Approximate Model-Based Diagnosis Using Greedy Stochastic Search

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

We propose a StochAstic Fault diagnosis AlgoRIthm, called Safari, which trades off guarantees of computing minimal diagnoses for computational efficiency. We empirically demonstrate, using the 74XXX and ISCAS85 suites of benchmark combinatorial circuits, that Safari achieves several orders-of-magnitude speedup over two well-known deterministic algorithms, CDA* and HA*, for multiple-fault diagnoses; further, Safari can compute a range of multiple-fault diagnoses that CDA* and HA* cannot. We also prove that Safari is optimal for a range of propositional fault models, such as the widely-used weak-fault models (models with ignorance of abnormal behavior), and strong-fault circuit models with stuck-at failure modes. We formally characterize this important subclass of strong-fault models with its set of subset-minimality diagnoses. By modeling the algorithm itself as a Markov chain, we provide exact bounds on the minimality of the diagnosis computed. Safari also displays strong anytime behavior, and will return a diagnosis after any non-trivial inference time.

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 (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.

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 (Bylander, Allemang, Tanner, & Josephson, 1991).

Since almost 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 (Williams & Ragno, 2007)), 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.

2. Related Work

We first compare Safari to "standard" MBD algorithms, and then to related algorithms.

On a gross level, one can classify the types of algorithms that have been applied to solve MBD as being based on search or compilation. The search algorithms take as input the diagnostic model and an observation, and then search for a diagnosis, which may be minimal with respect to some minimality criterion. Examples of search algorithms include A*-based algorithms, such as CDA* (Williams & Ragno, 2007) and hitting set algorithms (Reiter, 1987). Compilation algorithms pre-process the diagnostic model into a form that is more efficient for on-line diagnostic inference. Examples of such algorithms include the ATMS (de Kleer, 1986) and other prime-implicant methods (Kean & Tsiknis, 1993), DNNF (Darwiche, 1998), and OBDD (Bryant, 1992). To our knowledge, all of these approaches adopt exact methods to compute diagnoses; in contrast, Safari adopts a stochastic approach to computing diagnoses.

On the surface, Safari bears some resemblance to SAT local search algorithms; however, it is actually closer to optimization and abduction algorithms, since Safari solves an optimization problem, and its result depends on an input (OBS), whereas SAT (and its variants like #-SAT and MAXSAT) has no such dependence on an input.

Stochastic algorithms have been discussed in the framework of constraint satisfaction (Freuder, Dechter, Ginsberg, Selman, & Tsang, 1995) and Bayesian network inference (Kask & Dechter, 1999). 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.

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. (Vatan, Barrett, James, Williams, & Mackey, 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.

Our work bears the closest resemblance to preference-based or Cost-Based Abduction (CBA) (Charniak & Shimony, 1994; Santos Jr., 1994). Of the algorithmic work in this area, the primary paper that adopts stochastic local search is (Abdelbar, Gheita, & Amer, 2006). In this paper, Abdelbar et al. present a hybrid two-stage method that is based on Iterated Local Search (ILS) and Repetitive Simulated Annealing (RSA). The ILS stage of the algorithm uses a simple hill-climbing method (randomly flipping assumables) for the local search phase, and tabu search for the perturbation phase. RSA repeatedly applies Simulated Annealing (SA), starting each time from a random initial state. The hybrid method initially starts from an arbitrary state, or a greedily-chosen state. It then applies the ILS algorithm; if this algorithm fails to find the optimal solution after a fixed number τ of hill-climbing steps[1] or after a fixed number $\mathcal{R}$ of repetitions of the perturbation-local search cycle,[2] ILS-based search is terminated and the RSA algorithm is run until the optimal solution is found.

Our work differs from that of (Abdelbar et al., 2006) in several ways. First, our initial state is generated using a random SAT solution. The hill-climbing phase that we use next is similar to that of (Abdelbar et al., 2006); however, we randomly restart should hill-climbing not identify a “better” diagnosis, rather than applying tabu-search or simulated annealing. Our approach is simpler than that of (Abdelbar et al., 2006), and for the case of weak fault models is guaranteed to be optimal; in future work we plan to compare our approach to that of (Abdelbar et al., 2006) for strong fault models.

3. Technical Background

Our discussion continues by formalizing some MBD notions. This paper uses the traditional diagnostic definitions (de Kleer & 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*[3] variables gives us a diagnostic system.

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

[1] Hill-climbing proceeds as follows: given a current state s with a cost of $f(s)$, a neighbouring state s' is generated by flipping a randomly chosen assumable hypothesis. If $f(s')$ is better than $f(s)$, then s' becomes the current state; otherwise, it is discarded. If τ iterations elapse without a change in the current state, the local search exits.

[2] Perturbation-local search, starting from a current state s with a cost of $f(s)$, randomly chooses an assumable variable h, and applies tabu-search to identify a better state by flipping h based on its tabu status.

[3] 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.

Throughout this paper we will assume that OBS $\cap$ COMPS $= \emptyset$, SD $\not\models \perp$, and for any instantiation α of the variables in OBS, $\alpha \not\models \perp$. Not all propositional theories used as system descriptions are of interest to MBD. Diagnostic systems can be characterized by a restricted set of models, the restriction making the problem of computing diagnosis amenable to algorithms like the one presented in this paper. We consider two main classes of models.

Definition 2 (Weak-Fault Model)**.** A diagnostic system 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 $1 \leq i, j \leq n$, $\{h_i\} \subseteq$ COMPS, $F_j \in \mathbf{Wff}$, and none of h_i appears in F_j.

Note the conventional selection of the sign of the "health" variables $h_1, h_2, \ldots h_n$. Alternatively, negative literals, e.g., $f_1, f_2, \ldots f_n$ can be used to express faults, in which case a weak-fault model is in the form $(\neg f_1 \Rightarrow F_1) \wedge \ldots \wedge (\neg f_n \Rightarrow F_n)$. 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, & Reiter, 1992), or *implicit fault systems*. Alternatively, a model may specify faulty behavior for its components. In the following definition, with the aim of simplifying the formalism throughout this paper, we adopt a slightly restrictive representation of faults, allowing only a single fault-mode per assumable variable. This can be easily generalized by introducing multi-valued logic or suitable encodings (Hoos, 1999).

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

3.1 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 $h \Rightarrow [o \Leftrightarrow \neg (i_1 \Leftrightarrow i_2)]$.

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:

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

A strong-fault model for the Boolean circuit shown in Fig. 1 is constructed by assigning fault-modes to the different gate types. We will assume that, when malfunctioning, the output of an xor-gate has the value of one of its inputs, an or-gate can be stuck-at-one,

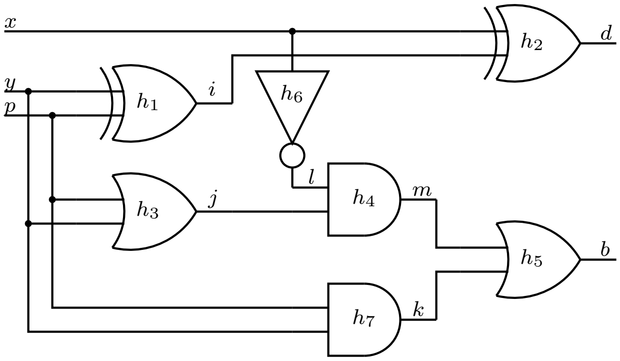


Figure 1: A subtractor circuit

an and-gate can be stuck-at-zero, and an inverter behaves like a buffer. This gives us the following strong-fault model formula for the Boolean subtractor circuit:

$$\begin{aligned}\mathrm{SD}_s = \mathrm{SD}_w \wedge [\neg h_1 \Rightarrow (i \Leftrightarrow y)] \wedge [\neg h_2 \Rightarrow (d \Leftrightarrow x)] \wedge (\neg h_3 \Rightarrow j) \wedge \\ \wedge (\neg h_4 \Rightarrow \neg m) \wedge (\neg h_5 \Rightarrow b) \wedge [\neg h_6 \Rightarrow (x \Leftrightarrow l)] \wedge (\neg h_7 \Rightarrow \neg k)\end{aligned} \tag{2}$$

For both models (SD_s and SD_w), the set of assumable variables is $\mathrm{COMPS} = \{h_1, h_2, \ldots, h_7\}$ and the set of observable variables is $\mathrm{OBS} = \{x, y, p, d, b\}$.

3.2 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 4 (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 $\mathit{Lit}^-(\mathrm{HA})$ and $\mathit{Lit}^+(\mathrm{HA})$, 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 5 (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 $\mathrm{SD} \wedge \alpha \wedge \omega \not\models \perp$.

Traditionally, other authors (de Kleer & 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_w 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

$\text{HA}_3 = \{\neg h_1 \wedge \ldots \wedge \neg h_7\}$) is a diagnosis for any instantiation of the observable variables in OBS (cf. Def. 2).

In the analysis of our algorithm we need the opposite notion of diagnosis, i.e., health assignments inconsistent with a model and an observation. In the MBD literature these assignments are usually called conflicts. Conflicts, however, do not necessarily instantiate all variables in COMPS. As in this paper we always use full health instantiations, the use of the term conflict is avoided to prevent confusion.

Definition 6 (Inconsistent Health Assignment)**.** Given a system $\text{DS} = \langle \text{SD}, \text{COMPS}, \text{OBS} \rangle$, an observation α over some variables in OBS, and a health assignment $\bar{\omega}$, $\bar{\omega}$ is an Inconsistent Health Assignment (IHA) iff $\text{SD} \wedge \alpha \wedge \bar{\omega} \models \perp$.

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

For the weak-fault model SD_w 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 $\omega_3^{\subseteq} = \neg h_1 \wedge h_2 \wedge h_3 \wedge h_4 \wedge \neg h_5 \wedge h_6 \wedge h_7$ and $\omega_4^{\subseteq} = \neg h_1 \wedge h_2 \wedge \ldots \wedge h_5 \wedge \neg h_6 \wedge \neg h_7$. 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 $\omega_3^{\subseteq}$ form a subset of the negative literals in ω_5.

Note that the set of all minimal diagnoses characterizes all diagnoses for a weak-fault model, but that does not hold in general for strong-fault models (de Kleer et al., 1992). In the latter case, 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. In our example, given SD_s and $\alpha_3 = \neg x \wedge \neg y \wedge \neg p \wedge b \wedge \neg d$, it follows that $\omega_6^{\subseteq} = h_1 \wedge h_2 \wedge \neg h_3 \wedge h_4 \wedge \ldots \wedge h_7$ is a diagnosis, but $\text{HA}_7 = h_1 \wedge h_2 \wedge \neg h_3 \wedge \neg h_4 \wedge \ldots \wedge h_7$ is not a diagnosis, despite the fact that the negative literals in HA_7 form a superset of the negative literals in ω_6.

Definition 8 (Cardinality of a Diagnosis)**.** The cardinality of a diagnosis, denoted as $|\omega|$, is defined as the number of negative literals in ω.

