A Greedy Stochastic Search Algorithm for Model-Based Diagnosis

Alexander Feldman[1] and **Gregory Provan**[2] and **Arjan van Gemund**[1]
[1]Delft University of Technology
Faculty of Electrical Engineering, Mathematics and Computer Science
Mekelweg 4, 2628 CD, Delft, The Netherlands
Tel.: +31 15 2781935, Fax: +31 15 2786632, e-mail: {a.b.feldman,a.j.c.vangemund}@tudelft.nl
[2]University College Cork, Department of Computer Science, College Road, Cork, Ireland
Tel: +353 21 4901816, Fax: +353 21 4274390, e-mail: g.provan@cs.ucc.ie

Abstract

To date, all algorithms for computing diagnoses within a model-based diagnosis framework have been deterministic. However, in SAT significant speedups have been achieved through a variety of randomized algorithms, such as randomized local search and random restarts. We design an algorithm inspired by randomized satisfiability algorithms to computing diagnoses within a model-based diagnosis framework. We have applied this algorithm to the 74XXX and ISCAS-85 suites of benchmark combinatorial circuits, demonstrating order-of-magnitude speedup over a well-known deterministic algorithm, CDA*, for multiple-fault diagnoses.

Introduction

While model-based diagnosis provides a sound and complete approach to enumerating multiple-fault diagnoses, the biggest challenge (and impediment to industrial deployment) is the computational complexity of the problem. The Model-Based Diagnosis (MBD) problem of isolating multiple-fault diagnoses is known to be NP-complete (Bylander *et al.* 1991; Friedrich, Gottlob, & Nejdl 1990); further, the task of finding a kernel diagnosis of minimal cardinality is $\Pi_2^{\mathbf{P}}$-complete (Eiter & Gottlob 1995). While heuristic methods have been successful in improving the diagnostic search performance, the complexity problem remains a major challenge to MBD.

To overcome this complexity problem, we focus on a novel approach, stochastic algorithms for multiple-fault diagnosis. Local search methods have been successful in solving an important class of satisfiability (SAT) problems and the MBD algorithm we propose in this paper uses a similar approach. The MBD problem, however, is different than SAT in that it is an optimization problem, i.e., one typically wants to find a minimal diagnosis, using some minimality criterion such as minimal-cardinality diagnosis. Hence, there is no one-to-one mapping from diagnosis to SAT.

This paper introduces a randomized restarts algorithm for computing diagnoses within a model-based diagnosis framework, which was inspired by randomized satisfiability algorithms. We have shown the theoretical justification for a randomized restarts algorithm, i.e., that weak fault models have a heavy-tailed distribution, and hence are amenable to this class of algorithm. We have applied this algorithm to a suite of benchmark combinatorial circuits, demonstrating order-of-magnitude speedup over a well-known deterministic algorithm, CDA*, for multiple-fault diagnoses. Moreover, whereas the search complexity for the deterministic algorithms tested increases exponentially with fault cardinality, the search complexity for this stochastic algorithm appears to be independent of fault cardinality.

The rest of this paper is organized as follows. In the next section we discuss related work. A section formalizing some theoretical notions in MBD follows. The fourth section discusses the probability distribution of diagnoses which is an important motivation for the algorithm shown in the fifth section. A section with experimental results follow before the conclusions and future work.

Related Work

This section reviews related work in SAT and MBD.

SAT Algorithms

The past several years has seen dramatic improvements in algorithms for solving the SAT problem. Most modern SAT solvers are based on one of two approaches: (a) the Davis-Putnam-Logemann-Loveland (DPLL) algorithm (Prasad, Biere, & Gupta 2005), which is based on a branching search with backtracking; or (b) stochastic local search (Hoos & Tsang 2006). The advantage of DPLL methods is that they are sound and complete; however, DPLL-based tree-search algorithms are intractable in the worst case, and, although they perform well on problem instances that are critically-constrained and over-constrained, they perform worst than local search algorithms on under-constrained problems. Stochastic local search algorithms, while incomplete, can scale to very large problem instances; however, they work best on random SAT instances, and in practice work less well on structured instances obtained from real problems, e.g., verification or planning problems.

Restarts-based algorithms (Gomes, Selman, & Kautz 1998) combine the advantages of both these approaches. The idea of restarts is based on the recently discovered phenomenon that the runtime distributions for many classes of soluble problems are heavy-tailed distributions (Gomes, Selman, & Crato 1997; Gomes *et al.* 2000).

