Modeling Diagnostic Stochastic Search

Alexander Feldman^{1,3}, Tom Janssen², and Arjan van Gemund³

¹ Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud Route de Cheseaux I, CH-1401 Yverdon-les-Bains, Switzerland alexander,feldman@heig-vd.ch

² Scarabee Logistic Systems B.V., Haringvliet 100, 3100 TH, Rotterdam, The Netherlands

tjanssen@scarabee.nl

³ Delft University of Technology, Mekelweg 4, 2628 CD, Delft, The Netherlands a.j.c.vangemund@tudelft.nl

ABSTRACT

One of the main problems of Model-Based Diagnosis (MBD) is, given a system description and an observation, to compute all cardinalityminimal diagnoses. This problem is often NPhard or harder and overcoming combinational explosion attracts a lot of research. One way to deal with the computation complexity of MBD is to use stochastic methods. Stochastic methods for MBD trade optimality for speed. Similar to related problems, such as stochastic ALLSAT or model counting, the main problem with stochastic diagnostic search is the lack of termination criterion. In this paper we propose a novel method for predicting the probability of stochastic diagnostic search returning a cardinality-minimal diagnosis. This method is based on a computationally very efficient approximate model of the diagnostic search process. The probability of computing a cardinality-minimal diagnosis is used as a termination criterion for the stochastic diagnostic search. We show several models with various trade-offs in complexity versus accuracy.

1 INTRODUCTION

Conventional algorithms for Model-Based Diagnosis (MBD) are deterministic and are fast in finding low cardinality faults, e.g., when there is only one faulty component in the system. A popular deterministic approach used in model-based diagnosis is the conflict-directed A* (CDA*) algorithm (Williams and Ragno, 2007), which is built upon the concepts of the general diagnostic engine (GDE) (De Kleer and Williams, 1987). This method tries to find the best next candidate while continuously decreasing the search space by checking for inconsistencies. A similar conflict-directed best first search algorithm is adopted by the Livingstone kernel (Williams and Nayak, 1996), which is successfully used in several missions of NASA (Bajwa and Sweet, 2003; Muscettola et al., 1998). Another state-of-the-art approach to model-based diagnosis, called HA* (Feldman and van Gemund, 2006), exploits the hierarchy of system

models. It consists of an expensive pre-processing step and a very fast diagnosis step. Because the exponential cost of pre-processing is done only once for a model, this is an attractive approach.

However, when multiple components fail at the same time, these deterministic algorithms suffer from poor performance due to the fact that time complexity is exponential in the number of faults. A popular example of a multiple component failure in the field of space exploration is the Apollo 13 crisis, as mentioned in (Muscettola *et al.*, 1998). In this situation no less than five faults occurred in the system simultaneously: three electrical shorts, the bursting of a tank-line, and a pressure jacket. With little sensor information, diagnosing the health state of the system was a huge challenge for ground control.

SAFARI (StochAstic Fault diagnosis AlgoRIthm) (Feldman *et al.*, 2010) has been developed to overcome the time complexity of the conventional models. Being a stochastic algorithm, the time to solve a problem becomes roughly constant, independent of the number of simultaneous faulty components in the system. It has been shown (Feldman *et al.*, 2008) that SAFARI is able to produce possible diagnoses for difficult problems that deterministic algorithms are not able to diagnose within a reasonable amount of time (weeks or months).

We define the *performance* of a stochastic diagnosis algorithm in terms of the probability density of the low cardinality solutions, as the solutions of low cardinality have the highest probability of being the correct diagnosis (Janssen, 2011). If a stochastic diagnosis engine has a high probability of returning a solution of low cardinality, then we consider the performance of the algorithm to be good. For example, for an imaginary problem there exists a pool of possible solutions with a Gaussian-like cardinality distribution. An algorithm with bad performance would randomly pick a solution from this pool. An algorithm with good performance would be able to return low cardinality solutions with higher probability. This is illustrated in Fig. 1.