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

Definition 9 (Minimal-Cardinality Diagnosis)**.** A diagnosis $\omega^{\leq}$ is defined as minimal-cardinality if no diagnosis $\tilde{\omega}^{\leq}$ exists such that $|\tilde{\omega}^{\leq}| < \omega^{\leq}$.

The cardinality of a minimal cardinality diagnosis computed from a system description SD and an observation α is denoted as $\mathit{MinCard}(\text{SD} \wedge \alpha)$. For our example model SD_w and an observation $\alpha_4 = x \wedge y \wedge p \wedge \neg b \wedge \neg d$, it follows that $\mathit{MinCard}(\text{SD}_w \wedge \alpha_4) = 2$. Note that in this case all minimal diagnoses are also cardinality-minimal diagnoses.

A minimal cardinality diagnosis is a minimal diagnosis, but the opposite does not hold. There are minimal diagnoses which are not minimal cardinality diagnoses. Consider the example SD_w and α_2 given earlier in this section, and the two resulting minimal diagnoses $\omega_3^{\leq}$ and $\omega_4^{\leq}$. From these two, only $\omega_3^{\leq}$ is a minimal cardinality diagnosis.

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)

Consider the Boolean subtractor shown in Fig. 1, its weak-fault model SD_w given by (1), and the observation α_4 from the preceding section. The four minimal diagnoses characterizing SD_w and α_4 are: $\omega_1 = \neg h_1 \wedge h_2 \wedge h_3 \wedge h_4 \wedge \neg h_5 \wedge h_6 \wedge h_7$, $\omega_2 = h_1 \wedge \neg h_2 \wedge h_3 \wedge h_4 \wedge \neg h_5 \wedge h_6 \wedge h_7$, $\omega_3 = \neg h_1 \wedge h_2 \wedge \ldots \wedge h_6 \wedge \neg h_7$, and $\omega_4 = h_1 \wedge \neg h_2 \wedge h_3 \wedge \ldots \wedge h_6 \wedge \neg h_7$.

A naïve deterministic algorithm would check the consistency of all the $2^{|\mathrm{COMPS}|}$ possible health assignments for a diagnostic problem, 128 in the case of our running example. Furthermore, most deterministic algorithms first enumerate health assignments of small cardinality but with high a priori probability, which renders these algorithms impractical in situations when the minimal diagnosis is of a higher cardinality. Such performance is not surprising even when using state-of-the art MBD algorithms which utilize, for example conflict learning, or partial compilation, considering the bad worst-case complexity of finding all minimal diagnoses (cf. Sec. 5).

In what follows, we will show a two-step diagnostic process that requires fewer consistency checks. The first step involves finding a random non-minimal diagnosis as a starting point. The second step attempts to minimize the fault cardinality of this diagnosis by repeated modification of the diagnosis.

The first step is to find one random, possibly non-minimal diagnosis of $\mathrm{SD}_w \wedge \alpha_4$. Such a diagnosis we can obtain from a classical DPLL solver after modifying it in two ways: (1) not only determine if the instance is satisfiable but also extract the satisfying solution and (2) find a *random* satisfiable solution every time the solver is invoked. Both modifications are trivial, as DPLL solvers typically store their current variable assignments and it is easy to choose a random variable and value when branching instead of deterministic ones. The latter modification may possibly harm a DPLL variable or value selection heuristics, but later in this paper we will see that this is of no concern for the type of problems we are considering as diagnostic systems are typically underconstrained.

In the subtractor example we call the DPLL solver with $\mathrm{SD}_w \wedge \alpha_4$ as an input and we consider the random solution (and obviously a diagnosis) $\omega_5 = \neg h_1 \wedge h_2 \wedge \neg h_3 \wedge h_4 \wedge h_5 \wedge \neg h_6 \wedge \neg h_7$ ($|\omega_5| = 4$). In the second step of our stochastic algorithm, we will try to minimize ω_5 by repetitively choosing a random negative literal, "flipping" its value to positive (thus obtaining a candidate with a smaller number of faults), and calling the DPLL solver. If the new candidate is a diagnosis, we will try to improve further this newly discovered diagnosis, otherwise we will mark the attempt a "failure" and choose another negative literal. After some constant number of "failures" (two in this example), we will terminate the search and will store the best diagnosis discovered so far in the process.

After changing the sign of $\neg h_7$ in ω_5 we discover that the new health assignment is not consistent with $\mathrm{SD}_w \wedge \alpha_4$, hence it is not a diagnosis and we discard it. Instead, the algorithm attempts changing $\neg h_6$ to h_6 in ω_5, this time successfully obtaining a new diagnosis $\omega_6 = \neg h_1 \wedge h_2 \wedge \neg h_3 \wedge h_4 \wedge h_5 \wedge h_6 \wedge \neg h_7$ of cardinality 3. Next the algorithm tries to find a diagnosis of even smaller cardinality by randomly choosing $\neg h_1$ and $\neg h_7$ in ω_6, respectively, and trying to change their sign, but both attempts return an inconsistency. Hence the "climb" is aborted and ω_6 is stored as the current best diagnosis.

Repeating the process from another random initial DPLL solution, gives us a new diagnosis $\omega_7 = \neg h_1 \wedge \neg h_2 \wedge h_3 \wedge \neg h_4 \wedge h_5 \wedge h_6 \wedge \neg h_7$. Changing the sign of $\neg h_7$, again, leads to inconsistency, but the next two "flips" (of $\neg h_4$ and $\neg h_2$) lead to a double-fault diagnosis $\omega_8 = \neg h_1 \wedge h_2 \wedge \ldots \wedge h_6 \wedge \neg h_7$. The diagnosis ω_8 can not be improved any further as it is minimal. Hence the next two attempts to improve ω_8 fail and ω_8 is stored in the result.

This process is illustrated in Fig. 2, the search for ω_6 is on the left and for ω_8 on the right. Gates which are shown in solid black are "suspected" as faulty when the health assignment they participate in is tested for consistency, and inconsistent candidates are crossed-out.

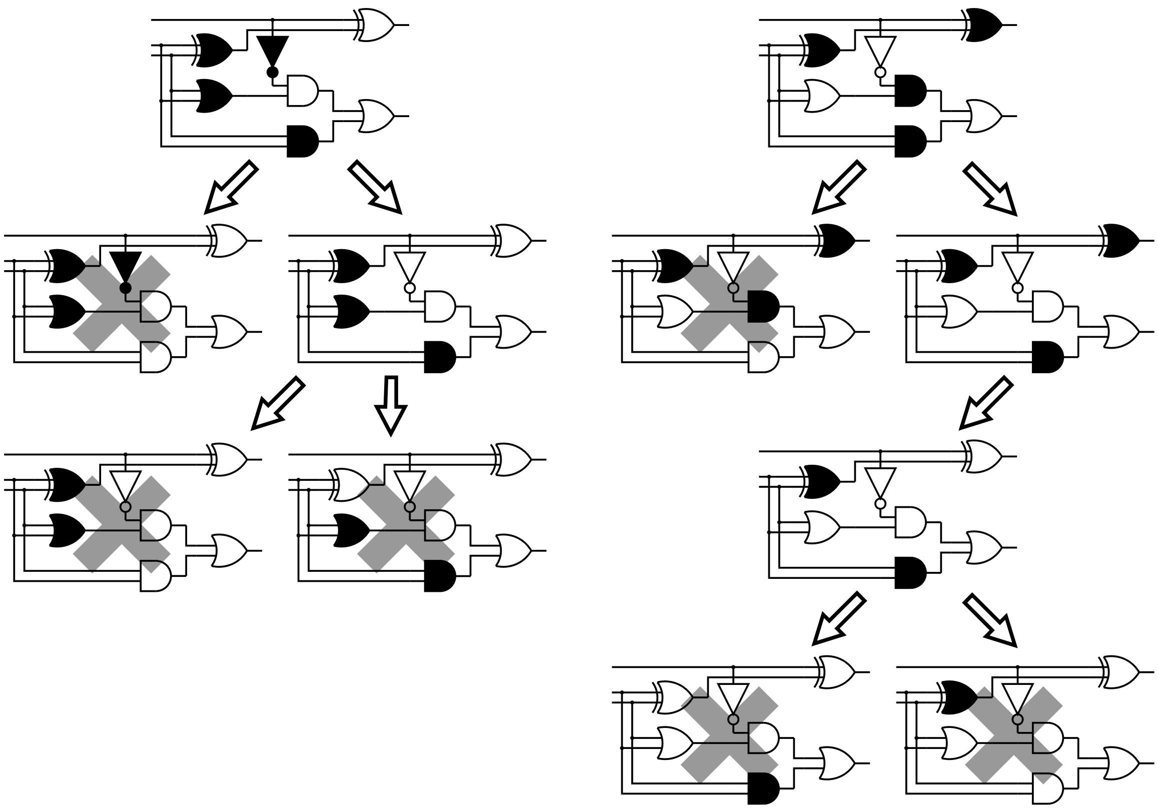
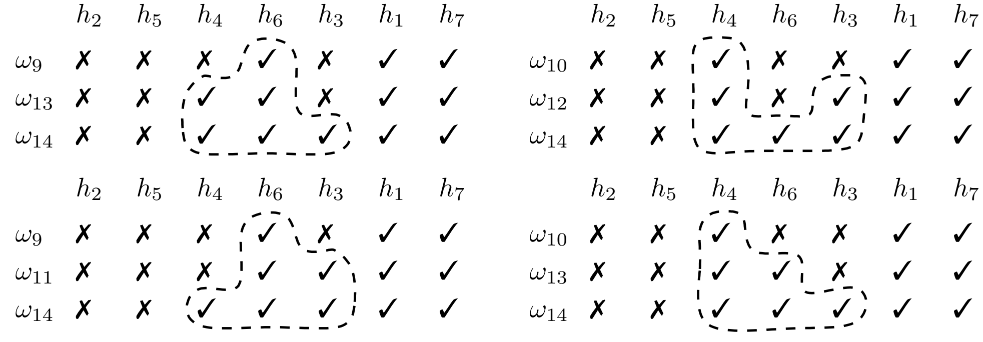


Figure 2: An example of a stochastic diagnostic process

Let us consider the result. We have found two diagnoses: ω_6 and ω_8, where ω_6 is not a minimal diagnosis. This we have done at the price of 11 calls to a DPLL subroutine. The suboptimal diagnosis ω_6 is of value as its cardinality is near the one of a minimal diagnosis. Hence we have demonstrated a way to find an approximation of all minimal diagnoses, while drastically reducing the number of consistency checks in comparison to a deterministic algorithm, sacrificing optimality. Next we will formalize our experience into an algorithm, the behavior of which we will analyze extensively in the section that follows.