Inspired by heavy-tailed distributions, randomization and rapid random restarts have proven to be extremely useful for solving very large problem instances (Sellmann &

Ansótegui 2006). The rapid random restarts approach to solving aims to find one of the short runs that contribute to the low median runtime that characterize heavy-tailed runtime distributions. By setting a restart cutoff, we define an upper bound on the number of backtracks in any run. If this cutoff is exceeded, we simply abandon the current search and restart from a random instantiation. By ensuring there is a degree of randomness in our search heuristics, we can be confident that we will search for a solution in quite a different way, hopefully solving the problem quickly by finding one of those short runs.

Model-Based Diagnosis Algorithms

We can classify MBD algorithms into two main groups: (a) algorithms that find solutions one at a time, e.g. (Venkatasubramanian, Rengaswamy, & Kavuri 2003); and (b) algorithms that find all solutions simultaneously, e.g., ATMS (de Kleer 1986) and causal-network (Darwiche 1998; Marquis 2000) algorithms.

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.

Vatan et al. (2003) have mapped the diagnosis problem into the monotone SAT problem, and then propose to use efficient SAT algorithms for computing diagnoses. This approach 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.

The MBD problem is different than SAT in that it is an optimization problem, i.e., one typically wants to find a minimal diagnosis, using some minimality criterion such as minimal-cardinality diagnosis. Hence one cannot map the diagnosis problem directly to SAT. We can show an encoding whereby one can use cardinality constraints (Bailleux & Boufkhad 2003) to encode a notion of minimal diagnosis.

Stochastic algorithms have been discussed in the framework of constraint satisfaction (Freuder *et al.* 1995) and Bayesian network inference (Kask & Dechter). The latter two approaches can be used for solving suitably translated MBD problems. It is often the case, though, that these new encodings are more difficult for search than more specialized ones.

Inspired by the success of stochastic SAT and stochastic constraint solving we show that, under certain conditions, the diagnosis inference complexity distribution is heavy-tailed and may benefit from random restarts. As finding a minimal-cardinality diagnosis is an optimization problem, we apply a greedy approach, the latter allowing us to find local optima very close to the global optimum. Combining the stochastic approach and greedy search specific to a wide class of MBD problems allows us to construct the algorithm which we present in this paper.

Technical Background

The discussion starts by formalizing some basic notions in MBD (de Kleer, Mackworth, & Reiter 1992). A *qualitative* model of an artifact is represented as a propositional $\mathbf{Wff}$ over some set of variables V. Discerning a subset of them as *assumable* or *observable* gives us a diagnostic system.

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

Let α be any (possibly partial) instantiation of the variables in OBS. The traditional diagnostic query in MBD results in finding terms of assumable variables which are explanations for $\mathrm{SD} \wedge \alpha$. These *implicants* of $\mathrm{SD} \wedge \alpha$ do not necessarily instantiate all the assumable variables in SD, i.e., "don't cares" are allowed.

Definition 2 (Partial Diagnosis). A term ω over a set of assumable variables $h \in \mathrm{COMPS}$ is a partial diagnosis of $\mathrm{SD} \wedge \alpha$ iff $\omega \models \mathrm{SD} \wedge \alpha$.

Under *lex parsimoniae* we are interested in computing these partial diagnoses only, which are not contained in other implicants of $\mathrm{SD} \wedge \alpha$. These partial diagnoses, minimal under subsumption, are the *prime implicants* of $\mathrm{SD} \wedge \alpha$.

Definition 3 (Kernel Diagnosis). A partial diagnosis ω is a kernel diagnosis iff no $\omega' \models \mathrm{SD} \wedge \alpha$ exists, such that ω' is the conjunction of a proper subset of the literals in ω.

It is possible for a kernel diagnosis ω to contain both positive and negative literals. This is different from the notion of *minimal diagnosis* as defined in (de Kleer & Williams 1987). The advantage is the lack of restrictions on SD, i.e., SD can be any propositional theory. Computing all kernel diagnoses, however, is strictly more difficult than computing the set of all minimal diagnoses.

Given a kernel diagnosis ω, we are often interested in the subset of these literals in ω which have the same (e.g., negative) polarity. Loosely speaking, each negative literal would manifest a single malfunctioning component. Simply counting the number of these negative literals in ω gives us *diagnosis cardinality*.

Definition 4 (Min-Cardinality). Let $\omega \models \mathrm{SD} \wedge \alpha$, ω be a term and $Card(\omega)$ denote the number of negative literals in ω. The minimum cardinality of $\mathrm{SD} \wedge \alpha$ is defined as $\min_{\omega \models \mathrm{SD} \wedge \alpha} Card(\omega)$.

The aforementioned notion of cardinality can be slightly generalized by introducing a cardinality function $f : \mathbb{B}^n \rightarrow \mathbb{Z}^+$ from the space of all possible diagnoses. For the purpose of this paper, however, it suffices to count the negative literals. The number of negative literals in the minimum cardinality diagnosis of $\mathrm{SD} \wedge \alpha$ is denoted as $MinCard(\alpha)$.