The applicability of the original SAFARI algorithm is limited due to the uncertainty of completeness and



Figure 1: (a) Bad performance: probability density highest at medium cardinalities. (b) Good performance: probability density highest at low cardinalities

optimality of the results. A correct *performance model* of the algorithm can give an indication of the optimality and completeness during the diagnosis process. This model may lead to a prediction of the behavior and a termination criterion that assures that the probability of finding a new relevant diagnosis is sufficiently small. The problem statement therefore becomes:

Can we find a correct performance model for the SAFARI algorithm and from this model devise a termination criterion, in order to provide a proper termination condition of the diagnosis process, while ensuring a certain degree of completeness of the returned diagnoses?

In this paper an analysis is made of the problem mentioned above and possible solutions are explored and discussed. The contributions of this paper are listed below.

- The current performance model of SAFARI as defined in (Feldman *et al.*, 2010), dubbed A_0 in this paper, is shown to be insufficient to form a basis for a termination criterion, as it disregards the impact of fault cardinalities other than the target cardinality.
- A characterization of performance S is introduced, that will categorize the many solutions possible for a system into a relatively small set, that is used to further analyze and model the SA-FARI algorithm.
- The characterization of performance S is shown to be the determining factor of the behavior of SAFARI.
- Different models of the SAFARI algorithms are explored, each model having a different level of abstraction from the actual algorithm. These models, A_1 , A_2 , and A_3 , are compared and one model is selected as best candidate for modeling SAFARI.
- Having analyzed and modeled the SAFARI algorithm, a criterion is proposed to terminate the algorithm while ensuring a certain level of completeness.

2 TECHNICAL BACKGROUND AND THE SAFARI ALGORITHM

The SAFARI algorithm for greedy diagnostic search is described by Feldman *et al.* (2010). This paper can be

considered as a continuation of the work of Feldman *et al.* whose results, notation, and terminology we use fully.

We have as a running example in the current paper a weak-fault model of the 74180 circuit (shown in Fig. 2), using the standard modeling approach and formalism.

3 PERFORMANCE MODELING

In what follows we explore a feature of the diagnostic search space that we can use for modeling (and predicting) the performance of SAFARI.

3.1 Performance Characteristics of Greedy Stochastic Search

SAFARI does not exploit the system model for optimizing the way it searches for solutions, i.e., the only subroutine of SAFARI that uses *all* information in SD is the consistency checking oracle. The algorithm flips literals randomly, not based on SD \land OBS, and the behavior of SAFARI when reaching one diagnosis of, for example, cardinality $|\omega| = 3$ is similar to its behavior when reaching another diagnosis of the same cardinality.

Consider the set of subset-minimal diagnoses $\Omega^{\subseteq}(\mathrm{SD} \wedge \alpha)$ for a given model SD and a given observation α . Let us denote the number of subset-minimal diagnoses of cardinality $|\omega^{\subseteq}|$ as $||\omega^{\subseteq}||$. Given $\mathrm{SD} \wedge \alpha$, the sizes of the various sets of subset-minimal diagnoses $||\omega^{\subseteq}||$ give us a distribution function S as defined next.

Definition 1 (S-signature). Given a model SD and an observation α , the S-signature of SD $\wedge \alpha$, denoted as $S(\text{SD} \wedge \alpha)$, is defined as the distribution of the sizes of the sets $\{\omega^{\subseteq} : |\omega^{\subseteq}| = i\}$ for i = 1, 2, ..., |COMPS|.

Abusing the notation, in this paper we do not specify anymore SD, because it is fixed, i.e., we write $S(\alpha)$ instead of $S(\text{SD} \land \alpha)$.

To illustrate S, let the search space for an observation α on a system of size |COMPS| = 14consist of one $|\omega^{\subseteq}| = 1$ subset-minimal diagnosis, four $|\omega^{\subseteq}| = 2$ subset-minimal diagnoses and twentyone $|\omega^{\subseteq}| = 3$ subset-minimal diagnoses. There are no subset-minimal diagnoses of higher cardinalities. Then for this observation α , we write the following expression:

For brevity, we omit the trailing zeroes and write:

$$S(\alpha) = \{1, 4, 21\}$$

Consider the set of all possible observations α . Clearly, for a system of |OBS| Boolean observables there are $2^{|OBS|}$ different observations (in MBD we usually assume that there are no physically impossible observations and that all observations are of equal likelihood). There are 4 096 different observations for our running example (|OBS| = 12). We have computed all S-signatures for the 74180 model by using an exhaustive method. The set of S-signatures shows surprisingly little variation as is visible in Table 1. It