Diagnosing a strong-fault model is known to be strictly more difficult than a weak-fault model (Friedrich, Gottlob, & Nejdl, 1990). In many diagnostic instances this problem is alleviated by the fact that there exist, although without a guarantee, continuities in the diagnostic search space similar to the one in the weak-fault models. Let us discuss the process of finding a minimal diagnosis of the subtractor's strong-fault model SD_s and the observation α_2 (both from Sec. 3.1).



	h_2	h_5	h_4	h_6	h_3	h_1	h_7
ω_9	✗	✗	✗	✓	✗	✓	✓
ω_{13}	✗	✗	✓	✓	✗	✓	✓
ω_{14}	✗	✗	✓	✓	✓	✓	✓

	h_2	h_5	h_4	h_6	h_3	h_1	h_7
ω_{10}	✗	✗	✓	✗	✗	✓	✓
ω_{12}	✗	✗	✓	✗	✓	✓	✓
ω_{14}	✗	✗	✓	✓	✓	✓	✓

	h_2	h_5	h_4	h_6	h_3	h_1	h_7
ω_9	✗	✗	✗	✓	✗	✓	✓
ω_{11}	✗	✗	✗	✓	✓	✓	✓
ω_{14}	✗	✗	✓	✓	✓	✓	✓

	h_2	h_5	h_4	h_6	h_3	h_1	h_7
ω_{10}	✗	✗	✓	✗	✗	✓	✓
ω_{13}	✗	✗	✓	✓	✗	✓	✓
ω_{14}	✗	✗	✓	✓	✓	✓	✓

Figure 3: Diagnoses of a strong-fault model

The six distinct diagnoses $\omega_9, \ldots, \omega_{14}$ of SD_s and α_2 are shown in Fig. 3. Of these only ω_9 and ω_{10} are minimal such that $|\omega_9| = |\omega_{10}| = 3$. It is visible in Fig. 3 that in all diagnoses component variables h_2 and h_5 are false, while h_1 and h_7 are true (healthy). Hence, any satisfying assignment of $\mathrm{SD}_s \wedge \alpha_2$ would contain $h_1 \wedge \neg h_2 \wedge \neg h_5 \wedge h_7$. Starting from the maximal-cardinality diagnosis $\omega_1 4$, we must "flip" the variables h_3, h_4, and h_6 in order to reach the two minimal diagnoses. The key insight is that, as shown in Fig. 3, this is always possible by "flipping" a single literal at a time from health to faulty and receiving another consistent assignment (diagnosis).

In what follows we will formalize our experience so far in a stochastic algorithm for finding minimal diagnoses.

4.2 A Greedy Stochastic Algorithm

A number of utility functions are used in the pseudocode listed in this paper. The FlipNegativeLiteral 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. Similarly, FlipPositiveLiteral changes the sign of a random positive literal. The implementation of RandomDiagnosis uses a modified DPLL solver returning a random SAT solution of $\mathrm{SD} \wedge \alpha$.

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 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.