We discover interesting properties in a class of theories often seen in MBD, these are the models which define normative behavior of their components only, i.e., models which

specify no fault-modes. These models are sometimes referred to as *weak-fault models*.

Definition 5 (Implicit Fault System). A diagnostic system DS belongs to the class **IFS** iff SD is in the form $(h_1 \Rightarrow F_1) \wedge \ldots \wedge (h_n \Rightarrow F_n)$ such that for $1 \leq i, j \leq n$, $\{h_i\} \subseteq$ COMPS, $F_j \in$ **Wff**, and none of h_i appears in F_j.

In the next section we show that diagnosis distribution for a system DS $\in$ **IFS** is heavy-tailed.

Diagnosis Distribution is Heavy-Tailed

This section shows that, under the assumption of computing maximum-probability diagnoses, the distribution of faults is heavy-tailed.[1] This is the theoretical underpinning for why the approach should work.

Heavy-Tailed Distributions

Heavy-tailed distributions (also known as power-law distributions) have been observed in many natural systems, such as physical (biological or man-made) and sociological systems.

Definition 6 (Heavy Tail). A probability density function (pdf) for a random variable X is said to have a heavy-tail if:

$$Pr(X > x) \sim x^{-\alpha}, \text{ as } x \to \infty, \ 0 < \alpha < 2.$$

This means that, regardless of the distribution, for small values of the random variable X, the distribution is heavy-tailed if its asymptotic shape is hyperbolic (Wikipedia 2007). The simplest heavy-tailed distribution is the Pareto distribution, which is hyperbolic over its entire range and has probability density function:

$$p(x; \alpha, k) = \alpha k^{\alpha} x^{-\alpha-1}, \text{ for } \ \alpha, k > 0, \ x \geq k.$$

The Pareto distribution has cumulative distribution function:

$$F(x; \alpha, \eta) = Pr[X \leq x] = 1 - \left(\frac{\eta}{x}\right)^{\alpha},$$

where η represents the smallest value the random variable X can take. Heavy-tailed distributions have properties that are qualitatively different to commonly-used (memoryless) distributions such as the exponential, normal or Poisson distribution.

Heavy-tailed and long-tailed distributions can be distinguished by examining the log-log plot of the pdf. In long-tailed distributions, the log-log plot of the tail is approximately linear over many orders of magnitude. If the logarithm of the range of an exponential distribution is found, the resulting plot is linear; in contrast, that of the heavy-tail distribution is still curvilinear (Wikipedia 2007).

Diagnosis Distributions

Consider a probabilistic ordering over the diagnoses. To compute this, we use a valuation function Pr : COMPS $\to [0, 1]$. In practice, Pr assigns a priori probabilities to the system failure modes. Under the assumption that all variables in SD are conditionally independent, we define the valuation of an assignment α to be $Pr(\alpha) = \Pi_{x \in \alpha \cup \text{COMPS}} Pr(x)$.

We assume that the failure-mode distribution Pr is skewed. i.e., there exists asymmetry in the probability distribution of Pr. In other words, we assume that every variable x_i has $Pr(x_i = OK) \geq \beta Pr(x_i = bad)$. Such a skewed distribution is normal in diagnostics, and implies that normal behavior is β times more likely than faulty behavior, $\beta \geq 1$.

We take one of the most common definitions of diagnosis ranking, i.e., a ranking imposed by the probability of the diagnosis. Let ω_i be a diagnosis with i broken components. By our skewness assumption, we know that $Pr(\omega_i) \geq \beta Pr(\omega_{i+1})$.

Theorem 1. *If we assume that $\beta Pr(X_i = bad) = Pr(X_i = OK)$ for all failure modes X_i, then we obtain a heavy-tailed (Zipf) distribution for the diagnosis distribution.*

Proof. We give a sketch of the proof. Let γ_k be the (normalized) probability that a diagnosis has k faults. Under the assumption that $\beta Pr(X_i = bad) = Pr(X_i = OK), \ \forall X_i$, we can then show that the probability distribution for γ is given by

$$\gamma(k; \beta, n) = \frac{1/k^{\beta}}{\sum_{j=1}^{n} 1/j^{\beta}},$$

where $\gamma(k; \beta, n)$ is the number of mode variables of the k^{th} ranked diagnosis, k is the rank of a diagnosis, and β is the diagnosis likelihood exponent characterizing the distribution.

