How to diagnose well with very little information

Johan de Kleer, Olivier Raiman
Xerox Palo Alto Research Center
3333 Coyote Hill Road, Palo Alto CA 94304 USA
Email: dekleer, raiman@parc.xerox.com

Abstract

Recently there have been a variety of algorithms and experimental results which suggest that the computational costs of lookahead and determining all candidate diagnoses have relatively little payoff. This paper analyzes the reasons why partial diagnostic information is often sufficient through a set of experiments. One of the surprising results of this paper is that one can perform diagnosis quite adequately with almost no information. This suprising result suggests that the new types of simple diagnostic algorithms which operate with partial information are likely to be of great utility.

1 Introduction

Much of the early research on model-based diagnosis focused on determining, as efficiently as possible, the candidate diagnoses which explain the observed symptoms. As the field progressed, research began focussing on not just determining which diagnoses explain the symptoms but also which measurement to make next. Given the set of candidate diagnoses, determining the next best measurement requires a complex and potentially computationally explosive lookahead and entropy calculation (such as in GDE [4]). As the field considered more complex examples, the computational cost of determining all the candidate diagnoses and the truely optimal next measurement was recognized as being too computationally expensive.

More recently there have been a variety of algorithms and experimental results which suggest that the computational costs of lookahead and determining all candidate diagnoses have relatively little payoff. Emperical studies of lookahead algorithms to determine the best next probe show that in most cases, one-step lookahead is almost optimal [6]. In fact, it can be shown that in many cases one-step lookahead is optimal [7]. Three recent results in model-based diagnosis demonstrate the success of diagnostic algorithms which operate with partial information about the diagnosis space. Sherlock [5] selects the next probe knowing only the best few diagnoses. Critical reasoning [11] is an alternative, fast, candidate generator which often cannot determine all minimal diagnoses yet diagnoses extremely well. Pragamatic reasoning [1] finds reasonable probes without full knowledge of the diagnosis space.

This paper analyzes the reasons why partial diagnostic information is often sufficient through a set of experiments and observations. One of the surprising results of this paper is that one can perform diagnosis quite adequately with almost no information. This suprising result suggests that the new types of simple diagnostic algorithms which operate with partial information are likely to be of great utility.

2 Diagnostic Algorithm

Our proposed algorithm still exploits one-step lookahead, but uses only a fraction of the full diagnosis space to determine the next best probe. We restrict the candidate generator to only construct k candidate diagnoses (k is typically a small number less than 10) and determine the best probe to make next based on these k candidates. We choose the k candidate diagnoses through a random selection process. The random selection process is guaranteed to select candidates in proportion to their probability. The diagnostic algorithm proceeds as follows:

1. Find k (if they exist) candidate diagnoses to be the current diagnosis set.
2. If $k = 1$, we are done.
3. Find the probe (using one-step lookahead and then entropy) which best discriminates among the current diagnosis set.
4. Perform the probe and eliminate all current diagnoses inconsistent with the result.
5. Find additional candidate diagnoses until the current diagnosis set is of size k again.
6. Go to Step 2.

This algorithm is surprisingly good. Even k's of 3 or 4 are very close to optimal for very large circuits. But before analyzing these results we must first explore our methodology for comparing different algorithms.

3 Experimental Methodololgy

In [6] we proposed a methodology for comparing diagnostic algorithms which we restate briefly here. The diagnostic task can be represented by a family of decision trees. The root node of each tree (Figure 1) represents a set of fault hypotheses generated by examining previously gathered evidence, e.g., the initial symptom. The

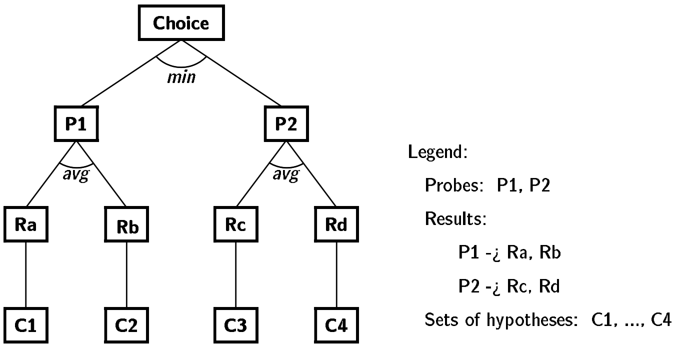

Figure 1: A family of decision trees.