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

```
function HillClimb($\mathrm{DS}, \alpha, M, N$) returns a trie
    inputs: $\mathrm{DS} = \langle \mathrm{SD}, \mathrm{COMPS}, \mathrm{OBS} \rangle$, diagnostic system
            $\alpha$, term, observation
            $M$, integer, climb restart limit
            $N$, integer, number of tries
    local variables: $\rho$, real, model weakness, $0 \leq \tau \leq 1$
    $n \leftarrow 0$
    while $n < N$ do
        $\omega \leftarrow$ RandomDiagnosis($\mathrm{SD}, \alpha$)          ▷ Get a random SAT solution.
        $m \leftarrow 0$
        while $m < M$ do
            $\omega' \leftarrow$ ImproveDiagnosis($\omega, \rho$)     ▷ Flip an "unflipped" health variable.
            if $\mathrm{SD} \wedge \alpha \wedge \omega' \not\models \bot$ then                 ▷ Consistency check.
                $\omega \leftarrow \omega'$
                $m \leftarrow 0$
            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 while
    return $R$
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.

There is no guarantee that two diagnostic searches, starting from random diagnoses, would not lead to the same minimal diagnosis. To prevent this, we store the generated diagnoses in a trie R (Forbus & 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.

5. Computational Complexity

This section first examines the complexity of the tasks that we are addressing, namely computing a single or all diagnoses with respect to a preference criterion. We then examine the complexity of the Safari algorithm itself.

5.1 Complexity of Diagnostic Inference

This section discusses the complexity of the problems in which we are interested, namely the problem of computing a single or the set of all minimal diagnoses, using two minimality criteria, subset minimality ($\subseteq$) and cardinality ($\leq$). We assume as input a CNF formula defined over a variable set V, of which $f = |\text{COMPS}|$ are assumable (or fault) variables.

By examining results for propositional abduction problems (PAPs) (Eiter & Gottlob, 1995), of which MBD is an instance, we will show that the worst-case complexity of computing diagnoses in propositional models is intractable. Further, the worst-case complexity of *approximating* diagnoses to within a fixed factor of optimal (e.g., to within a fixed factor of the minimal-cardinality diagnosis) is also intractable.

Table 1 introduces the notation that we use to define these 4 types of diagnosis.

Table 1: Summary of definitions of types of diagnosis of interest

Symbol	Diagnoses	Preference Criterion
$\omega^{\subseteq}$	1	$\subseteq$ (subset-minimality)
$\omega^{\leq}$	1	$\leq$ (cardinality-minimality)
$\Omega^{\subseteq}$	all	$\subseteq$ (subset-minimality)
$\Omega^{\leq}$	all	$\leq$ (cardinality-minimality)

The diagnosis problems of interest have worst-case complexity at the second level of the polynomial hierarchy. This can be shown by recognizing that MBD is an instance of a PAP, the complexity of which has been studied by Eiter and Gottlob (Eiter & Gottlob, 1995). Eiter and Gottlob show that for a propositional PAP, the problem of determining if a solution exists is Σ_2^P-complete. Computing a minimal diagnosis is a search problem, and hence it is more difficult to pose a decision question for proving complexity results. Consequently, one can just note that computing a diagnosis minimal with respect to $\subseteq$ / $\leq$ requires $O(|\text{COMPS}|)$ calls to an NP/Σ_2^P oracle respectively (Eiter & Gottlob, 1995).

The complexity of computing the *set* of all diagnoses is not easier than computing a single diagnosis. This problem is bounded from below by the problem of counting the number of diagnoses. This problem has been shown to be #co-NP-Complete (Hermann & Pichler, 2007). These results indicate that the MBD problems we are interested in, i.e., computing the set of minimal-cardinality diagnoses over propositional models, are intractable.

If we restrict our clauses to be Horn or definite Horn, then we can reduce the complexity of the problems that we are solving, at the expense of decreased model expressiveness. Recall that restricting the clauses to be Horn means that we exclude any components with multiple inputs or outputs. Hence for circuits we allow only inverters or buffers, for example.

Under a Horn-clause restriction, for SD $\in$ **WFM**, **SFM** (weak or strong fault models), computing a minimal (or cardinality-minimal) diagnosis is NP-complete (Bylander et al., 1991; Friedrich et al., 1990). In the case of computing all solutions, the problems drop down one level of the complexity hierarchy under the assumption of Horn clauses.

Safari approximates the intractable problems denoted in Table 1. We will show that for **WFM** with a single diagnosis Safari can compute this optimal diagnosis. For SD $\in$ **SFM**, Safari generates a sound but possibly sub-optimal diagnosis (or set of diagnoses).

Results on abduction problems indicate that the task of approximate diagnosis is intractable. Roth (Roth, 1996) has addressed the problems of abductive inference, and of approximating such inference. Roth focuses on counting the number of satisfying assignments for a range of AI problems, including some instances of propositional abduction problems. In addition, Roth shows that approximating the number of satisfying assignments for these problems is intractable.

Abdelbar (Abdelbar, 2004) has studied the complexity of approximating Horn abduction problems, showing that even for a particular Horn restriction of the propositional problem of interest, the approximation problem is intractable. In particular, for an abduction problem with costs assigned to the assumables (which can be used to model both the preference-orderings $\subseteq, \leq$), he has examined the complexity of finding the Least Cost Proof (LCP) for the evidence (OBS), where the cost of a proof is taken to be the sum of the costs of all hypotheses that must be assumed in order to complete the proof. For this problem he has shown that it is NP-hard to approximate an LCP within a fixed ratio r of the cost of an optimal solution, for any $r < 0$.

These results indicate that it is intractable to approximate, within a fixed ratio, a minimal diagnosis. In the following, we adopt a stochastic approach, and show that this approach cannot provide fixed-ratio guarantees. However, Safari trades off optimality for efficiency and can compute most diagnoses with high likelihood.

5.2 Complexity of Inference using Greedy Stochastic Search

This section defines the complexity of Safari, and its stochastic approach to computing sound but incomplete diagnoses. We will show that the primary determinant of the inference complexity is the consistency checking. Safari randomly computes a partial assignment π, and then checks if π can be extended to create a satisfying assignment during each consistency check, i.e., it checks the consistency of π with SD. This is solving the satisfiability problem (SAT), which is NP-complete (Garey & Johnson, 1990). We will show how we can use incomplete satisfiability checking to reduce this complexity, at the cost of completeness guarantees.

5.2.1 **WFM** With No Observable Variable Restrictions

This section examines the general problem solved by Safari, where we place no restrictions on observable values. In the following, we call Θ the complexity of a consistency check, and assume that there are f components that can fail, i.e., $f = |\text{COMPS}|$.

Lemma 1. *Given a diagnostic system* DS $= \langle \text{SD}, \text{COMPS}, \text{OBS} \rangle$ *with* SD $\in$ **WFM**, *the worst-case complexity of finding any minimal diagnosis is* $O(f^2\Theta)$, *where* Θ *is the cost of a consistency check.*

Proof. From Proposition 1 it follows that there is a an upper bound of f^2 consistency checks for finding a single minimal diagnosis, since at each step of the algorithm we must flip at most f literals. The total complexity is hence $O(f^2\Theta)$, since we perform a consistency check after each flip. □

In the case of BCP, we have the complexity as $O(f^2cn)$.

Lemma 2. *Given a diagnostic system* $\text{DS} = \langle \text{SD}, \text{COMPS}, \text{OBS} \rangle$ *with* $\text{SD} \in \mathbf{WFM}$*, the worst-case complexity under* $\mathbf{WFM}$ *of finding any minimal diagnosis is* $O(f^2cn)$ *when using BCP for consistency checks.*

In most 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 $O(f^2\xi)$, where ξ is the number of minimal diagnoses for the given observation.

5.2.2 Impact of Input/Output Restrictions

Consider the case where we have $SD \in \mathbf{WFM}$. The complexity of inference of Safari for computing *all* minimal diagnoses is $O(f^2\xi)$, where ξ is the total number of minimal diagnoses. If we place no restriction on the number of diagnoses using observables, then there is an exponential number of minimal diagnoses under $\subseteq$.

Lemma 3. *The number of diagnoses under* $\mathbf{WFM}$ *is* $O(2^f)$.

Proof. When $SD \in \mathbf{WFM}$, the worst-case number of diagnoses occurs when each diagnosis is of size $\lceil \frac{f}{2} \rceil$. In this case, there are $\binom{f}{\lceil \frac{f}{2} \rceil}$ diagnoses, and hence we have $O(2^f)$ possible diagnoses. □

However, if we assume that we partition the observables into two classes of observable variable, input OBS_I, which is assumed to be always "correct", and output OBS_O, which is allowed to take on anomalous values, then we can significantly restrict the number of allowable diagnoses. For q anomalous outputs there are at most n^q minimal diagnoses. Hence the complexity of finding all minimal diagnoses is polynomial in n, as is stated formally below.

Theorem 1. *Given a diagnostic system* $\text{DS} = \langle \text{SD}, \text{COMPS}, \text{OBS} \rangle$ *with* $\text{SD} \in \mathbf{WFM}$ *and a partition of* OBS *into input and output observables, the worst-case complexity of finding all minimal diagnoses is* $O(f^2n^q\Theta)$*, where* Θ *is the cost of a consistency check.*