This rank distribution over the diagnoses is known as a power-law (or Zipf's) distribution, which, by definition, is heavy-tailed. □

We can also define the diagnosis cumulative distribution function (cdf) as:

$$\Gamma(k; \beta, n) = \frac{\sum_{j=1}^{k} 1/j^{\beta}}{\sum_{j=1}^{n} 1/j^{\beta}}.$$

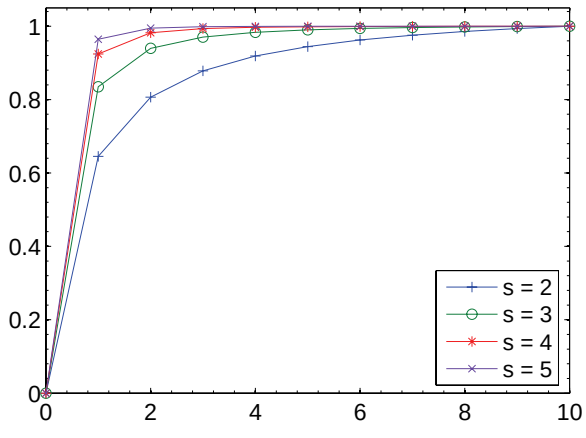

Figure 1: CDF $\Gamma(k; s, 10)$.

[1] This analysis can be extended to other orderings over diagnoses.

Figure 1 shows the cdf of $\Gamma(k; s, 10)$ plotted against index k, for various values of the distribution exponent s.

If we relax the assumption that all fault probabilities are identical, then we can generalize our proof to show that the diagnosis distribution is a Pareto distribution. Note that both the Zipf and Pareto distribution are, by definition, heavy-tailed distributions.

Distribution of Diagnosis Inference Complexity is Heavy-Tailed

To fully demonstrate the reasons for the success of randomized restarts for diagnosis, we need to show that the inference necessary to locate the faults is heavy-tailed. We can show that the minimal-cardinality diagnosis problem can be mapped to SAT by introducing just $O(n)$ extra clauses (Bailleux & Boufkhad 2003; Sinz 2005). Given such a mapping, we can just rely on the well-established result that the number of backtracks necessary to solve SAT is heavy-tailed (Gomes, Selman, & Crato 1997).

For example, consider the encoding proposed by Bailleux and Boufkhad (2003). This encoding is such that, if the cardinality constraint is less than the number N of variables, as soon as N variables are true, i.e., appear in a positive unit clause, the SAT routine unit resolution can derive a contradiction. This approach been empirically demonstrated to be efficient in solving difficult problems in parity learning (Bailleux & Boufkhad 2003) and in finding good quality plans in AI planning (Büttner & Rintanen 2005).

Stochastic MBD Algorithm

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

A Simple Example

The Boolean circuit shown in Figure 2 is used to give a basic idea about the intuition behind the algorithm discussed in this paper. The subtractor consists of total of seven components: an inverter, two or-gates, two xor-gates, and two and-gates.

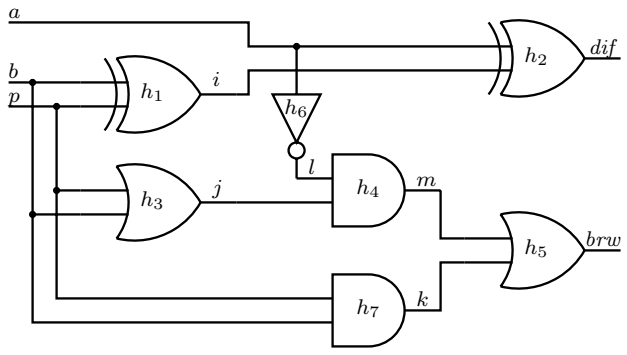

Figure 2: A subtractor circuit.

The expression $h \Rightarrow (o \Leftrightarrow \neg i)$ models an inverter, where the variables i, o, and h represent input, output and health respectively. Similarly, an and-gate is modeled as $h \Rightarrow (o \Leftrightarrow i_1 \wedge i_2)$ and an or-gate is $h \Rightarrow (o \Leftrightarrow i_1 \vee i_2)$. Finally, an xor-gate is specified as $h \Rightarrow (o \Leftrightarrow \neg(i_1 \Leftrightarrow i_2))$.

