosis by the HA* algorithm. T is the time for finding a single diagnosis by Alg. 1. C is the cardinality of the diagnosis generated by Safari. K is the number of cardinality-minimal diagnoses as counted by the deterministic algorithm HA*. The time for finding all of these diagnoses by HA* is denoted as T'_h. T' is the time for Safari to find K' multiple cardinality-minimal diagnoses.

Safari is designed to find multiple cardinality-minimal diagnosis. In these experiments, we have configured it with a number N of runs in order to return a small number of diagnoses, and we have ignored all but the first diagnosis. We performed the single fault experiments, shown in Table 2, with $N = 8$ and $M = 4$. For the multiple diagnoses, the algorithm is configured with $M = M'$ and $N = N'$ as described in Table 2.

We have averaged the results of all the experiments involving Safari over 10 runs. Safari is a local search algorithm, and hence it can compute a suboptimal diagnosis. This was the case in the 74283 model experiments, in which 5 out of 10 runs returned a cardinality-minimal diagnosis with 6 faults, while the global optimum has 5 faults, resulting in the 5.5 value for C in Table 2.

Table 2 demonstrates the main advantage of Safari, that its performance does not depend on the number of faults in the cardinality-minimal diagnoses. Furthermore, Safari finds a good coverage of all cardinality-minimal diagnoses. This coverage varies from 81% in 74182 to 90% in 74L85.

We have also run Safari on bigger circuits. Figure 2 shows the time for finding multiple-fault diagnosis for c880 and c7552. The diagnosis was run k times where k is the number of outputs in each circuit. For run $x = 0$, we have assigned random values to the inputs and computed (by using propagation) the values of all the outputs. For $x = 1$ we have flipped one output in α, for $x = 2$ two outputs, etc. Thus on the horizontal axis in Fig. 2 we have the Hamming distance between α and an observation consistent with a no-fault (nominal) diagnosis. Again, Safari showed no dependency on α, only a small increase in the diagnostic time for c7552 due to the difficulty of finding an initial diagnosis. The latter can be easily overcome by scaling the initial a-priori probabilities.

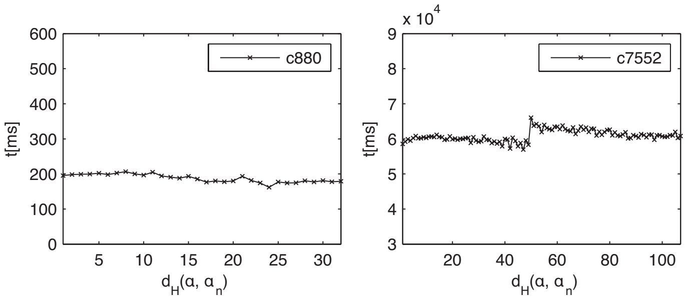

Fig. 2. Diagnosis time of Safari with multiple observation vectors

6.2 Comparison to CDA*

Table 3 shows the result from finding single and double faults with arbitrary (manually computed) observations. Finding single faults is known to be trivial in MBD and CDA* performs well on these simple problems. The time for finding a single fault by CDA* is shown in column T_1^*. The time for the CDA* algorithm to find a double fault is shown in column T_2^*. The CDA* algorithm could not compute a double fault diagnosis in less than 10 min time for the five biggest circuits, in which cases we have interrupted the search.

Table 3. Running times [ms] of CDA* and Alg. 1

	Single-Fault			Double-Fault		
Name	T_1^*	T_1	C_1	T_2^*	T_2	C_2
c432	9	32	1	5	34	2
c499	3	53	1	152	64	2
c1908	34	95	1	509	94	2
c880	18	186	1	62 068	186	2
c1355	11	285	1	4 300	310	2
c2670	1 425	1 362	1	–	1 352	2
c3540	3 050	3 080	1	–	3 115	2
c5315	13 849	19 322	1	–	19 764	2
c6288	18 317	11 070	1.4	–	11 366	2.2
c7552	35 801	37 269	1	–	37 585	2.2

The times for the stochastic algorithm to discover a single and a double fault are denoted as T_1 and T_2 respectively. We have used $M = 4$ and $N = 8$ for the search, that is, maximum number of four retries before giving up the climb, and a total of 4 attempts. In some of the cases, our stochastic algorithm could not find a cardinality-minimal diagnosis, but a suboptimal one. We have shown the cardinality of the results for single and double faults in columns C_1 and C_2, respectively. Again, the values of T_1, C_1, T_2, and C_2 are averaged over 10 runs.