Figure 2: The 74180 circuit

MC			S			count
0	{0,	0,	0,	0,	0}	1024
1	{1,	4,	21,	0,	0}	128
	$\{2,$	7,	21,	0,	0}	256
	{2,	10,	0,	0,	0}	256
	{2,	21,	0,	0,	0}	256
	<i>₹</i> 3,	0,	0,	0,	0}	512
	{3,	14,	0,	0,	0}	256
	{3,	28,	0,	0,	0}	256
	{4,	21,	0,	0,	0}	256
	$\{7,$	8,	0,	0,	0}	256
	₹7,	9,	0,	0,	0}	256
2	{0,	9,	0,	0,	0}	256
	Į0,	11,	21,	0,	0}	128

Table 1: All possible S-signatures of the 74180 model

turns out that the example system contains only 13 different values of S.

Next consider all observations α leading to a certain signature $S(\alpha)$. In the following experiment on our example model 74180, we have configured the "greediness" parameter M of SAFARI to M = 5 (recall that M is the number of unsuccessful literal flips before SAFARI gives-up improving the current, possibly suboptimal, diagnosis). Figure 3 shows a box plot of the progress of SAFARI with all observations leading to a signature $S(\alpha) = \{1, 4, 21\}$. We have repeated the experiment 1 000 times to compensate the stochastic nature of SAFARI. As there are 128 observations leading to this signature (see Table 1), we have executed a total of 128 000 SAFARI runs for generating the data for this plot.

We have repeated the experiment with another signature $S(\alpha) = \{2, 10\}$. The results are shown in Fig. 4.

The plots in Fig. 3 and Fig. 4 show that the progress of SAFARI depends on the S-signature and is largely



Figure 3: Progress of SAFARI (M = 5) of all observations with $S = \{1, 4, 21\}$

independent of the actual composition of the observation α . For example, given a signature $S(\alpha) = \{1, 4, 21\}$, 50% of the observations lead to a diagnosis of cardinality 1 with probability between 0.31 and 0.37. Similarly, given a signature $S(\alpha) = \{2, 10\}$, 50% of the observations lead to a diagnosis of cardinality 2 with probability between 0.36 and 0.41.

Motivated by the above example we can make the main hypothesis of this paper:

Hypothesis 1. Given a system description SD, an *S*-signature $S(\alpha)$ is a statistically significant predictor of the performance of SAFARI.

3.2 High-Fidelity Model A₃

In what follows we assume that SAFARI is run only once (N = 1) and that it is configured to guarantee the computation of one subset-minimal diagnosis (M = |COMPS|).

We can express the state of SAFARI as an Ssignature. In the beginning, there is non-zero probability of reaching any of the subset-minimal diag21st International Workshop on Principles of Diagnosis



Figure 5: State space of SAFARI with a system of size |COMPS| = 6 and an observation of $S = \{0, 1, 2\}$ (terminating states are with thicker lines and gray background)



Figure 4: Progress of SAFARI (M = 5) of all observations with $S = \{2, 10\}$

noses in the S-signature. With each literal flip, SA-FARI makes zero or more subset-minimal diagnoses unreachable until there is only one remaining subsetminimal diagnosis. The whole state-space of SAFARI for a given system of size |COMPS| = 6 and an initial S-signature $\{0, 1, 2\}$ is shown in Fig. 5.

The progress of SAFARI in each run is expressed as k. Initially¹, k = |COMPS|. The state of SAFARI at

step k is denoted as S(k). Consider the change of the state of SAFARI transitioning from step k to step k-1. We denote this change as:

$$\dot{S}(k) = S(k) - S(k-1)$$

For example, if $S(3) = \{0, 1, 2\}$ and $S(2) = \{0, 0, 1\}$, then $\dot{S}(k) = \{0, 1, 1\}$.