These propositional formulae are copied for each gate in Figure 2 and their variables renamed in such a way as to properly connect the circuit and disambiguate the assumables, thus receiving a propositional formula for SD:

$$\mathrm{SD} = \begin{cases} h_1 \Rightarrow (i \Leftrightarrow \neg(b \Leftrightarrow p)) \\ h_2 \Rightarrow (dif \Leftrightarrow \neg(a \Leftrightarrow i)) \\ h_3 \Rightarrow (j \Leftrightarrow b \vee p) \\ h_4 \Rightarrow (m \Leftrightarrow l \wedge j) \\ h_5 \Rightarrow (brw \Leftrightarrow m \vee k) \\ h_6 \Rightarrow (a \Leftrightarrow \neg l) \\ h_7 \Rightarrow (k \Leftrightarrow b \wedge p) \end{cases}$$

The set of component (assumable) variables is $\mathrm{COMPS} = \{h_1, \ldots, h_7\}$. Consider an example observation $\mathrm{OBS} = a \wedge b \wedge \neg p \wedge dif \wedge \neg brw$. This leads to a triple fault minimal-cardinality diagnosis: $\omega_1 = \neg h_1 \wedge \neg h_3 \wedge \neg h_7$. The same observation is explained by five more triple fault diagnoses, which we will omit from this example for brevity. We notice also that $\neg h_7$ has the same sign in all diagnoses of $\mathrm{SD} \wedge \mathrm{OBS}$.

A deterministic A* search for the above diagnosis discovers it after 95 assignments to assumable variables and 27 consistency checks. Even enabling conflict focusing does not reduce the number of assignment below 42 and the number of consistency checks below 16, which shows how deterministic diagnosis search becomes impractical with bigger systems.

We will now show a two-step diagnostic process which takes a smaller number of variable assignments and consistency checks. Step 1 involves randomly choosing candidates. Step 2 attempts to minimize the fault cardinality in these candidates.

In step 1, the stochastic diagnostic search for the subtractor example will start from a random quintuple candidate[2]. In this particular version of our algorithm, once a component is marked as healthy, it cannot be changed back to faulty. To compensate for that and to provide for the heavy-tailed inference complexity outlined above, we perform multiple restarts from a random candidate. In our subtractor example, if h_7 is in the initial "guess" it will prove inconsistent with $\mathrm{SD} \wedge \mathrm{OBS}$ and another quintuple fault candidate will be guessed.

Assume that the second candidate is $\omega_2' = \neg h_1 \wedge h_2 \wedge \neg h_3 \wedge \neg h_4 \wedge \neg h_5 \wedge h_6 \wedge \neg h_7$. Clearly, $\omega' \models \mathrm{SD} \wedge \mathrm{OBS}$. The search algorithm may next try to improve the diagnosis by "flipping" h_4. The candidate $\omega_3' = \neg h_1 \wedge h_2 \wedge \neg h_3 \wedge h_4 \wedge \neg h_5 \wedge h_6 \wedge \neg h_7$ is a valid quadruple fault diagnosis and it can be improved once more by "flipping" $\neg h_5$. This gives us the final triple fault diagnosis.

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

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

A Greedy Stochastic Algorithm for Computing Minimal-Cardinality Diagnosis

The greedy stochastic algorithm, we introduce next, can find multiple minimal-cardinality diagnoses (if such exist). We will restrict our discussion, however, to finding a single, arbitrary, minimal-cardinality diagnosis.

The stochastic algorithm presented in this paper uses the previously-described extension to DS, a valuation function $Pr : \text{COMPS} \rightarrow [0, 1]$. In the analysis of our algorithm we use Pr to determine if an assignment to a health variable denotes a failure or a healthy mode. The performance of the algorithm presented in this paper is not sensitive to Pr and the only use of the probabilities is to guide the search for more efficient performance.

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

```
1:  function HillClimb(DS, M, N, Pr) returns a trie
        inputs: DS = ⟨SD, COMPS, OBS⟩, a diag. system
                M, integer, climb restart limit
                N, integer, number of tries
                Pr, a valuation function
        local variables: m, n, integers
                         ω, ω′, terms
                         R, a trie
2:      n ← 0
3:      while n < N do
4:          ω ← RandomCandidate(Pr)
5:          if ω ⊨ SD ∧ OBS then
6:              m ← 0
7:              while m < M do
8:                  ω′ ← ImproveDiagnosis(Pr, ω)
9:                  if ω′ ⊨ SD ∧ OBS then
10:                     ω ← ω′
11:                     m ← 0
12:                 else
13:                     m ← m + 1
14:                 end if
15:             end while
16:             unless IsSubsumed(R, ω) then
17:                 AddToTrie(R, ω)
18:                 RemoveSubsumed(R, ω)
19:             end unless
20:             n ← n + 1
21:         end if
22:     end while
23:     return R
24: end function
```

The randomized search process performed by Algorithm 1 is parameterized by the parameters M and N. There are N independent searches that start from randomly generated candidates. After an initial candidate ω is found to be consistent with SD $\wedge$ OBS, i.e., ω is a diagnosis, the algorithm tries to improve the cardinality of the diagnosis (while preserving its consistency) by randomly "flipping" fault literals.