The relatively small number of restarts lead to small overall search time and in a very few cases to suboptimal result for the diagnosis cardinality. Increasing N would lead to finding a global cardinality-minimal diagnosis in all the cases. We note that increasing M would not help to finding a diagnosis of lower cardinality.

As is visible from Table 3, in the single fault scenario, CDA* performs better than the stochastic algorithm, which is not surprising as in CDA* all single fault candidates are tested first. On the other hand, the stochastic method performed 8 independent attempts to find a cardinality-minimal diagnosis which, having the overhead of consistency checking, led to the slightly worse performance for computing single fault diagnoses.

Again, the performance of Safari does not degrade when the number of faults increases. This is not the case with deterministic algorithms like CDA* or HA*. The time for the stochastic algorithm to find a double fault is the same as for finding a single fault, while CDA* suffers from a combinatorial explosion.

7 Conclusion and Future Work

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 that almost all cardinality-minimal diagnoses can be computed for more general fault models.

We have applied this algorithm to a suite of benchmark combinatorial circuits encoded using weak fault models, and shown significant performance improvements for multiple-fault diagnoses, compared to a well-known deterministic algorithm, CDA*. Our results indicate that, although the greedy stochastic algorithm is outperformed for the single-fault diagnoses, it shows at least an order-of-magnitude speedup over CDA* 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.

We have demonstrated the superior performance (over deterministic algorithms) of Safari for the class of discrete circuits specified using weak fault models. 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.

In future work, we plan to experiment on models with a combination of weak and strong failure-mode descriptions. We also plan on experimenting with a wider variety of stochastic methods, such as simulated annealing and genetic search, using a larger set of benchmark models. Last, we plan to apply our algorithms to a wider class of abduction and constraint optimization problems.

References

1. Reiter, R.: A theory of diagnosis from first principles. Artificial Intelligence **32**(1) (1987) 57–95
2. Bylander, T., Allemang, D., Tanner, M., Josephson, J.: The computational complexity of abduction. Artificial Intelligence **49** (1991) 25–60
3. Friedrich, G., Gottlob, G., Nejdl, W.: Physical impossibility instead of fault models. In: Proc. AAAI. (1990)
4. Williams, B., Ragno, R.: Conflict-directed A* and its role in model-based embedded systems. Journal of Discrete Applied Mathematics (2004)
5. Vatan, F., Barrett, A., James, M., Williams, C., Mackey, R.: A novel model-based diagnosis engine: Theory and applications. In: IEEE Aerospace Conf. (2003)
6. Freuder, E.C., Dechter, R., Ginsberg, B., Selman, B., Tsang, E.P.K.: Systematic versus stochastic constraint satisfaction. In: Proc. IJCAI 95. Volume 2. (1995)
7. Kask, K., Dechter, R.: Stochastic local search for Bayesian networks. In: Proc. AISTAT'99. (1999)
8. de Kleer, J., Mackworth, A., Reiter, R.: Characterizing diagnoses and systems. Artificial Intelligence **56**(2-3) (1992) 197–222
9. McAllester, D.: Truth maintenance. In: Proc. AAAI'90. Volume 2. (1990)
10. Forbus, K., de Kleer, J.: Building Problem Solvers. MIT Press (1993)
11. Feldman, A., Provan, G., van Gemund, A.: On the performance of Safari algorithms. Technical Report TUD-SERG-2007-011, TU Delft (2007)
12. Mozetič, I.: A polynomial-time algorithm for model-based diagnosis. In: Proc. ECAI'92. (August 1992) 729–733
13. Zabih, R., McAllester, D.: A rearrangement search strategy for determining propositional satisfiability. In: Proc. AAAI'88. (1988) 155–160
14. Feldman, A., van Gemund, A.: A two-step hierarchical algorithm for model-based diagnosis. In: Proc. AAAI'06. (July 2006)
15. Feldman, A., Provan, G., van Gemund, A.: Generating manifestations of max-fault min-cardinality diagnoses. In: Proc. DX'07. (May 2007)