Consider the general change of state of SAFARI:

$$\dot{S}(k) = \{\dot{s}_1, \dot{s}_2, \dots, \dot{s}_m\}$$

where m = |S|.

Assuming subset-minimal diagnoses do not share literals we have:

$$\max_{i} S_i(k) = 1$$

If we also assume that false literals are uniformly distributed, the probability of a change of state is:

$$p(k = \dot{S}_i) = \frac{i \cdot c_i}{k} \tag{1}$$

where *i* is the cardinality of the diagnosis that is made unreachable. Notice that *k* is the number of "unflipped" false literals at step *k*, initially k = |COMPS|.

In model A_3 we assume that the solutions that are present in S(k) are randomly positioned in the health vector. The probability of each traversal from state S(k) to state S(k-1) depends on the number of ways the new state can be reached, the probability that certain solutions are made infeasible, and the probability that the remaining solutions have no variables flipped. For example, suppose that the actual solutions to a

¹In practice, SAFARI starts from a random diagnosis ω that is constructed from the solution of a random satisfiable assignment of SD $\wedge \alpha$ but for this analysis we assume that SAFARI starts from the all-faulty assumption without loss of much generality.

state	h	$ \omega^{\subseteq} $		
$egin{array}{c} s_1 \ s_2 \ s_3 \end{array}$	001111 010011 101100	$\begin{aligned} \omega^{\subseteq} &= 2\\ \omega^{\subseteq} &= 3\\ \omega^{\subseteq} &= 3 \end{aligned}$		

Table 2: A sample initial configuration of SAFARI

problem of a system of |COMPS| = 6 are shown in Table 2.

The starting state for the initial configuration shown in Table 2 is $S(k = 6) = \{0, 1, 2\}$. Given this system and its starting state of Model A_3 , consider the traversal to state $S(k = 5) = \{0, 1, 1\}$. This is a decrease of S_3 by one, i.e. one of the $|\omega^{\subseteq}| = 3$ solutions, s_2 or s_3 , is made infeasible by flipping one of its variables. The number of solutions per cardinality that are made infeasible is given by:

$$S(k) - S(k-1) = \{0, 0, 1\}$$

In this case, either s_2 or s_3 is made infeasible, making the number of combinations equal to 2. In general, the number of ways one can choose the solutions that are made infeasible out of the complete set of solutions is given by

$$\prod_{i} \binom{S_i(k)}{S_i(k-1)}$$

where S_i is the number of solutions in S for cardinality $|\omega^{\subseteq}| = i$.

In order to make a $|\omega^{\subseteq}| = 3$ solution infeasible, either a variable of s_2 is flipped, or a variable belonging to s_3 . The probability of making any $|\omega^{\subseteq}| = 3$ solution infeasible given k = 6 is equal to $\frac{3}{6}$ since only one of its variables needs to be flipped in order for the solution to be impossible to be reached. In the case of the example, either h_1 , h_3 or h_4 needs to be flipped in order to make solution s_2 infeasible.

The probability that the remaining solutions (s_1 and the remaining $|\omega^{\subseteq}| = 3$ solution) have none of their variables flipped is equal to:

$$\left(1-\frac{2}{6}\right)\cdot\left(1-\frac{3}{6}\right) = \frac{1}{3}$$

Combining it all results in a probability $p = \frac{1}{3}$ that a variable flip results in reaching a state where one of the two $|\omega^{\subseteq}| = 3$ solutions is made infeasible, while the other C = 3 solution and the $|\omega^{\subseteq}| = 2$ solution remain feasible.

In general this probability is equal to:

$$p = \prod_{i} \left[\binom{S_i(k)}{S_i(k-1)} \left(\frac{i}{k}\right)^{S_i(k)} \left(1 - \frac{i}{k}\right)^{S_i(k-1)} \right]$$

To compute the probability of SAFARI reaching an end state S we have to sum all $p(k = \dot{S}_i)$ from the initial state of SAFARI to the end state. The major problem with this model is that there are exponentially many paths from the initial state of SAFARI to an end state. Consider, for example, Fig. 5. To end-up in state $\{0, 1, 0\}$, SAFARI can make a direct transition at step k = 6, or it can remain at $\{0, 1, 2\}$ until step k = 4 and then it may transition to the final state $\{0, 1, 0\}$.