Circuit	Type	Gates	Test Points
c432	priority decoder	160	432
c499	ECAT	202	499
c880	ALU and control	383	880
c1355	ECAT	546	1355
c1908	ECAT	880	1908
c2670	ALU and control	1193	2670
c3540	ALU and control	1669	4100
c5315	ALU and selector	2307	5315
c6288	Multiplier	2416	6288
c7552	ALU and control	3512	7552

Table 1: Circuits for Experimentation

next level represents alternative probes, and the next, their respective possible outcomes. Each leaf node represents the set of fault hypotheses that would result if the path from root to that leaf were taken. Different diagnostic algorithms incorporate different strategies for choosing the next probe—each yielding a particular decision tree.

The diagnostic cost of applying a particular probe selection strategy is simply the average depth of the decision tree resulting from using the probe selection algorithm (for the purposes of this paper we assume all measurement costs are equal).

In order to compare different selection strategies, we define the notion of *diagnostic cost* (DC). Diagnostic cost is a property of both the device and the algorithm used to diagnose it. Intuitively the diagnostic cost is the expected cost to diagnose the device given that it has a symptom. Thus, conceptually:

$$DC(D, A) = \Sigma_{s \in S} p(s) c(D, s, A),$$

where D is the device, A is the algorithm used, and $c(D, s, A)$ is the cost of diagnosing device D with algorithm A in situation s. A diagnostic situation is a fault plus an initial symptom, S is the set of possible diagnostic situations, and $p(s)$ is the probability of a given situation,

In order to estimate DC for complex devices in reasonable time, we make several simplifying assumptions in addition to the usual ones made within the GDE/Sherlock framework [4; 5]:

1. We assume single, stuck-at faults. Assuming single faults reduces the space of faults to a size we can sample experimentally.
2. We assume that all situations which cause a symptom to be manifest for a particular fault are equally difficult to diagnose. We then randomly select a few situations for each fault and measure its cost.
3. We assume that all probe costs are fixed and equal (not context dependent) and known precisely.

Under the assumption that we are only considering single faults, a diagnostic situation is a tuple $[f, v]$ where f specifies a faulty component behavior, and v is a sensitive vector (a tuple of input(s) and output(s) which causes D to manifest a symptom). Thus, diagnostic cost can be expressed as:

$$DC(D, A) = \sum_{f \in sf(D)} p(f) \times \sum_{v \in sv(f,D)} \frac{c(D, [f, v], A)}{|sv(f, d)|},$$

where $sf(D)$ are the single faults of D, the inputs under which f causes misbehavior, $sv(f, D)$ are the sensitive vectors v for which $[f, v]$ is a diagnostic situation, and $p(f)$ the probability of fault f given that some fault is known to be present. For a device with even a small number of inputs $|sv(f, D)|$ is very large, therefore under the assumption that all situations are equally likely we estimate DC by only sampling a few of the sensitive vectors.

Table 1 lists the combinatorial circuits we used to compare different diagnostic algorithms. The circuits named, Cn, are part of a standard test suite of combinatorial circuits [2].

4 Experimental Results

We applied our diagnostic algorithm for $k = 2, \ldots, 15$ and each of the circuits in the test suite. The number of samples was always at least $10,000$ (we hope to increase this for the final version of the paper). We report the results in a series of graphs which plot diagnostic cost against k. Figure 2 graphs the results of the experiment for circuit c432. Even $k = 2$ is close to optimal ($k = 15$ is essentially optimal in all our experiments). Diagnostic cost drops slowly with k. The standard deviation is relatively large and decreases slowly with k. The negligible change in mean diagnostic cost from $k = 2$ to $k = 3$ is a result of the fact that there still remain only two possible lookahead scores when there are 3 diagnoses (i.e., all interesting probes split the diagnoses in the same way: 1-to-2). Standard deviation always drops consistently with k because with larger diagnosis sets there is less chance of choosing bad probe points. The standard deviation provides interesting insight into the diagnostic approach, for example, if the diagnostic session must be