This input/output distinction hence can make a big impact on the inference complexity of MBD. The question is whether this is a reasonable assumption to make. For a wide range of systems, such as circuits, process-control systems, and other mechanical systems, it is a reasonable assumption, since we can make two assumptions: (a) *causality* (input to output flows) can clearly be distinguished, and (b) one can assume that inputs are correct. If we cannot assume that inputs are correct, then we can model the system in such a way that we maintain the input/output distinction and can also diagnose anomalous input values. The systems for which this distinction truly does not hold are those for which as assumption of causality does not hold, which we might call *non-causal systems*.

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 will see that while it is more computationally efficient to configure $M < f$, it is still possible to find a minimal diagnosis with high probability.

6. Optimality Analysis of Greedy Stochastic Search

One of the key factors in the success of the proposed algorithm is the exploitation of the continuity of the search-space of diagnosis models, where by continuity we mean that we can monotonically reduce the cardinality of a non-minimal diagnosis. 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.

6.1 Optimality Guarantee for Minimal Diagnosis in Weak-Fault Models

The hypothesis which comes next is well studied in prior work (de Kleer et al., 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 theoretical 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 4. *Given a diagnostic 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. *Given a diagnostic system* $\text{DS} = \langle \text{SD}, \text{COMPS}, \text{OBS} \rangle$*, an observation* α*, and* $\text{SD} \in \mathbf{WFM}$*, the greedy stochastic algorithm can be configured to compute a minimal diagnosis.*

Proof. Let us configure Alg. 1 with $M = |\text{COMPS}|$. 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 $\text{SD} \in \mathbf{WFM}$. 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 4 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 will see that while it is more computationally efficient to configure $M < |\text{COMPS}|$, it is still possible to find a minimal diagnosis with high probability. We now formalize this notion of flips, in order to characterize when Safari will be able to compute a minimal diagnosis. We can define two types of flips, which differ on what kind of literal we are flipping.

Definition 10 (Superset Flip $\Phi_{\Uparrow}$)**.** Given a diagnostic system $\text{DS} = \langle \text{SD}, \text{COMPSOBS} \rangle$ and a health assignment HA with a non-empty set of positive literals ($Lit^{+}(\text{HA}) \neq \emptyset$), a superset flip $\Phi_{\Uparrow}$ turns one of the positive literals in HA to a negative literal, i.e., it creates a health assignment HA' with one more negative literal.

The converse type of flip is a subset flip:

Definition 11 (Subset Flip $\Phi_{\Downarrow}$)**.** Given a diagnostic system $\text{DS} = \langle \text{SD}, \text{COMPSOBS} \rangle$ and a health assignment HA with a non-empty set of negative literals ($Lit^{-}(\text{HA}) \neq \emptyset$), a subset flip $\Phi_{\Uparrow}$ turns one of the negative literals in HA to a positive literal, i.e., it creates a health assignment HA' with one more positive literal.

Safari operates by performing subset flips on non-minimal diagnoses, attempting to compute minimal diagnoses. We now characterize flips as valid or invalid based on whether they produce consistent models after the flip.

Definition 12 (Invalid Subset Flip)**.** Given a diagnostic system $\text{DS} = \langle \text{SD}, \text{COMPS}, \text{OBS} \rangle$ and an observation α, the ordered pair $X = \langle \omega, l \rangle$ is defined as an "invalid subset flip" iff ω is a non-minimal diagnosis, $l \in Lit^{-}(\omega)$, and $\text{SD} \wedge \alpha \wedge (\omega \vee l) \wedge l \models \perp$.

We can define an invalid superset flip in an analogous fashion. What we are really interested in are valid flips. We now define a "Valid Subset Flip", and note that a "Valid Superset Flip" has an analogous definition.

Definition 13 (Valid Subset Flip)**.** Given a diagnostic system $\mathrm{DS} = \langle \mathrm{SD}, \mathrm{COMPS}, \mathrm{OBS} \rangle$ an observation α, and a non-minimal diagnosis ω, a valid flip exists if we can perform a subset flip in ω which is not an invalid subset flip.

Given these notions, we can now characterize continuity of the diagnosis search space for weak fault models, as a corollary to Lemma 4. The proof of this corollary follows directly from the Lemma and definitions of valid flips.

Corollary 1. *Given a diagnostic system* $\mathrm{DS} = \langle \mathrm{SD}, \mathrm{COMPS}, \mathrm{OBS} \rangle$*,* $\mathrm{SD} \in \mathbf{WFM}$*, the diagnosis space is continuous, in that the following hold:*

- *starting from a diagnosis with all negative literals,* $\exists$ *a sequence of valid subset flips until we reach a minimal diagnosis;*
- *starting from a minimal diagnosis, there exists a sequence of valid superset flips until we reach a diagnosis with all negative literals.*

We can also characterize the guarantee of finding a minimal diagnosis with Safari in terms of a continuous diagnosis space. Note that this is a sufficient, but not necessary, condition; for example, we may configure Safari to flip multiple literals at a time to circumvent problems of getting trapped in discontinuous diagnosis spaces.

Theorem 2. *Given a diagnostic system* $\mathrm{DS} = \langle \mathrm{SD}, \mathrm{COMPS}, \mathrm{OBS} \rangle$*, and an starting diagnosis* ω*,* Safari *is guaranteed to compute a minimal diagnosis if the diagnosis space is continuous.*

Proof. Given an initial diagnosis ω, Safari attempts to compute a minimal diagnosis by performing subset flips. If the diagnosis space is continuous, then we know that there exists a sequence of valid flips leading to a minimal diagnosis. Hence Safari is guaranteed to find a minimal diagnosis from ω. □

6.2 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.

6.2.1 Basic Model

Consider a diagnostic system $\mathrm{DS} = \langle \mathrm{SD}, \mathrm{COMPS}, \mathrm{OBS} \rangle$ such that $\mathrm{SD} \in \mathbf{WFM}$ 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 $|\mathrm{COMPS}| - k$ negative literals. The probability of the next variable flip being successful equals the probability that the next negative to positive flip out of the $H - k$ negative literals does not conflict with a negative literal belonging to a diagnosis solution ω. Consequently, of the $|\omega| - k$ literals only $\mathrm{COMPS}| - |\omega| - k$ literals are allowed to flip, and therefore the success probability equals:

$$p\left(k\right) = \frac{|\mathrm{COMPS}| - |\omega| - k}{|\mathrm{COMPS}| - k} = 1 - \frac{|\omega|}{|\mathrm{COMPS}| - k} \tag{3}$$

The search process can be modeled in terms of the Markov chain depicted in Fig. 4, where k equals the state of the algorithm. Running into an inconsistency is modeled by the transitions to the state denoted "fail".

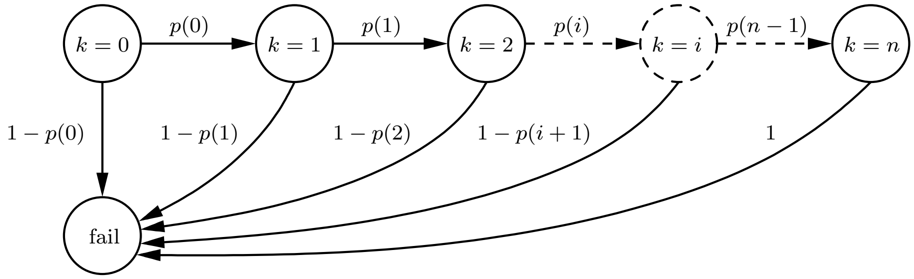


Figure 4: Model of a Safari run for $M = 1$ and a single diagnosis ω ($n = |\mathrm{COMPS}| - |\omega|$)

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

$$f(k) = (1 - p(k+1)) \prod_{i=0}^{k} p(i) \tag{4}$$

After substituting (3) in (4) we receive the pdf of k:

$$f(k) = \frac{|\omega|}{|\mathrm{COMPS}| - k + 1} \prod_{i=0}^{k} \left[1 - \frac{|\omega|}{|\mathrm{COMPS}| - i}\right] \tag{5}$$

At the optimum goal state $k = |\mathrm{COMPS}| - |\omega|$ the failure probability term in (5) 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 $k = |\mathrm{COMPS}| - |\omega|$ 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. 5 shows the pdf for problem instances with $|\text{COMPS}| = 100$ for an increasing fault cardinality $|\omega|$.

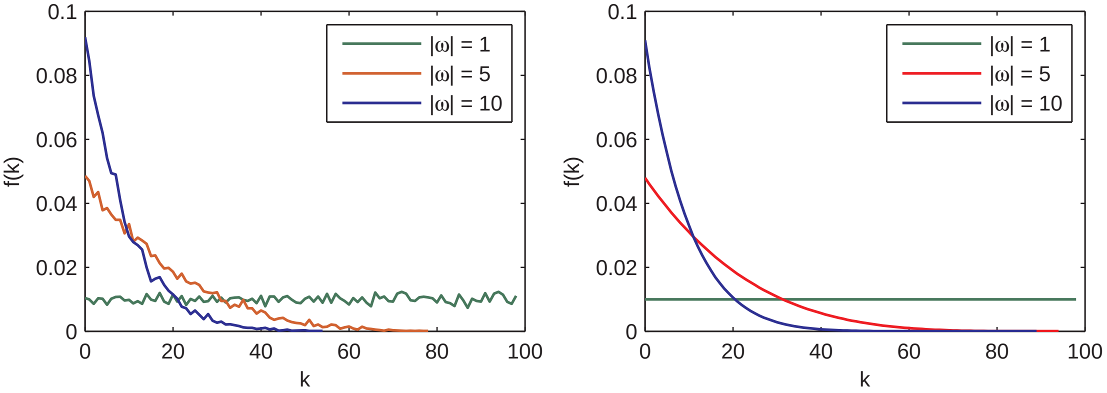


Figure 5: Empirical (left) and analytic (right) $f(k)$ for no retries and a single diagnosis

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.

6.2.2 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 elementary combinatorics we can compute the probability of having a diagnosis after flipping any of M different negative literals at step k. Similar to (3), at stage k there are $|\text{COMPS}| - k$ 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:

$$p'(k) = 1 - \frac{\binom{|\omega|}{M}}{\binom{|\text{COMPS}|-k}{M}} \tag{6}$$

The progress of Safari can be modeled for values of $M > 1$ as a Markov chain, similar to the one shown in Fig. 4 with the transition probability of p replaced by p'. The resulting pdf of the number of successful steps becomes:

$$f'(k) = \frac{\binom{|\omega|}{M}}{\binom{|\text{COMPS}|-k+1}{M}} \prod_{i=0}^{k} \left[1 - \frac{\binom{|\omega|}{M}}{\binom{|\text{COMPS}|-i}{M}}\right] \tag{7}$$

It can be seen that (5) is a private case of (7) for $M = 1$.

The retry effect on the shape of the pdf is profound. Whereas for single-fault solutions the shape for $M = 0$ is uniform, for $M = 1$ most of the probability mass is already located at the optimum $k = |\mathrm{COMPS}| - |\omega|$. Fig. 6 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 $|\mathrm{COMPS}| - |\omega|$.

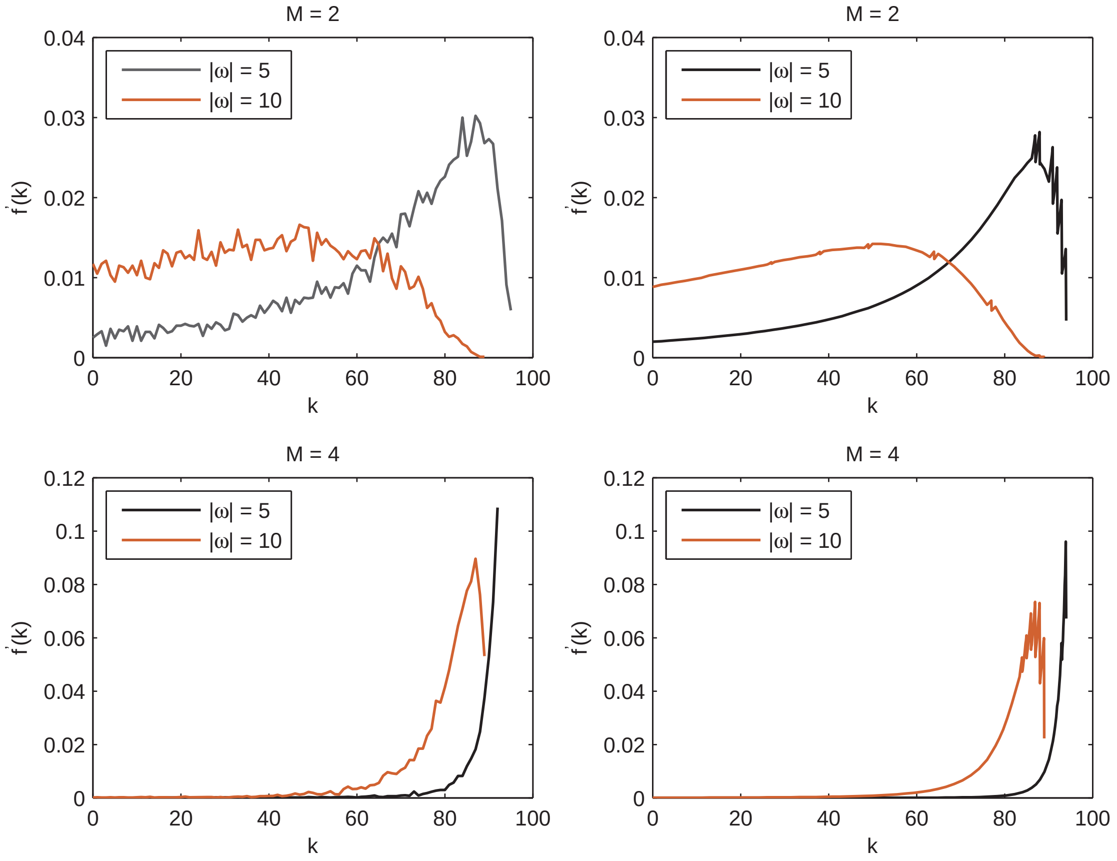


Figure 6: Empirical (left) and analytic (right) $f'(k)$ for multiple retries and a single diagnosis

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, we expect the pdf to have a considerable probability mass located at $k = |\mathrm{COMPS}| - |\omega|$, depending on M relative to $|\mathrm{COMPS}|$.

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. In what follows we will discuss how this fact influences the performance model of Safari.

6.2.3 Solution Multiplicity

In the preceding text of this section we have considered an observation leading to a single minimal diagnosis. Although this is a desirable situation for real-world systems (diagnostic results containing one minimal solution only maximize the diagnostic precision), it is often the case that multiple minimal diagnoses exist due to uncertainty in the model or in the observation. More solutions implies that the search success is prolonged due to the fact that although a change of the sign of literal may lead to a failure for one particular diagnosis, for other diagnoses the flip may prove consistent.