3.3 Medium-Fidelity Model A₂

Based on model A_3 another model is created to minimize model complexity while abstracting as little as possible. The approach used in this model A_2 is to simulate a SAFARI run while keeping track of the *mean* intermediate S at each step of the algorithm. In this approach, the cardinality values within S(k) are mean values, averaged over all possible trajectories leading to S(k). The intermediate S(k) is expressed in terms of the intermediate S(k-1).

For example, consider a system of |COMPS| = 6components and an S-signature $S = \{0, 1, 2\}$, where the actual subset-minimal diagnoses are shown in Table 2. As with the real algorithm, the model starts with k = |COMPS|. S(6) contains one $|\omega^{\subseteq}| = 2$ diagnosis and two $|\omega^{\subseteq}| = 3$ diagnoses. Flipping a variable which causes s_1 to be unreachable has a probability of $\frac{2}{6}$. If this is done a large number of times, on average, $\frac{1}{3}$ of the $|\omega^{\subseteq}| = 2$ diagnoses would remain. The average number of $|\omega^{\subseteq}| = 2$ diagnoses, which is made impossible to reach by flipping one of the six variables, is equal to:

$$S_2 \cdot \frac{2}{k} = 1 \cdot \frac{2}{6} = \frac{1}{3}$$

In other words, on average, a third of the number of $|\omega^{\subseteq}| = 2$ subset-minimal diagnoses is made unreachable, because the variable that was flipped belonged to a $|\omega^{\subseteq}| = 2$ subset-minimal diagnosis. The same is done for $|\omega^{\subseteq}| = 3$, where the average number of subset-minimal diagnoses that is made impossible to reach is equal to

$$S_3 \cdot \frac{3}{k} = 2 \cdot \frac{3}{6} = 1$$

These average values are then subtracted from the previous S to obtain the average S at the new k, making $S(k-1) = S(k) - \{0, \frac{1}{3}, 1\} = \{0, \frac{2}{3}, 1\}$. The average S(k-2) is calculated in the same way, but it is based on the generated average S(k-1).

The above can be generalized in the following formula for updating the S-signature at step $k, k = |\text{COMPS}|, |\text{COMPS}| - 1, \dots, 1$:

$$S_i(k-1) = S_i(k) - S_i(k) \cdot \frac{i}{k}$$
 (2)

When the model is applied on a starting value of S a complete trace of averages of intermediate S is acquired. This equals the average number of subsetminimal diagnoses of each cardinality that would still be reachable at each step k of the actual SAFARI algorithm. The complete trace of the example is shown in Table 3.

Note, that, compared to model A_3 , the Markov chain has reduced to a 1-dimensional chain, greatly reducing model complexity. Table 4 shows example traces of the example system of size |COMPS| =

k		S	
$\begin{array}{c} 6\\ 5\\ 4\\ 3\\ 2\end{array}$	$\{ \begin{matrix} \{0, \\ $	$1.00, \\ 0.67, \\ 0.40, \\ 0.20, \\ 0.07, $	2.00 1.00 0.40 0.10 0.00
1	<i>{</i> 0,	0.00,	0.00

Table 3: Execution trace of model A_2 and the example from Table 2

14 and an observation with an S-signature S = $\{1, 4, 21\}$. Both traces are acquired by running SA-FARI multiple times and taking the average number of subset-minimal diagnoses at each k. The trace of model A_2 is very accurate when compared with the actual SAFARI algorithm. Unfortunately, however, this is only a valid model for M = 1, which is displayed on the left of Table 4. At the right hand side of the table is the trace of SAFARI when run with M = 5 on the same observation, averaged over 10 000 runs. Both traces show that at k = 14 the intermediate S is still equal to the original. The values of S at each of the cardinalities decrease with every next k, but we see that the value of S_3 decreases significantly faster than the others, which is because these relatively larger subsetminimal diagnoses are made unreachable with a larger probability. The difference in the two traces is that with M = 1 the numbers decrease to zero, while with a large enough M the values decrease to some value higher than zero at a k equal to the cardinality.