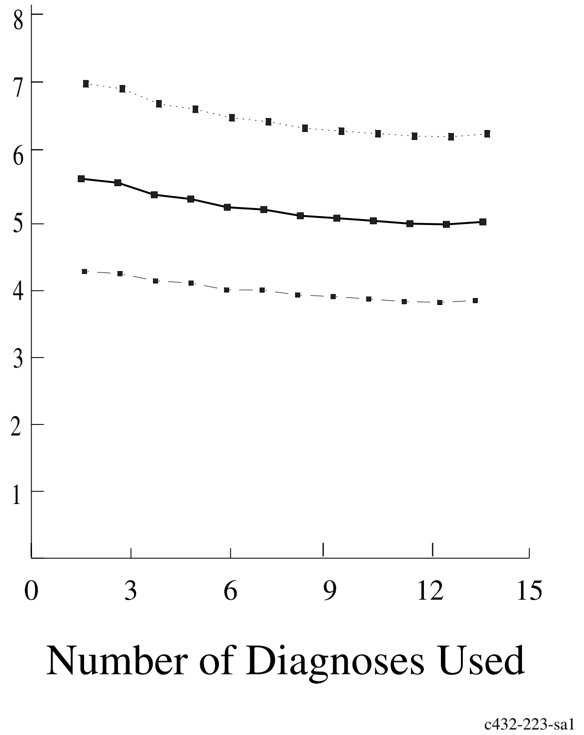

Figure 2: Scores of C432. All faults have equal probability. The middle curve is the mean diagnostic cost. The difference betweeen the top and bottom curves is the standard deviation.

finished under a certain cost (within a certain period of time perhaps), we can calculate the probability of identifying the fault within that cost constraint.

Figure 3 illustrates the results on the larger circuit c880. There we see k dropping quickly by $k = 4$ and the standard deviation dropping with k.

Figure 4 describes the circuit c499. The curves for all of the circuits in the test suite look similar. Even for the largest circuit c7552 (see Figure 7), which has large initial diagnosis sets, demonstrates exactly the same behaviors as the others.

We have also experimented with varying fault probabilities. This has little consequence on the shape of the curves. Varying cost of probes significantly changes the curves making small k far less optimal.

5 Analysis of Experimental Results

The main result of the experiment is that extremely small k are adequate for even very large circuits. We do not have a good underlying theory of why this is the case. One possibility is that there simply are a significant number of good probes available at every point in the diagnostic session. Therefore, a few randomly chosen diagnoses will usually pinpoint one of the good probes. This is consistent with the observation in [6]

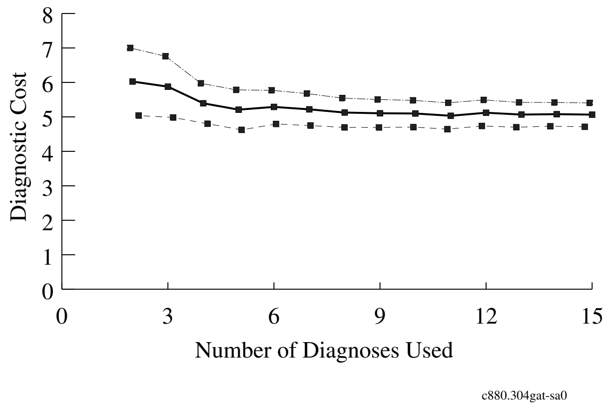

Figure 3: Scores of C880. All faults have equal probability.

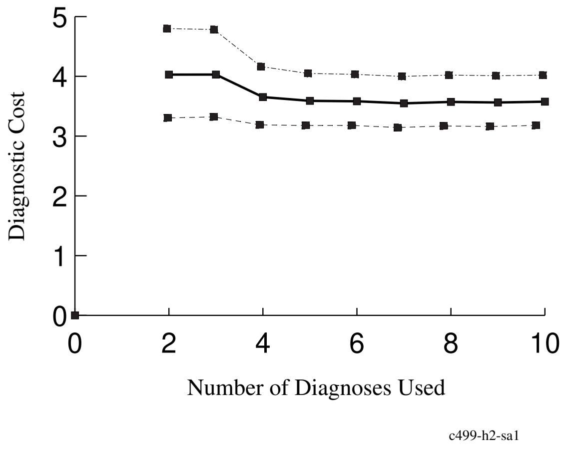

Figure 4: Scores of C499. All faults have equal probability.