As the success criterion of the algorithm is the logical disjunction of the success criteria per individual diagnosis solution, the number of successful flips k becomes the *maximum* of the individual k outcomes per single diagnosis solution. Let k_s denote the value k would have reached in case of a single diagnosis s, $s = 1, \ldots, S$. Then the value k will reach in case of S non-overlapping diagnoses is simply:

$$k = \overset{s=1}{\max_{s}} k_s \tag{8}$$

The effect of the multiplicity of the diagnostic solution space on the pdf of k is that the resulting pdf is that of the highest order statistic (maximum) of the group of S independent, identically distributed (iid) variates k_s, each with a pdf $f(k)$. Again, this implies another shift of probability mass to the region of the optimal solution. Let $g(k)$ denote the pdf of the maximum of the k_s with pdf $f(k)$ as given by (7). Let $G(k)$ and $F(k)$ denote the *cdf* of g and f, respectively. Since the cdf of the highest order statistic of S iid variates is given by $G(k) = F(k)^S$, it follows that

$$g(k) = \frac{dF(k)^S}{dk} \tag{9}$$

The left plot of Fig. 7 shows empirically $g(k)$ for a number of previous synthetic problem instances with increasing S while the right plot from the same figure compares the effect of M and S on $g(k)$.

Note that in the above derivation we have assumed the existence of a set of S non-overlapping diagnoses. In reality, this assumption cannot be made. In practice, however, the influence of this approximation is small, as from Fig. 7 it is readily seen that the order-statistical influence of S on the shape of the *pdf* of k is far smaller than the effect of M. As our analysis is aimed to provide a basic understanding into the workings of the Safari algorithm, we refrain from a more thorough investigation into the effect of solution multiplicity.

6.2.4 Optimality of Safari in Strong-Fault Models

From the above analysis we have seen that in **WFM** it is easy, starting from a non-minimal diagnosis, to reach a minimal diagnosis. As will be discussed in more detail below, this is not necessarily the case for strong-fault models. In many practical cases, however, strong-fault models exhibit, at least partially, behavior similar to MDH, thus allowing greedy algorithms like Safari to achieve optimal or near-optimal results. In what follows we will restrict our attention to a large subclass of **SFM** which is of great practical significance (Struss & Dressler, 1992), **SFSM**.

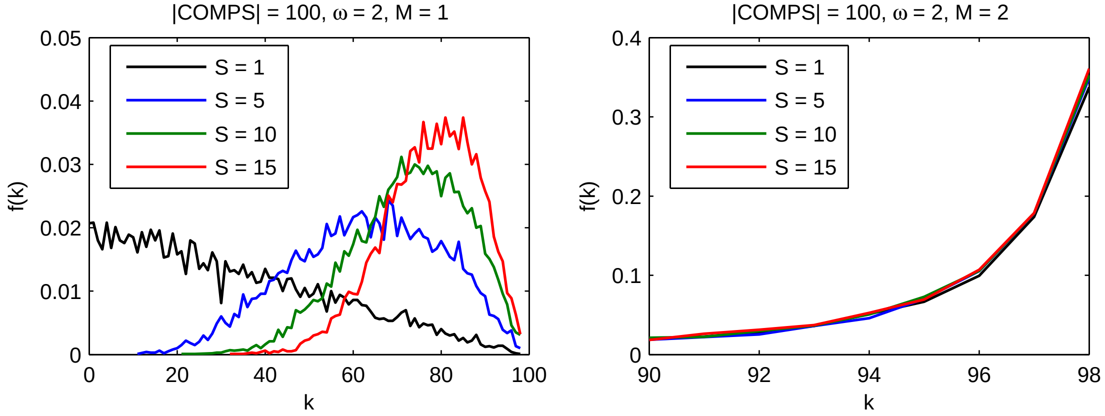


Figure 7: $g(k)$ for various S (left) and comparison of the effects of M and S (right)

Definition 14 (Strong-Fault Stuck-At Model)**.** A system $\mathrm{DS} = \langle \mathrm{SD}, \mathrm{COMPS}, \mathrm{OBS} \rangle$ belongs to the class **SFSM** iff SD is in the form $(h_1 \Rightarrow F_1) \wedge (\neg h_1 \Rightarrow l_1) \wedge \cdots \wedge (h_n \Rightarrow F_n) \wedge (\neg h_n \Rightarrow l_n)$ such that $1 \leq i, j \leq n$, $\{h_i\} \subseteq \mathrm{COMPS}$, $F_j \in \mathbf{Wff}$, none of h_i appears in F_j, and l_j is a positive or negative literal in F_j.

Before we analyze the optimality of Safari with **SFSM**, we first illustrate the process of computing minimal diagnoses with such a model by continuing the running example started in Sec. 3.

A Simple Example (Continued) First, we will create a system description SD_{sa} for a **SFSM** model. Let $\mathrm{SD}_{sa} = \mathrm{SD}_w \wedge \mathrm{SD}_f$, where SD_w is given by Equation (1). The second part of SD_{sa}, the strong fault description SD_f, specifies that the output of a faulty gate must be stuck-at-1:

$$\begin{aligned}\mathrm{SD}_f = & (\neg h_1 \Rightarrow i) \wedge (\neg h_2 \Rightarrow d) \wedge (\neg h_3 \Rightarrow j) \wedge (\neg h_4 \Rightarrow m) \wedge (\neg h_5 \Rightarrow b) \wedge \\ & \wedge (\neg h_6 \Rightarrow l) \wedge (\neg h_7 \Rightarrow k)\end{aligned} \tag{10}$$

A key idea to analyzing **SFSM** models is to "split" the diagnosis into "weak fault" and "strong fault" parts. As a result we will be diagnosing two "simultaneous" diagnostic problems: $\mathrm{DS}_w = \langle \mathrm{SD}_w, \mathrm{COMPS}, \mathrm{OBS} \rangle$ and $\mathrm{DS}_f = \langle \mathrm{SD}_f, \mathrm{COMPS}, \mathrm{OBS} \rangle$ with the same observation α_1.

It is clear that $\mathrm{SD}_{sa} \in \mathbf{SFSM}$. We will next compute the diagnoses of $\mathrm{SD}_{sa} \wedge \alpha_1$ (cf. Sec. 3 for the value of α_1). There is one minimal diagnosis of $\mathrm{SD}_{sa} \wedge \alpha_1$ and it is $\omega_5^{\subseteq} = \neg h_1 \wedge h_2 \wedge h_3 \wedge \cdots \wedge h_7$. If we choose the two literals h_3 and h_4 from $\omega_5^{\subseteq}$ and change the signs of h_3 and h_4, we create two new health assignments: $\omega_{15} = \neg h_1 \wedge h_2 \wedge \neg h_3 \wedge h_4 \wedge h_5 \wedge h_6 \wedge h_7$ and $\omega_{16} = \neg h_1 \wedge h_2 \wedge h_3 \wedge \neg h_4 \wedge h_5 \wedge h_6 \wedge h_7$. It can be checked that both ω_{15} and ω_{16} are diagnoses, i.e., $\mathrm{SD}_{sa} \wedge \alpha_1 \wedge \omega_{15} \not\models \perp$ and $\mathrm{SD}_{sa} \wedge \alpha_1 \wedge \omega_{16} \not\models \perp$. The fact that ω_{15} and ω_{16} are diagnoses follows from (1) the fact that ω_{15} and ω_{16} are diagnoses of $\mathrm{SD}_w \wedge \alpha_1$ (this is true because of MDH and the fact that $\omega_5^{\subseteq}$ is a minimal diagnosis of $\mathrm{SD}_w \wedge \alpha_1$) and (2) the strong part of the model SD_s is consistent with the values of the internal variables which must be modified to ensure a consistent global assignment once we negate h_3 and h_4; these internal variables are j and m, respectively. Equivalently, if negating h_3 in $\omega_5^{\subseteq}$, which makes j stuck-at-1, results in a diagnosis, and negating h_4 in $\omega_5^{\subseteq}$, which makes m stuck-at-1, also results in a diagnosis, negating *both* h_3 and h_4 in $\omega_5^{\subseteq}$ will also result in a diagnosis (consider the fact that the fault mode of h_4 sets m only, but does not impose constraints on j).

The above argument can be extended inductively to h_5, h_6, and h_7. Hence, for any assignment of COMPS containing $\neg h_1 \wedge h_2$ is a diagnosis of $\text{SD}_{sa} \wedge \alpha_1$, no matter what combination of signs we take for h_3, h_4, h_5, h_6, and h_7.

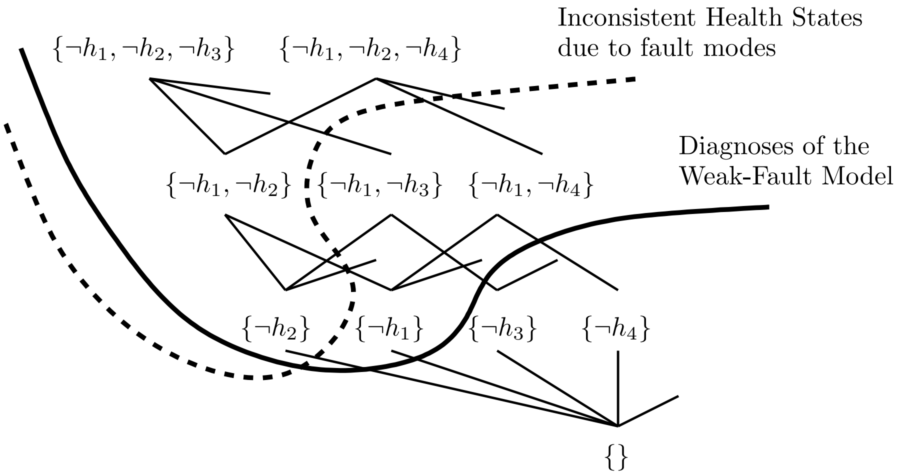

$\{\neg h_1, \neg h_2, \neg h_3\}$
$\{\neg h_1, \neg h_2, \neg h_4\}$
Inconsistent Health States due to fault modes
Diagnoses of the Weak-Fault Model
$\{\neg h_1, \neg h_2\}$
$\{\neg h_1, \neg h_3\}$
$\{\neg h_1, \neg h_4\}$
$\{\neg h_2\}$
$\{\neg h_1\}$
$\{\neg h_3\}$
$\{\neg h_4\}$
$\{\}$

Figure 8: Part of the diagnosis space for $\text{SD}_{sa} \wedge \alpha_1$

What is more important than computing the set of minimal diagnoses of DS_{sa} and α_1 is that we have seen a mechanism of how the weak portion SD_w of SD_{sa}, an observation α, and a diagnosis ω assign unique values to all internal variables which must "agree" with the "stuck-at" values from the "strong" part SD_f. This, combined with the fact that "stuck-at" faults in different components are independent of each other (they can create contradictions with $\text{SD}_w \wedge \alpha \wedge \omega$ only but not amongst themselves), results in one minimal diagnosis $\omega_5^{\subseteq}$ with every health assignment containing $\neg h_1 \wedge h_2$ being also a diagnosis. Part of the diagnosis space is shown in Fig. 8. The search space defined in this way is still continuous from the viewpoint of Safari. Before we formalize these notions in the general case and draw conclusions about the optimality of Safari, we will see that this is not necessarily the case for strong-fault models not members of **SFSM**.

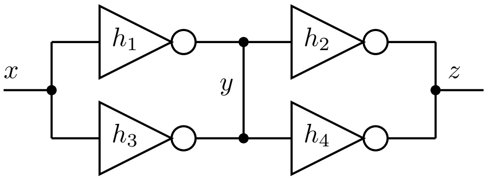


Figure 9: A four inverters circuit

An Example of a Discontinuous Diagnosis Space There exist strong-fault models that can impose arbitrary difficulty to Safari, leading to suboptimal diagnoses of any cardinality. Consider for example, the Boolean circuit shown in Fig. 9 and modeled by the propositional formula

$$\mathrm{SD}_d = \begin{cases} [h_i \Rightarrow (y \Leftrightarrow \neg x)] \wedge [\neg h_i \Rightarrow (y \Leftrightarrow x)]\,, & \text{for } i = 1 \text{ and } i = 3 \\ [h_i \Rightarrow (z \Leftrightarrow \neg y)] \wedge [\neg h_i \Rightarrow (z \Leftrightarrow y)]\,, & \text{for } i = 2 \text{ and } i = 4 \end{cases} \tag{11}$$

and an observation $\alpha_d = x \wedge z$. There are exactly two diagnoses of $\mathrm{SD}_d \wedge \alpha_d$: $\omega_{15} = h_1 \wedge h_2 \wedge h_3 \wedge h_4$ and $\omega_{16} = \neg h_1 \wedge \neg h_2 \wedge \neg h_3 \wedge \neg h_4$. As only ω_{15} is minimal, $|\omega_{15}| = 0$, and $|\omega_{16}| = 4$, if the algorithm starts from ω_{16} it is not possible to reach the minimal diagnosis ω_{15} by performing single flips. Similarly we can construct models which impose an arbitrarily bad bound on the optimality of Safari. Such models, however, are not common and we will see that the greedy algorithm performs well on a wide class of strong-fault models.

Optimality Guarantee for Minimal Diagnosis in Strong-Fault Stuck-At Models The theorem that follows contains an important results for **SFSM** and is similar to MDH for a strong-fault stuck-at models. An important consequence of it is that the set of minimal diagnoses of a system description SD $\in$ **SFSM** characterizes all diagnoses of SD. This paper uses the theorem for analyzing the optimality of Safari, since it shows that the underlying diagnosis search space is continuous.

Theorem 3 (**SFSM** MDH)**.** *Consider a system* $\mathrm{DS} = \langle \mathrm{SD}, \mathrm{COMPS}, \mathrm{OBS}\rangle$*,* $\mathrm{SD} \in \mathbf{SFSM}$*, an observation* α*, and a minimal diagnosis* $\omega^{\subseteq}$ *of* $\mathrm{SD} \wedge \alpha$*. The diagnosis space for* (DS, α) *is continuous.*