k	${\cal S}$ for ${\cal M}$	S for $M=5$			
14	{1.00, 4.00,	$21.00\}$	{1.00,	4.00,	21.00}
13	$\{0.93, 3.43,$	16.50}	<i>{</i> 0.93,	3.43,	16.49
12	$\{0.86, 2.90,$	12.70}	<i>{</i> 0.86,	2.90,	12.70}
11	$\{0.79, 2.42,$	9.52}	<i>{</i> 0.78,	2.42,	9.53}
10	$\{0.71, 1.98,$	6.92}	<i>{</i> 0.71,	1.98,	6.94}
9	$\{0.64, 1.58,$	4.85	<i>{</i> 0.65,	1.60,	4.89
8	$\{0.57, 1.23,$	3.23	<i>{</i> 0.58,	1.26,	3.32}
7	$\{0.50, 0.92,$	2.02	<i>{</i> 0.52,	0.97,	2.17
6	{0.43, 0.66,	1.15}	<i>{</i> 0.47,	0.75,	1.37
5	$\{0.36, 0.44,$	0.58	<i>{</i> 0.43,	0.58,	0.84
4	$\{0.29, 0.26,$	0.23}	{0.40,	0.46,	0.48
3	$\{0.21, 0.13,$	0.06}	{0.38,	0.39,	<u>0.26</u> }
2	$\{0.14, 0.04,$	0.00	{0.38,	<u>0.36</u> ,	0.00
1	$\{0.07, 0.00,$	0.00°	{0.38,	0.00,	0.00

Table 4: Averaged S over 10 000 runs of SAFARI on a 74180 observation with an initial $S = \{1, 4, 21\}$

The effect of the average number of subset-minimal diagnoses decreasing less when a larger value for M is chosen is explained by the fact that a variable flip is retried after an inconsistency. The algorithm is able to reach lower k, adding to the average number of subset-minimal diagnoses at that k. For M = 5 (i.e., exhaustive retrying) it turns out that the values of the average

number of subset-minimal diagnoses at k equal to the cardinality, which are underlined in Table 4, are exactly equal to the distribution of these cardinalities of the actual SAFARI algorithm. This leads us to the following theorem:

Theorem 1. If M = |COMPS|, then the trace of an ideal model A_2 contains the probability distribution of cardinalities on its diagonal.

Proof. The diagonal contains the average values of S for which the subset-minimal diagnosis cardinality is equal to the remaining number of unflipped variables (k = c). If SAFARI is run with a sufficiently large M, the intermediate set of reachable subset-minimal diagnoses contains only one subset-minimal diagnosis when a cardinality is returned. Because a maximum of one reachable subset-minimal diagnoses can remain at k = C, and minimality of subset-minimal diagnoses is guaranteed, the sum of the average number of subset-minimal diagnoses remaining at k = C for all C is equal to one. Therefore the diagonal contains the probability distribution of cardinalities which are returned by SAFARI.

Theorem 1 and Eq. (2) give us the probability $p(|\omega^{\subseteq}| = i)$ of SAFARI configured with M = |COMPS| computing a subset-minimal diagnosis ω^{\subseteq} of cardinality *i*:

$$p\left(\left|\omega^{\subseteq}\right|=i\right)=S_{i}\left(i\right) \tag{3}$$

The complexity of computing the A_2 model is low – all probabilities in Eq. 3 can be computed in time $O(|\text{COMPS}|^2)$.

3.4 Low-Fidelity Model A₁

Model A_3 and, less so, model A_2 try to follow the SAFARI algorithm step by step, calculating intermediate probabilities of each cardinality. The low level of abstraction of model A_3 results in a computationally complex model, with the number of possible states exponentially growing for larger M. Model A_2 is more abstract than A_3 resulting in lower computational cost. Only $\frac{1}{2}M^2$ calculations are required to compute the estimated probability distribution function (pdf). With the next model, A_1 , we will further decrease the computational complexity and have an even greater abstraction.

Where models A_3 and A_2 follow the steps of the SAFARI algorithm, lowering k at each step, model A_1 computes the probability of each cardinality directly from S. Again, we assume that M = |COMPS|.