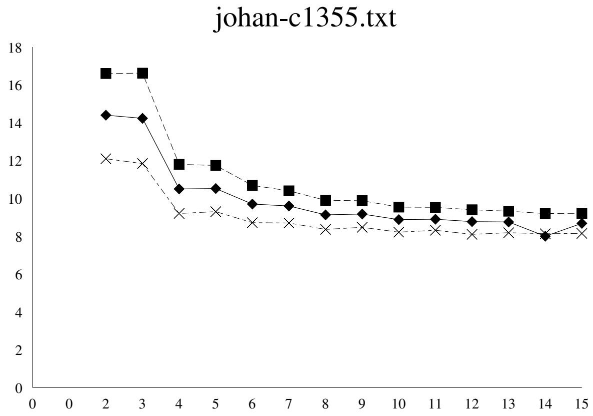

Figure 5: Scores of C1355. All faults have equal probability.

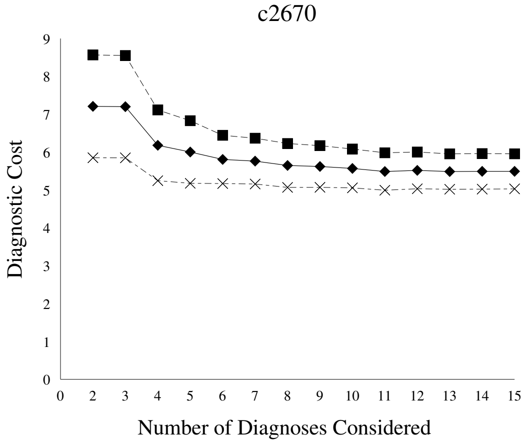

Figure 6: Scores of C2670. All faults have equal probability.

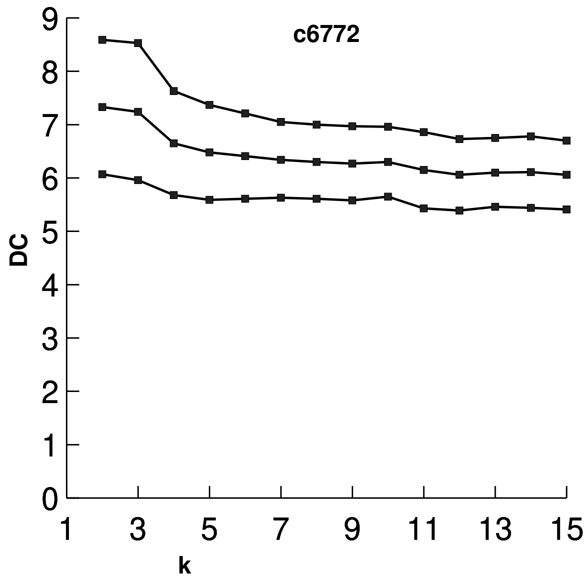

Figure 7: Scores of C7552. All faults have equal probability.

that randomly choosing amongst all probes which provide any discrimination among the complete diagnostic set is surprisingly good.

Nevertheless the results of this experiment place in crisper perspective why some existing, incomplete, diagnostic algorithms perform as well as they do. For example, Sherlock [5] which identifies the best few diagnoses is nearly optimal. Similarly, critical reasoning [11] which sacrifices diagnostic completeness for computational efficiency proposes remarkably good probes. It is also then of little surprise that Bakker's [1] more syntatic probe selection process is also very good. In general, it seems likely that any diagnostic algorithm which somehow characterizes even a little of the diagnostic space yields a good diagnostician. Of course, it is possible to conjure algorithms which do very badly. For example, an algorithm which simply steps from the symptom backwards is systematically choosing very bad probe points and performs relatively badly.

6 Consequences for Algorithm Design

Our experiments explain why some existing algorithms perform surprisingly well. However, can the insights lead to new, more computationally efficient, diagnostic algorithms? When k's are small, all the cost is generating the candidates—how can an algorithm exploit the fact it does not have to generate all candidates? If one uses a GDE-like approach, a large computational cost is simply the dependency-recording simulation. We have experimented with the following approach based on generate-and-test.

Generate-and-test candidate generation algorithms use conventional (non-TMS) simulators to generate the behavior of all possible candidates. A candidate becomes a diagnosis if its predictions are consistent with the observations. Generate-and-test candidate generators have fallen out of favor because they are inherently inefficient as they require a full simulation of every possible fault. GDE-like approaches use conflict sets to quickly focus the candidate generator. However, they perform best when presumes computation is free and probes are infinitely expensive. This is rarely the case.