Proof (Sketch). We prove this theorem by first partitioning COMPS such that COMPS $= W \cup S$, $W \cap S = \emptyset$, and then showing that a health assignment $\tilde{\omega}$ is a diagnosis of $\mathrm{SD} \wedge \alpha$ iff $Lit^-(\tilde{\omega}_w) \supseteq Lit^-(\omega^{\subseteq}_w)$ and $\tilde{\omega}_s = \omega^{\subseteq}_s$, where $\omega^{\subseteq}_w$ contains variables from W only and $\omega^{\subseteq}_s$ contains variables from S only. We start by reordering the conjunctions in the general **SFSM** representation given by Def. 14:

$$\mathrm{SD}_{sa} = \begin{cases} (h_1 \Rightarrow F_1) \wedge (h_2 \Rightarrow F_2) \wedge \cdots \wedge (h_n \Rightarrow F_n) \\ (\neg h_1 \Rightarrow l_1) \wedge (\neg h_2 \Rightarrow l_2) \wedge \cdots \wedge (\neg h_n \Rightarrow l_n) \end{cases} \tag{12}$$

Next SD_{sa} is split in two system descriptions, which have disjoint sets of clauses, but which are built on the same set of literals: $\mathrm{SD}_{sa} = \mathrm{SD}_w \wedge \mathrm{SD}_s$, $\mathrm{SD}_w = (h_1 \Rightarrow F_1) \wedge (h_2 \Rightarrow F_2) \wedge \cdots \wedge (h_n \Rightarrow F_n)$, and $\mathrm{SD}_s = (\neg h_1 \Rightarrow l_1) \wedge (\neg h_2 \Rightarrow l_2) \wedge \cdots \wedge (\neg h_n \Rightarrow l_n)$. Consider an observation α and a minimal diagnosis $\omega^{\subseteq}$ of $\mathrm{SD}_w \wedge \alpha$. If we assume that we have a well-formed circuit, and all inputs and outputs observed, α and $\omega^{\subseteq}$ assign unique values to all internal variables[4] in SD_w (respectively SD_{sa}). We further assume that there exists at least one minimal-diagnosis $\omega^{\subseteq}$ of $\mathrm{SD}_w \wedge \alpha$, such that the "faulty" literals of $\omega^{\subseteq}$ assign exactly the "stuck-at" values implied by $\mathrm{SD}_s \wedge \alpha$ (otherwise $\mathrm{SD}_{sa} \wedge \alpha \models \perp$). This $\omega^{\subseteq}$, then, is also a minimal-diagnosis of $\mathrm{SD}_{sa} \wedge \alpha$.

Next, we start from the minimal diagnosis $\omega^{\subseteq}$ and do a proof by contradiction to show that there can be no strong fault diagnosis leading to a discontinuous diagnosis space.

We assume that the diagnosis space for the strong fault model is discontinuous. From Corollary 1, we know that the diagnosis space for SD_w is continuous. Hence there must

[4] An internal variable is any variable v such that $v \notin \mathrm{OBS} \cup \mathrm{COMPS}$.

be some set of valid superset flips that leads to the weak-fault diagnosis where all literals in the diagnosis are faulty (negative). We can call this sequence of diagnoses $\mathcal{D} = \{\omega_1^{\subseteq}, \omega_2^{\subseteq}, \ldots, \omega_m^{\subseteq}\}$. First we can take some $\omega_i^{\subseteq} \in \mathcal{D}$, and assume that the corresponding strong-fault diagnosis is inconsistent (i.e., the diagnosis space for the strong fault model is discontinuous).

By definition, $\omega_i^{\subseteq}$ is consistent with $\mathrm{SD} \wedge \alpha$; further, we know that in the **WFM** model, all internal variables will be assigned values. Since $\mathrm{SD}_w = (h_1 \Rightarrow F_1) \wedge (h_2 \Rightarrow F_2) \wedge \cdots \wedge (h_n \Rightarrow F_n)$, all the $F_i (i = 1, 2, \ldots, n)$ must be assigned consistent values. In other words, we can take $\mathcal{F} = \bigwedge_{i=1}^{n} F_i$ (since these are forced consistent values) such that $\mathcal{F}$ is consistent with $\mathrm{SD} \wedge \alpha$. This gives exactly the definition of a consistent diagnosis in the strong fault model, since $\mathrm{SD}_s \subseteq \mathrm{SD}$. But by our prior assumption, this strong fault diagnosis was inconsistent, meaning that we have a contradiction. □

The above theorem allows us to evaluate the optimality of Safari with **SFSM**.

Corollary 2. *Given a diagnostic system* $\mathrm{DS} = \langle \mathrm{SD}, \mathrm{COMPS}, \mathrm{OBS} \rangle$*, an observation* α*, and* $\mathrm{SD} \in$ **SFSM***, the greedy stochastic algorithm can be configured to compute a minimal diagnosis.*

Proof (Sketch). The idea is that starting from any arbitrary diagnosis ω, as a consequence of Theorem 3, ω can be split into $\omega = \omega_s \wedge \omega_w$ where ω_s would be fixed by the observation α and SD_s. What remains is to "flip" the negative literals in ω_w and, by trying all negative literals in ω, we are guaranteed to find the ones which appear in ω_w. The greedy algorithm is then guaranteed to reach an optimum (minimal diagnosis) in a way similar to the one given in Proposition 1. □

Performance Modeling with Stuck-At Models To further characterize the optimality of Safari in strong-fault models, we first define a case in which the algorithm cannot improve a non-minimal diagnosis by changing the sign of a faulty literal. Note that the existence of such cases is not a sufficient condition for Safari to be suboptimal, as it is possible to reach a minimal diagnosis by first changing the sign of some other faulty literal, thus "circumventing" the missing diagnosis.

From the preceding section we know that the probability of encountering an "invalid flip" is constant throughout the search (it is determined by the observation vector and the fault modes). The probability of Safari to progress from any non-minimal diagnosis becomes

$$p(k) = 1 - \frac{\binom{|\omega|+|X|}{M}}{\binom{|\mathrm{COMPS}|-k}{M}} \tag{13}$$

where $|X|$ is the number of "invalid flips" in stage k. The ratio of the number of "invalid flips" to the cardinality of the non-minimal diagnoses we will call **SFM** density d. Alternatively, the probability of success of Safari is:

$$p(k) = 1 - \frac{\binom{|\omega|}{M}}{\binom{|\mathrm{COMPS}|-k}{M}} - d \tag{14}$$

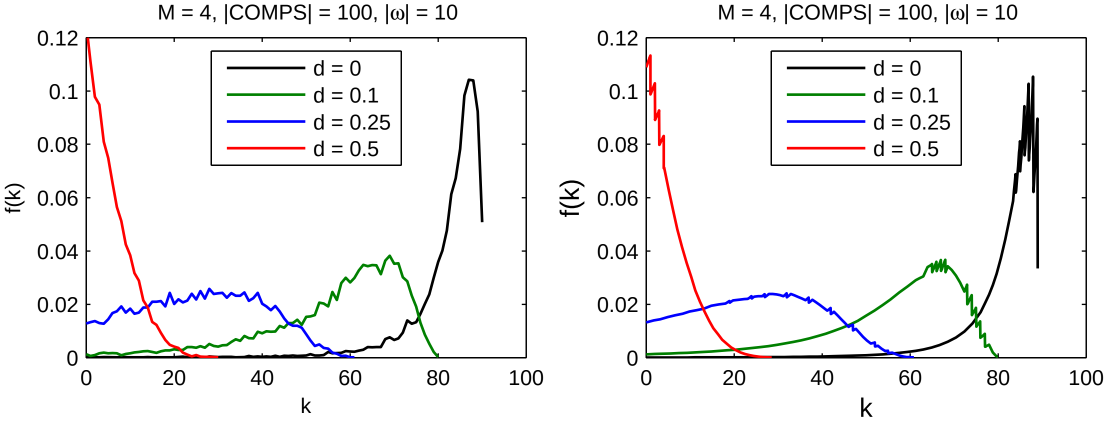


Figure 10: Empirical (left) and analytic (right) $f'(k)$ for various diagnostic densities, multiple retries and a single diagnosis

Plugging p into (4) allows us to predict $f(k)$ for the **SFM** models for which our assumptions hold. This pdf, both measured from an implementation of Safari and generated from (4) and (14) is shown in Fig. 10 for different values of the density d.

From Fig. 10 it is visible that increasing the density d leads to a shift of the probability density of the length of the walk k to the left. The effect however is not that profound even for large values of d and is easily compensated by increasing M as discussed in the preceding sections.

It is interesting to note that d can be computed from the system description and the algorithm and can be further used for increasing the performance of Safari.

6.2.5 Validation

In the preceding sections we have illustrated the progress of Safari with synthetic circuits exposing specific behavior (diagnoses). In the remainder of this section we will plot the pdf of the greedy search on one of the small benchmark circuits (for more information on the 74181 model cf. Sec. 7).

The progress of Safari with a weak-fault model of the 74181 circuit is shown in Fig. 11. We have chosen a difficult observation leading to a minimal diagnosis of cardinality 7 (left) and an easy observation leading to a single fault diagnosis (right). Both plots show that the probability mass shifts to the right when increasing M and the effect is more profound for the smaller cardinality.

The effect of the density d on the probability of success of Safari is shown in Fig. 12.

Obviously, in this case the effect of increasing M is smaller, although still depending on the difficulty of the observation vector. Last, even for small values of M, the absolute probability of Safari finding a minimal diagnosis is sizeable, allowing the use of Safari as a practical *anytime* algorithm which always returns a diagnosis, the optimality of which depends on the time allocated to its computation.

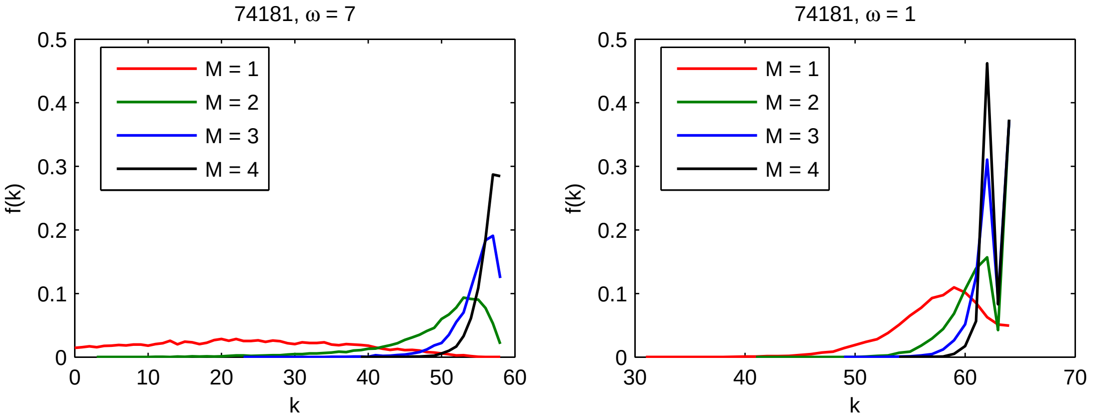


Figure 11: $f(k)$ for a weak-fault model of the 74181 circuit with observations leading to two different minimal-cardinality diagnoses and various M

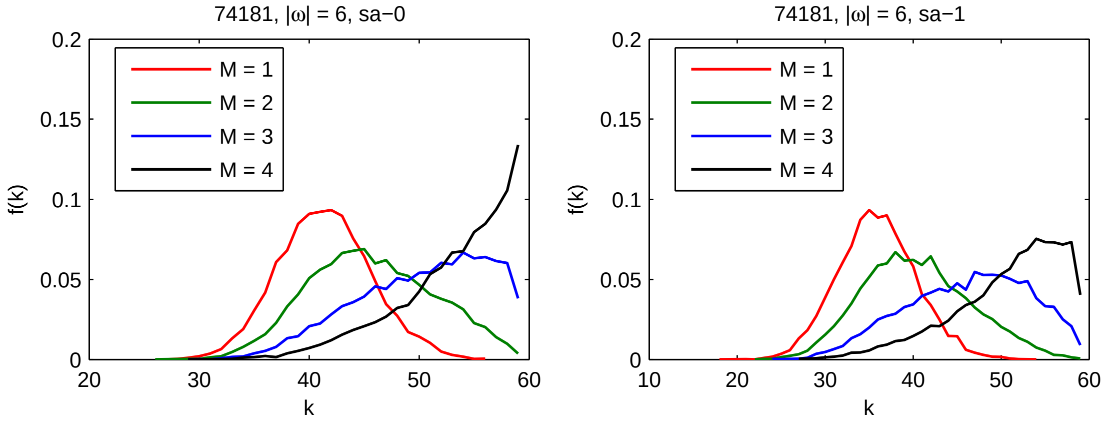


Figure 12: $f(k)$ for two strong-fault models of the 74181 circuit with various M

Next we will continue our argument with experimenting on bigger models.

7. Experimental Results

This section discusses empirical results measured from an implementation of Safari. For the experiments, we have performed a total of 248 820 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, Safari, due to its randomized character, has been run 10 times on each model and observation vector.

7.1 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.[5]

Traditionally, MBD algorithms have been tested on diagnostic models of digital circuits like the ones included in the ISCAS85 benchmark suite (Brglez & 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, & Hayes, 1999). In order to provide both weak- and strong-fault cases, we have translated each circuit to a weak, stuck-at-0 (S-A-0), and stuck-at-1 (S-A-1) model. In the stuck-at models, the output of each faulty gate is assumed to be the same constant (cf. Def. 14).

Table 2: An overview of the 74XXX/ISCAS85 benchmark circuits

		Variables		Observations		
Name	Description	|OBS|	|COMPS|	Weak	S-A-0	S-A-1
74182	4-bit carry-lookahead generator	14	19	250	150	82
74L85	4-bit magnitude comparator	14	33	150	58	89
74283	4-bit adder	14	36	202	202	202
74181	4-bit ALU	22	65	350	143	213
c432	27-channel interrupt controller	43	160	301	301	301
c499	32-bit SEC circuit	73	202	835	235	835
c880	8-bit ALU	86	383	1 182	217	335
c1355	32-bit SEC circuit	73	546	836	836	836
c1908	16-bit SEC/DEC	58	880	846	846	846
c2670	12-bit ALU	373	1 193	1 162	134	123
c3540	8-bit ALU	72	1 669	756	625	743
c5315	9-bit ALU	301	2 307	2 038	158	228
c6288	32-bit multiplier	64	2 416	404	274	366
c7552	32-bit adder	315	3 512	1 557	255	233

The performance of diagnostic algorithms depends to a various extent on the observation vectors. Hence, we have performed our experimentation with a number of different observations for each model. The generation of these observation vectors is a topic on its own (Feldman, Provan, & van Gemund, 2007). These observations lead to diagnoses of different minimal-cardinality, 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 2 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,