Given an S-signature S, the non-normalized probability $n(|\omega^{\subseteq}| = i)$ of SAFARI returning a subsetminimal diagnosis of cardinality *i* is modeled as:

$$n\left(\left|\omega^{\subseteq}\right|=i\right) = \frac{S_i}{\binom{|\text{COMPS}|}{i}} \tag{4}$$

and the normalized probability $p(|\omega^{\subseteq}| = i)$ is:

$$p\left(\left|\omega^{\subseteq}\right|=i\right) = \frac{n_i}{\sum n} \tag{5}$$

where S_i is the number of subset-minimal diagnoses of cardinality *i* in $S(\alpha)$.

In A_1 , the probability of returning a diagnosis of cardinality *i* is determined by the number of diagnoses of this cardinality divided by the number of ways a diagnosis can be reached, independently of other diagnoses. This probability is decreased if more diagnoses exist in the search space, however, an $|\omega^{\subseteq}| = 1$ diagnosis has more impact than an $|\omega^{\subseteq}| = 3$ diagnosis.

4 AUTOMATIC TERMINATION OF SAFARI

In what follows we show how to use the prediction from the SAFARI model for terminating the greedy stochastic search.

Algorithm 1 computes an approximation to all cardinality-minimal diagnoses and acts as a wrapper around SAFARI. The idea is to first call SAFARI with a small N to compute an initial S-signature. This signature S is used to predict the probability p^{\leq} that SAFARI will compute a cardinality-minimal diagnosis. From this probability we compute a new value of N. The process is repeated until no new cardinality-minimal diagnoses are computed.

Algorithm 1 Use SAFARI to compute an approximation to all cardinality-minimal diagnoses

1: function AUTOSAFARI(DS, α , M, N')
2: returns a set of diagnoses
inputs: DS diagnostic system
α , term, observation
M, integer, climb restart limit
N', integer, initial number of runs
local variables: $\Omega, \Omega^{\subseteq}$, sets of diagnoses,
initially empty
S, S-signature
p^{\leq} , real, probability
B, integer, subsequent run
3: $\Omega \leftarrow SAFARI(DS, \alpha, M, N')$
4: while NEWDIAGNOSES ^{\leq} $(\Omega, \Omega^{\subseteq})$ do
5: $\Omega^{\subseteq} \leftarrow \Omega^{\subseteq} \cup \Omega$
6: $S \leftarrow \text{SIGNATURE}(\Omega^{\subseteq})$
7: $p^{\leq} \leftarrow \text{MODEL}(S)$
8: $B \leftarrow \lceil \frac{1}{p^{\leq}} \rceil$
9: $\Omega \leftarrow \hat{SAFARI}(DS, \alpha, M, B)$
10: end while
11: end function

Algorithm 1 works as follows. First SAFARI is called with a small number of runs, for example, N = 10 (line 3). The initial set of diagnoses is stored in Ω . The subroutine NEWDIAGNOSES returns **true** iff there are new diagnoses in Ω , compared to Ω^{\subseteq} . Clearly the first indication always returns **true**. The signature of the set of subset-minimal diagnoses Ω is computed in line 6. In line 7 we estimate a model based on the intermediate set of diagnoses (*S*-signature). Based on this model we compute a new value of the number of runs of SAFARI. The process is repeated until no new cardinality-minimal diagnoses are computed.

5 EXPERIMENTAL RESULTS

Figure 6 shows a comparison of SAFARI and models A_3 , A_2 , and A_1 for 74180 and different S-signatures.

When compared to the SAFARI output, model A_2 seems to model SAFARI more accurately that model A_3 , which serves as a simulator which should resemble SAFARI closest. This can be explained by the fact that model A_3 assumes random observations of S, while SAFARI is run on real observations on the system, which are considered outliers compared to the random observations. The output of model A_2 differs from the output of model A_3 because of the assumption that the variables of the solutions are assumed to be randomly distributed in the health vector and therefore does not account for solution subsumption. By chance, this difference transforms the resulting pdf to resemble the SAFARI outcome more that model A_3 . It would be more accurate to compare the model to the reference model A_3 , which stands as close as possible to the actual algorithm, but which excludes the information of the positioning of solution variables in the health vector. Looking at the output of the normalized version of model A_2 we see that the lower cardinalities have a slightly larger probability mass when compared to the output of model A_3 . Nonetheless, the difference between model A_3 and the normalized model A_2 is small, indicating that model A_2 is usable as a model for the behavior of the SAFARI algorithm. Both model A_3 and A_2 result in outputs with no probability mass at cardinalities of which there are no existing solutions.