We are experimenting with the following diagnostic algorithm.

1. Use generate-and-test find k candidate diagnoses (selected randomly in proportion to their probability).
2. Stop if $k = 1$.
3. Find and make the probe which best discriminates among the current diagnoses.
4. Eliminate the current diagnoses which are inconsistent with the probe result.
5. Using generate-and-test make sure there are k current diagnoses (if they exist).
6. Go step 2.

Initial indications are that this algorithm performs very well. The fundamental reason for this is that the simulations become progressively more efficient as more evidence becomes available. The following estimation of running time explains the good performance. Let S be the running time of a single simulation of the device, d be the number of components in the device, c be the number of diagnoses consistent with the initial symptom. We presume all fault probabilities are equal. The cost of finding each diagnosis is, on average, $\frac{Sd}{c}$. Therefore, the cost of Step 1 of the algorithm is $\frac{Sdk}{c}$. For digital circuits, a probe will eliminate $\frac{k}{2}$ diagnoses, thereby requiring finding $\frac{k}{2}$ additional diagnoses at Step 5. Generate-and-test now has to do twice as many simulations to find those diagnoses at total cost $\frac{Sdk}{c}$. So far the cost is the same. However, our simulator, on average, doubles its performance with each new observation. As the number of simulations goes up, the cost of simulation goes down. It will require about $logc$ steps to find the single offending diagnosis. The total running time of the algorithm is thus approximately:

$$\frac{Sdk[logc + 2]}{c}.$$

If c (the number of initial candidate diagnoses) is small, this algorithm will perform relatively poorly. For example, if $c = 1$ then it will simulate, expensively, every possible fault. However, if c is large this algorithm very effficient. This has the reverse performance profile of GDE-like conflict-based algorithms which do best when the initial conflict is small.

Our initial experimentation with this algorithm supports our theoretical analysis. In the full version of the paper we will report on the computation times of this algorithm on this test suite.

References

[1] Bakker, R.R., M. Bolurseau, and M.K. de Weger, Pragmatic reasoning in model-based diagnosis, in: Proceedings of the Third International Workshop on Principles of Diagnosis, 1982, 107–116.

[2] Brglez, F., and H. Fujiwara, A neutral netlist of 10 combinational benchmark circuits and a target translator in FORTRAN, distributed on a tape to participants of the Special Session on ATPG and Fault Simulation, Int. Symposium on Circuits and Systems, June 1985; partially characterized in F. Brglez, P. Pownall, and R. Hum, Accelerated ATPG and fault grading via testability analysis, *Proc. IEEE Int. Symposium on Circuits and Systems*, (June, 1985) 695–698.

[3] de Kleer, J., Using crude probability estimates to guide diagnosis, *Artificial Intelligence* **45** (1990) 381–391.2

[4] de Kleer, J. and Williams, B.C., Diagnosing multiple faults, *Artificial Intelligence* **32** (1987) 97-130. Also in *Readings in NonMonotonic Reasoning*, edited by Matthew L. Ginsberg, (Morgan Kaufman, 1987), 372–388.

[5] de Kleer, J. and Williams, B.C., Diagnosis with behavioral modes, in: *Proceedings IJCAI-89*, Detroit, MI (1989) 104–109.

[6] de Kleer, J., Raiman, O. and Shirley, M.H., One step lookahead is pretty good, in: Readings in Model-Based Diagnosis, edited W. Hamscher, J. de Kleer and L. Console, (Morgan Kaufmann, 1992), 138–142.

[7] de Kleer, J., Raiman, O., One step lookahead is optimal, forthcoming.

[8] Freitag, H., A Generic Measurement Proposer, in: G. Gottlob and W. Nejdl (Eds.), *Expert Systems in Engineering: Principles and Applications*, (Springer-Verlag, Berlin, 1991) 79-89.

[9] Horvitz, E., J. Breese, and M. Henrion, *Decision Theory in Expert Systems and Artificial Intelligence*, Technical Report KSL-88-13, Stanford University Knowledge Systems Laboratory, February, 1988.

[10] Russell, S. and E. Wefald, On Optimal Game-Tree Search using Rational Meta-Reasoning, in: *Proceedings IJCAI-89*, Detroit, MI (1989) 334–340.

[11] Raiman, O., J. de Kleer, and V. Saraswat, Critical Reasoning, in *Proceedings IJCAI-93*.