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

Similar to deterministic methods for MBD, Algorithm 1 uses a SAT-based procedure for checking the consistency of SD $\wedge$ OBS $\wedge$ ω. Because SD and OBS do not change in consistency checks, using an LTMS (McAllester 1990) can improve search efficiency. In our implementation, we have combined a BCP-based LTMS to check for inconsistencies, and if a candidate does not prove to be inconsistent, a second check with a DPLL-based method is invoked for completeness.

The RandomCandidate function generates a candidate diagnosis, used for "seeding" the diagnostic search. The valuation function can be modified in such a way as to provide a more informed starting point, thus decreasing the number of "climbing" steps. The initial diagnosis ω should be of high cardinality, to increase the likelihood of $\omega \models$ SD $\wedge$ OBS. In order to do that, we generate an instantiation of ω by using Pr and *scaling* the a priori probabilities in Pr to bias the pdf from which we draw the initial candidates. Consider an example in which each component $h \in$ COMPS fails with a probability of 5%. The valuation function is $Pr(h = \textbf{False}) = 0.05$. We may use a scaling coefficient $k = 5$ which would lead to RandomCandidate returning a candidate with a quarter of the components failing.

The biasing of Pr can improve the efficiency of Algorithm 1 by exploiting knowledge about the likelihood of the cardinality of the minimal-cardinality diagnosis. In particular, when expecting minimal-cardinality diagnoses of high cardinality Pr should be configured to return an initial fault of higher cardinality. If the expected faults are of small cardinality, the search may start from a candidate with smaller number of faulty components, in which case more attempts (increased N) would be necessary to find local diagnoses close to the global optimum.

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

There is no guarantee that two diagnostic searches, starting from a random diagnoses, would not lead to the same minimal-cardinality diagnosis. To prevent this, we store the generated diagnoses in a trie R, from which it is straightforwards to extract the resulting diagnoses by recursively visiting its nodes. A diagnosis ω is added to the trie R by the function AddToTrie, iff no subsuming diagnosis is contained in R (the IsSubsumed subroutine checks on that condition). After adding a diagnosis ω to the resulting trie

R, all diagnoses contained in R and subsumed by ω are removed by a call to REMOVESUBSUMED. The workings of the trie functions as well as a thorough description of the trie data structure can be found in (Forbus & de Kleer 1993).

The complexity of Algorithm 1 is derived in a straightforward way.

Proposition 1. *The time complexity of Algorithm 1 is $O(MN \log NC)$, where C is the complexity of the consistency checking procedure.*

The $\log N$ factor comes from the trie maintenance, which contains a maximum number of N diagnoses with some ordering imposed on their literals. Note, that the average case complexity of consistency checking, although exponential in the worst case, is low polynomial when incomplete methods like BCP are used (Zabih & McAllester 1988) or when the model is highly-observable.

Note that this is effectively a polynomial-time algorithm that trades off some small amount of completeness and optimality for significant improvements in efficiency relative to deterministic diagnosis algorithms.

Experimental Results

The next section discusses some empirical results measured from an implementation of the algorithm shown in this paper.

Implementation Notes and Test Set Description

We have implemented our algorithm in approximately 700 lines of C code (excluding the LTMS and DPLL consistency checking) and it is a part of the LYDIA[3] package. LYDIA employs deterministic algorithms for computing minimal-cardinality diagnoses based on conflict-based search (Williams & Ragno 2004) and exploitation of structure (Feldman & van Gemund 2006).

Table 1 summarizes the benchmark problems we have used for testing our algorithms. All models are derived from the 74XXX and ISCAS85 family of benchmark circuits. We have added assumable variables for each component in each model. Weak fault models, similar to the one from the example in the previous section, have been used.
The basic characteristics of the ISCAS-85 models are shown in Table 1. The number of assumable variables is denoted as H. We have counted the number of variables V and the number of clauses in the CNF representation is denoted as C_w. The number of observable variables is denoted as O.

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

Performance Analysis

In our experiments we compared the diagnostic time of the stochastic algorithm and implementations of the CDA* (Williams & Ragno 2004) and HA* algorithms (Feldman & van Gemund 2006). We have parameterized Algorithm 1 with $M = 4$ and $N = 8$, i.e., it makes 8 independent attempts to find a minimal-cardinality diagnosis and in each try the algorithm gives up after 4 unsuccessful attempts to improve the cardinality of a diagnosis.

[3]The LYDIA package for model-based fault diagnosis can be downloaded from `http://fdir.org/lydia/`.