6 CONCLUSIONS

In this paper we have explored three new performance models for the SAFARI algorithm, as an improvement on the original model, A_0 . All three models use S as input and return the pdf of solution cardinalities f_a .

First, we have discussed model A_3 , the least abstract model, which defines all possible transitions from one state (intermediate S) to all possible next states. Any state consisting of a single cardinality serves as ending state for the model. By calculating the probabilities of all state transitions, the probability of reaching each cardinality can be determined. However, this is a very expensive method and unrealistic for large systems, since the number of possible states grow exponentially for larger M. The Monte Carlo approach, traversing the model multiple times to achieve a pdf of solution cardinalities, is another possibility, however, this method consumes time and could suffer from lack of numerical precision.

Second, the more abstract model A_2 is introduced, which starts with a given S, and, with each step k, calculates the average S for that step in the algorithm. The resulting pdf is constructed by using the values on the diagonal of the trace of average S of the model. Since this computational complexity of this model is small $(O(M^2))$, and because this model seems to model the actual algorithm more accurately, this is a far more attractive model to use, compared to model A_3 .

Last, we have our most abstract model, A_1 , which computes the pdf directly (in a two-step approach) from the characteristic S. It calculates the probability of returning each cardinality, given only the number of

21st International Workshop on Principles of Diagnosis



Figure 6: Comparison of SAFARI output with models A_3 , A_2 , and A_1

components and the number of solutions of the same cardinality. After calculating this for all cardinalities, the pdf is calculated by normalizing the values, which models the influence that solutions of different cardinality have on each other. This is the computationally cheapest model and experiments show that it seems to model SAFARI more accurately than model A_2 . However, more extensive testing should be performed, with different systems and observations, to be able to decide which of the performance models is best at modeling SAFARI.

REFERENCES

- (Bajwa and Sweet, 2003) A. Bajwa and A. Sweet. The Livingstone Model of a Main Propulsion System. In *Proceedings of the IEEE Aerospace Conference*, 2003.
- (De Kleer and Williams, 1987) J. De Kleer and B.C. Williams. Diagnosing Multiple Faults. *Artificial Intelligence*, 32(1):97–130, 1987.
- (Feldman and van Gemund, 2006) A. Feldman and A. van Gemund. A Two-step Hierarchical Algorithm for Model-based Diagnosis. In *Proceeding of the National Conference on Artificial Intelligence*, volume 21, page 827. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006.
- (Feldman et al., 2008) A. Feldman, G. Provan, and A. Van Gemund. Computing Minimal Diagnoses by Greedy Stochastic Search. Proc. AAAI08, pages 919–924, 2008.
- (Feldman et al., 2010) Alexander Feldman, Gregory Provan, and Arjan van Gemund. Approximate model-based diagnosis using greedy stochastic

search. Journal of Artificial Intelligence Research, 38:371–413, 2010.

- (Janssen, 2011) Tom Janssen. Performance modeling of stochastic diagnosis engines. Master's thesis, Delft University of Technology, 2011.
- (Muscettola *et al.*, 1998) N. Muscettola, P.P. Nayak, B. Pell, and B.C. Williams. Remote Agent: To boldly go where no AI system has gone before. *Artificial Intelligence*, 103(1-2):5–47, 1998.
- (Williams and Nayak, 1996) B.C. Williams and P.P. Nayak. A Model-based Approach to Reactive Selfconfiguring Systems. In *Proceedings of the National Conference on Artificial Intelligence*, pages 971–978, 1996.
- (Williams and Ragno, 2007) B.C. Williams and R.J. Ragno. Conflict-directed A* and its Role in Modelbased Embedded Systems. *Discrete Applied Mathematics*, 155(12):1562–1595, 2007.