[5] Lydia, Safari, and the diagnostic benchmark can be downloaded from `http://fdir.org/lydia/`.

which characterize the size of the circuits. The next three columns show the number of observation vectors with which we have tested the weak, S-A-0, and S-A-1 models. For the stuck-at models, we have chosen these weak-fault model observations which are consistent with their respective system descriptions (while in strong-fault models it is often the case that $\mathrm{SD} \wedge \alpha \models \perp$, we have not considered such scenarios).

7.2 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, & Somenzi, 2005). 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 & 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 3.

Table 3: Model count and time for counting

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	$> 5.6283 \times 10^{15}$	$> 3\,600$

The second column of Table 3 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.

A satisfiability-based method for diagnosing an optimized version of ISCAS85 has been used by (Smith, Veneris, & Viglas, 2004). In a recent paper (Smith, Veneris, Ali, & Viglas, 2005), the SAT-based approach has been replaced by Quantified Boolean Formula (QBF) solver for computing multiple-fault diagnoses. These methods report good absolute performance for single and double-faults (and we believe that they scale well for higher cardinalities), but require modifications of the initial circuits (i.e., introduce cardinality and test constraints) and suggest specialized heuristics for the SAT solvers in order to improve the search performance. Comparison of the performance of Safari to the timings reported by these papers would be difficult due to a number of reasons like the use of different and optimized benchmark sets, trading-off memory for speed, rewriting the original circuits, etc.

7.3 Comparison to Complete Algorithms

Table 4 shows the results from comparing Safari to implementations of two state-of-the-art complete and deterministic diagnostic algorithms: a modification for completeness of CDA* (Williams & Ragno, 2007) and HA* (Feldman & van Gemund, 2006).

Table 4: Comparison of CDA*, HA*, and Safari [% of tests solved]

	CDA*			HA*			Safari		
Name	Weak	S-A-0	S-A-1	Weak	S-A-0	S-A-1	Weak	S-A-0	S-A-1
74182	100	100	100	100	100	100	100	100	100
74L85	100	100	100	100	100	100	100	100	100
74283	100	100	100	100	100	100	100	100	100
74181	79.1	98.6	97.7	100	100	100	100	100	100
c432	74.1	75.4	73.1	71.1	94.7	69.1	100	100	100
c499	29	45.5	27.7	24.1	77.9	25.9	100	100	100
c880	11.6	44.7	32.2	12.4	62.2	41.5	100	100	100
c1355	3.8	4.7	5.4	10.8	10.6	12.2	100	100	100
c1908	0	0	0	6.1	6	6.5	100	100	100
c2670	0	0	0	5	64.2	44.7	100	100	100
c3540	0	0	0	1.1	3.8	2.2	100	100	100
c5315	0	0	0	1.1	8.2	5.7	100	100	100
c6288	0	0	0	3.5	5.1	3.3	100	100	100
c7552	0	0	0	3.9	7.8	12	100	100	100

Table 4 shows, for each model and for each algorithm, the percentage of all tests for which a diagnosis could be computed within 1 min cut-off time.

As it is visible from the three rightmost columns of Table 4, Safari could find diagnoses for all observation vectors, while the performance of the two deterministic algorithms (columns two to seven) degraded with the increase of the model size. 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.

7.4 Performance of the Greedy Stochastic Search

Table 5 shows the absolute performance of Safari. This varies from under a millisecond for the small models, to approx. 30 s for the largest strong-fault model. These fast absolute times show that Safari is suitable for on-line reasoning tasks, where autonomy depends on speedy computation of diagnosis.

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 theoretic results 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.

Table 5: Performance of Safari [ms]

	Weak		S-A-0		S-A-1	
Name	$t_{\min}$	$t_{\max}$	$t_{\min}$	$t_{\max}$	$t_{\min}$	$t_{\max}$
74182	0.41	1.25	0.39	4.41	0.40	0.98
74L85	0.78	7.47	0.72	1.89	0.69	4.77
74283	0.92	4.84	0.88	3.65	0.92	5.2
74181	2.04	6.94	2.13	22.4	2.07	7.19
c432	8.65	38.94	7.58	30.59	7.96	38.27
c499	14.19	31.78	11.03	30.32	10.79	31.11
c880	48.08	88.87	37.08	80.74	38.47	81.34
c1355	95.03	141.59	76.57	150.29	83.14	135.29
c1908	237.77	349.96	196.13	300.11	217.32	442.91
c2670	500.54	801.12	646.95	1 776.72	463.24	931.8
c3540	984.31	1 300.98	1 248.5	2 516.46	976.56	2 565.18
c5315	1 950.12	2 635.71	3 346.49	7 845.41	2 034.5	4 671.17
c6288	2 105.28	2 688.34	2 246.84	3 554.4	1 799.18	2 469.48
c7552	4 557.4	6 545.21	9 975.04	32 210.71	5 338.97	12 101.61

7.5 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 these observations which lead to single and double faults we have evaluated the average optimality of Safari. Table 6 shows these optimality results for the greedy search. The second column of Table 6 shows the number of observation vectors leading to single faults for each weak-fault model. The third column shows the average cardinality of Safari. The second and third column are repeated for the S-A-0 and S-A-1 models, and then, all the six columns are repeated for double faults.

Table 6 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 Sec. 6.

With strong-fault models results are close to optimal for the small models and the quality of diagnosis deteriorates for c3540 and bigger. This is not surprising considering the modest number of retries and number of “flips” with which Safari was configured.

8. 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.

Table 6: Optimality of Safari [average cardinality]

	Single Faults						Double Faults					
	Weak		S-A-0		S-A-1		Weak		S-A-0		S-A-1	
Name	#	Card.	#	Card.	#	Card.	#	Card.	#	Card.	#	Card.
74182	50	1	37	1	40	1	50	2	38	2	18	2
74L85	50	1.04	18	1.02	40	1.03	50	2.12	17	2.06	35	2.07
74283	50	1.08	34	1.59	46	1.88	50	2.2	45	2.41	42	2.6
74181	50	1.19	36	2.81	46	2.6	50	2.25	36	3.61	43	3.16
c432	58	1.19	52	1.06	37	1.04	82	2.46	80	2.25	48	2.15
c499	84	1.49	53	1.49	84	1.01	115	3.27	34	3.01	115	2.03
c880	50	1	39	1.1	40	1.05	50	2.01	34	2.14	35	2.07
c1355	84	1.66	82	1	84	1.02	6	2.15	7	2	18	2.07
c1908	52	1.05	49	2.91	52	4.79	–	–	2	3	3	3.17
c2670	29	1.03	39	1.77	28	2.06	13	2.12	24	2.78	15	3.27
c3540	8	1.01	23	2.5	16	3.74	–	–	1	4.9	–	–
c5315	14	1	9	3.54	12	5.4	7	2	3	3.7	1	3.8
c6288	13	1	13	28.83	12	28.68	1	2	1	27	–	–
c7552	27	1.01	11	17.37	18	23.38	16	2	4	18.5	6	27.53

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

Abdelbar, A. M. (2004). Approximating cost-based abduction is np-hard. *Artificial Intelligence*, *159*(1-2), 231–239.

Abdelbar, A. M., Gheita, S. H., & Amer, H. A. (2006). Exploring the fitness landscape and the run-time behaviour of an iterated local search algorithm for cost-based abduction. *J. Exp. Theor. Artificial Intelligence*, *18*(3), 365–386.

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

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

Bryant, R. E. (1992). Symbolic boolean manipulation with ordered binary-decision diagrams. *ACM Comput. Surv.*, *24*(3), 293–318.

Bylander, T., Allemang, D., Tanner, M., & Josephson, J. (1991). The computational complexity of abduction. *Artificial Intelligence*, *49*, 25–60.

Charniak, E., & Shimony, S. E. (1994). Cost-based abduction and map explanation. *Artificial Intelligence*, *66*(2), 345–374.

Darwiche, A. (1998). Model-based diagnosis using structured system descriptions. *JAIR*, *8*, 165–222.

de Kleer, J. (1986). An assumption-based TMS. *Artificial Intelligence*, *28*(2), 127–162.

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

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

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

Feldman, A., Provan, G., & van Gemund, A. (2007). Generating manifestations of max-fault min-cardinality diagnoses. In *Proc. DX'07*.

Feldman, A., & van Gemund, A. (2006). A two-step hierarchical algorithm for model-based diagnosis. In *Proc. AAAI'06*.

Forbus, K., & de Kleer, J. (1993). *Building Problem Solvers*. MIT Press.

Freuder, E. C., Dechter, R., Ginsberg, B., Selman, B., & Tsang, E. P. K. (1995). Systematic versus stochastic constraint satisfaction. In *Proc. IJCAI 95*, Vol. 2.

Friedrich, G., Gottlob, G., & Nejdl, W. (1990). Physical impossibility instead of fault models. In *Proc. AAAI*.

Garey, M. R., & Johnson, D. S. (1990). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co.

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

Hermann, M., & Pichler, R. (2007). Counting complexity of propositional abduction. In *IJCAI*, pp. 417–422.

Hoos, H. (1999). SAT-encodings, search space structure, and local search performance. In *Proc. IJCAI'99*, pp. 296–303.

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

Kask, K., & Dechter, R. (1999). Stochastic local search for Bayesian networks. In *Proc. AISTAT'99*.

Kean, A., & Tsiknis, G. K. (1993). Clause management systems. *Computational Intelligence*, *9*, 11–40.

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

McAllester, D. (1990). Truth maintenance. In *Proc. AAAI'90*, Vol. 2.

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

Roth, D. (1996). On the hardness of approximate reasoning. *Artificial Intelligence*, *82*(1-2), 273–302.

Santos Jr., E. (1994). A linear constraint satisfaction approach to cost-based abduction. *Artificial Intelligence*, *65*(1), 1–28.

Smith, A., Veneris, A., Ali, M. F., & Viglas, A. (2005). Fault diagnosis and logic debugging using boolean satisfiability. *IEEE Trans. on CAD of Integrated Circuits and Systems*, *24*(10), 1606–1621.

Smith, A., Veneris, A., & Viglas, A. (2004). Design diagnosis using boolean satisfiability. In *Proc. ASP-DAC'04*, pp. 218–223.

Struss, P., & Dressler, O. (1992). “Physical negation” - integrating fault models into the General Diagnostic Engine. In *Readings in Model-Based Diagnosis*, pp. 153–158. Morgan Kaufmann Publishers Inc.

Vatan, F., Barrett, A., James, M., Williams, C., & Mackey, R. (2003). A novel model-based diagnosis engine: Theory and applications. In *IEEE Aerospace Conf.*

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