Name	Description	H	V	C_w	O
74180	9-bit parity check	14	38	48	12
74139	2-to-4 decoders	18	42	52	14
74153	4-to-1 selector	16	44	62	14
74182	4-bit CLA	19	47	75	14
74283	4-bit adder	40	89	130	14
74L85	4-bit comparator	41	93	134	14
74181	4-bit ALU	62	138	216	22
c432	27-chan. int. controller	146	328	486	43
c499	32-bit SEC circuit	202	445	714	73
c880	8-bit ALU	383	826	1 112	86
c1355	32-bit SEC circuit	514	1 069	1 546	73
c1908	16-bit SEC/DEC	252	541	911	58
c2670	12-bit ALU	983	2 153	2 856	226
c3540	8-bit ALU	1 297	2 685	3 861	72
c5315	9-bit ALU	2 202	4 800	6 983	295
c6288	32-bit multiplier	2 416	4 864	7 216	64
c7552	32-bit adder	3 024	6 465	9 085	325

Table 1: Basic characteristics of the 74XXX and ISCAS-85 fault models. H, V, C_w and O denote the number of assumable variables, variables, clauses (in CNF representation), and observable variables, respectively.

We have used the same valuation function Pr for all the experiments. In particular, $Pr(h = \mathbf{False}) = 0.01$, and $Pr(h = \mathbf{True}) = 0.99$. For the initial candidate generation, we have scaled Pr: $Pr(h = \mathbf{False}) = Pr(h = \mathbf{True}) = 0.5$. Hence, the initial diagnosis has half of its literals denoting a faulty state.

In our first experiment we use small models with observations maximizing the number of faults in a minimal-cardinality diagnosis. The results, shown in Table 2, illustrate the most important advantage of the stochastic algorithm: its performance does not degrade when the fault cardinality increases.

Name	Faults	T_h	T	C
74180	2	2	1	2
74139	8	10	2	8
74153	2	3	1	2
74182	5	40	2	5
74283	5	4 633	4	5.4
74L85	3	143	4	3
74181	7	169 583	9	7

Table 2: Times [ms] for diagnosing multiple faults of higher cardinality.

Table 2 compares the time for finding a single minimal-cardinality diagnosis of the HA* algorithm (which in these cases performs better than CDA*) and the stochastic algorithm shown in this paper. We have denoted the search time of HA* as T_h and that of the stochastic algorithm as T.

The stochastic method discussed in this paper, is a local search algorithm, and hence it can compute a suboptimal diagnosis. This was the case in the 74283 model experiments,

in which 4 out of 10 runs returned a minimal-cardinality diagnosis with 6 faults, while the global optimum has 5 faults, hence the 5.4 value in Table 2. We have denoted the cardinality of the diagnosis returned by the stochastic algorithm as C. To allow for the random component in the stochastic search, we have run it 10 times and the results in C and T are averaged.

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

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

Table 3: Running times [ms] for the CDA* and the stochastic algorithms.

The times for the stochastic algorithm to discover a single and a double fault are denoted as T_1 and T_2 respectively. In some of the cases, our stochastic algorithm could not find a minimal-cardinality diagnosis, but a suboptimal one. We have shown the cardinality of the results for single and double faults in columns C_1 and C_2, respectively. Again, the values of T_1, C_1, T_2, and C_2 are averaged over 10 runs.

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

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

Similar to the earlier experiments, the performance of Algorithm 1 does not degrade when the number of faults increases. This is not the case with heuristic-based deterministic algorithms like CDA* or HA*. The time for the stochastic algorithm to find a double fault is the same as for finding a single fault, while CDA* suffers from a combinatorial explosion.

Based on these experiments, we may conclude that in most of the cases, the stochastic algorithm finds diagnoses of near optimal cardinality. The diagnostic time of the stochastic algorithm is not affected by the number of faults in the minimal-cardinality diagnosis, which is certainly not the case with the two deterministic algorithms. The only case in which the stochastic algorithm performs slightly worse than a deterministic one is with single fault diagnoses.

Discussion and Future Work

For an important class of fault models, the distribution of diagnoses is heavy-tailed, that is, a few diagnoses of small cardinality concentrate most of the probability mass for some suitably-defined valuation function. Furthermore, the inference complexity for discovering these few diagnoses has the same distribution. As a consequence, deterministic backtracking or heuristic-based methods are efficient only for a small class of systems.

In this paper, we suggest a stochastic approach for improving the average-case complexity of minimal-cardinality diagnosis, using a method that provably takes advantage of heavy-tailed distributions. We introduce an algorithm for finding diagnoses of near-minimal cardinality. The main idea behind our algorithm is randomly generating candidates until one of them turns out to be a diagnosis, and then to trying to improve the cardinality of the diagnosis by "flipping" fault variables assigned as *Bad* to *OK*.

Having once generated an initial diagnosis, reducing its cardinality (or "climbing") is facilitated by the fault modeling. In particular, the minimal diagnosis hypothesis says that for the class of models where no fault modes are specified, every superset of a diagnosis is also a diagnosis. The opposite, of course, does not hold, but starting from a superset of a diagnosis, it is easy to find the minimal one due to the favorable nature of the search space.

We have tested our algorithm on a family of 74XXX and ISCAS-85 circuits. The stochastic search could find nearly optimal diagnoses, even for the biggest circuit, where deterministic methods failed to do that. Furthermore, the performance of the stochastic algorithm is determined only by the size of the model and not by the cardinality of the minimal-cardinality diagnosis it searches for. This makes our algorithm especially suitable for systems where many faults have to be diagnosed.

This paper raises a number of research questions, which we plan to address in future work. On the theoretical side, we have stipulated that weak-fault models have a heavy-tailed diagnosis distribution. Outside this class, artificial cases can be created having any fault distribution, but we expect that these cases are rarely seen in reality. We expect semi-weak models, for which nominal behavior and *some* failure modes are specified, to be dominant in fault-modeling. We plan a more extensive investigation of such fault distributions when more empirical data from these cases is collected.

On the algorithmic side, we would like to experiment with a wider variety of stochastic methods. These include simulated annealing, genetic search and others. The algorithmic work would benefit from an extensive set of benchmarking models, coming not only from digital circuits, but random models, real-world models and others. Last, we are interested to expanding our algorithms on a wider class of abduction and constraint optimization problems.

Conclusion

We have described a randomized restarts algorithm for computing diagnoses within a model-based diagnosis framework, which was inspired by randomized satisfiability algorithms. We have shown that weak fault models have a heavy-tailed distribution, and hence are amenable to this class of algorithm. We have applied this algorithm to a suite of benchmark combinatorial circuits, and shown significant performance improvements for multiple-fault diagnoses, compared to a well-known deterministic algorithm, CDA*. Our results indicate that, although the randomized restarts algorithm is outperformed for the single-fault diagnoses, it shows at least an order-of-magnitude speedup over CDA* for double-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.

Acknowledgments

This work has been supported by STW grant DES.7015 and SFI grant 04/IN3/I524.

References

Bailleux, O., and Boufkhad, Y. 2003. Efficient cnf encoding of boolean cardinality constraints. In *CP*, 108–122.

Büttner, M., and Rintanen, J. 2005. Satisfiability planning with constraints on the number of actions. In *Proc. ICAPS'05*, 292–299.

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

Darwiche, A. 1998. Model-based diagnosis using structured system descriptions. *J. Artificial Intelligence Research* 8:165–222.

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

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

de Kleer, J. 1986. An assumption-based TMS. *Artif. Intell.* 28(2):127–162.

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

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

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

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

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

Gomes, C. P.; Selman, B.; Crato, N.; and Kautz, H. A. 2000. Heavy-tailed phenomena in satisfiability and constraint satisfaction problems. *J. Autom. Reasoning* 24(1/2):67–100.

Gomes, C. P.; Selman, B.; and Crato, N. 1997. Heavy-tailed distributions in combinatorial search. In *CP*, 121–135.

Gomes, C. P.; Selman, B.; and Kautz, H. A. 1998. Boosting combinatorial search through randomization. In *Proc. AAAI'98*, 431–437.

Hoos, H. H., and Tsang, E. 2006. Local search methods. In *Handbook of Constraint Programming*. 245–276.

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

Marquis, P. 2000. Consequence finding algorithms. In *Algorithms for Defeasible and Uncertain Reasoning*, volume 5 of *Handbook on Deafeasible Reasoning and Uncertainty Management Systems*. 41–145.

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

Prasad, M. R.; Biere, A.; and Gupta, A. 2005. A survey of recent advances in SAT-based formal verification. *International Journal on Software Tools for Technology Transfer (STTT)* 7(2):156–173.

Sellmann, M., and Ansótegui, C. 2006. Disco - Novo - GoGo: Integrating local search and complete search with restarts. In *Proc. AAAI'06*.

Sinz, C. 2005. Towards an optimal cnf encoding of boolean cardinality constraints. In *CP*, 827–831.

2003. A novel model-based diagnosis engine: theory and applications. In *IEEE Aerospace Conference*.

Venkatasubramanian, V.; Rengaswamy, R.; and Kavuri, S. N. 2003. Review of process fault diagnosis - part II: Qualitative models and search strategies. *Computers and Chemical Engineering* 27(3):313–326.

Wikipedia. 2007. *Heavy-Tailed Distributions*.

Williams, B., and Ragno, R. 2004. Conflict-directed A* and its role in model-based embedded systems. *Journal of Discrete Applied Mathematics*.

Zabih, R., and McAllester, D. 1988. A rearrangement search strategy for determining propositional satisfiability. In *Proc. AAAI'88*, 155–160.