# A Final Determination Of the Complexity Of Current Formulations Of Model-Based Diagnosis (Or Maybe Not Final?)

MS-CIS-91-13
LINC LAB 194

Ron Rymon

Department of Computer and Information Science
School of Engineering and Applied Science
University of Pennsylvania
Philadelphia, PA 19104-6389

Revised
June 1991

# A Final Determination of the Complexity of Current Formulations of Model-Based Diagnosis (Or Maybe Not Final?)

Ron Rymon
Department of Computer and Information Science
University of Pennsylvania, Philadelphia, PA [*]
rymon@linc.cis.upenn.edu

### Abstract

There are three parts to this paper. First, I present what I hope is a conclusive, worst-case, complexity analysis of two well-known formulations of the Minimal Diagnosis problem – those of [Reiter 87] and [Reggia et al 85].

I then show that Reiter's conflict-sets solution to the problem decomposes the single exponential problem into two problems, *each* exponential, that need be solved *sequentially*. From a worst case perspective, this only amounts to a factor of two, in which case I see no reason to prefer it over a simple generate-and-test approach. This is only emphasized with the results of the third part of the paper.

Here I argue for a different perspective on algorithms, that of expected, rather than worst-case performance. From that point of view, a sequence of two exponential algorithms has lesser *probability* to finish early than a single such algorithm. I show that the straightforward generate-and-test approach may in fact be somewhat attractive as it has high probability to conclude in a polynomial time, given a random problem instance.

## 1 Overview

Two well-known formulations of model-based *first principles diagnosis* are [Reiter 87, Reggia et al 85]. In the first part of this paper, after presenting their problem formulations, I present what I hope is the *final* word on their complexity.

Both problems are widely believed to be NP-Hard and the intractability of the latter was proved by both [Reggia et al 85] and [Allemang et al 89]. However, NP-Hardness is hardly a complete characterization of a problem's actual complexity. All it says is that the problem is at least as hard as any NP-Complete problem. This paper provides a final accord for such complexity analysis in that it presents *same* upper and lower bounds for the two problems. Showing that both problems may require exponential output, it excludes them from the class of NP problems – those that can be solved non-deterministically in a polynomial time.

While NP-Complete problems are already hard enough, I urge the reader not to discount a problem's amenability to non-deterministic algorithms. There are at least two reasons to distinguish NP (even complete) problems from exponential ones. The first is theoretic and has to do with the fact that, as of 1990, nobody has yet proven that NP≠P. The second is more practical, NP problems may be approximated (for example, using randomization) so as to achieve probabilistic results while any problem that requires exponential output is not amenable to such techniques.

[*]This research was supported by the Army Research Office, ARO Grant DAAL03-89-C0031PRI. I would like to thank Greg Provan, Sanguthevar Rajasekaran and Bonnie Webber for comments and discussion of earlier drafts of this paper.

The second part of this paper shows that, at least on the surface[1], the indirect approach to diagnosis generation taken by [Reiter 87] essentially decomposes the diagnosis problem into two subproblems, *each* of exponential complexity that have to be solved sequentially.

From the point of view of worst case complexity analysis this is just a factor of two. In this case, I only argue that [Reiter 87] (as well as [de Kleer and Williams 87]) do not show why their algorithm is superior to the simple generate-and-test method. However, from a probabilistic point of view, the need to solve two worst-case exponential problems sequentially may decrease the number of instances that are solved in polynomial time.

Indeed, I think that worst-case analysis should not be the final word in analyzing AI algorithms since most, if not all, AI problems have proven to be NP-Hard, exponential, or even undecidable. If the runtime performance of those algorithms is to be compared to each other, new metrics must be determined. One such metric, I argue, is the likelihood that such an algorithm finishes in a reasonable time, given a random problem instance. The third part of this paper shows that an almost straightforward generate-and-test approach to solve a problem of the type defined by [Reiter 87] possesses some good probabilistic properties of that type.

## 2 First Principles, Static, Diagnosis of Multiple Faults

Research in first principles diagnosis seeks generally applicable paradigms for representing and solving diagnostic tasks. Such paradigms do not rely on heuristics, empirical evidence or modelling of human expertise. In what follows, I only consider the static (concurrent) setting, in which one can only use information available at the outset. No new measurement or observations are allowed as diagnosis is carried out.

Section 2.1 outlines a formal definition of the Minimal Diagnosis problem as proposed by [Reiter 87]. Section 2.2 outlines Reiter's solution. Section 2.3 outlines the problem definition proposed by [Reggia et al 85]. Before proceeding, note that many of the definitions in Reiter's formulation rely on consistency of collection of formulae. While in general, consistency checking is undecidable, it is feasible in many practical settings. Time for consistency checking is totally disregarded in the following analysis. Since it is used to a similar extent in the two algorithms compared, this has little or no effect on the comparison made.

### 2.1 Problem Definition [Reiter]

**Definition 2.1** A System [Reiter]

*A system is a pair* (SD,COMPS) *where:*

*1.* SD *– the system description, is a set of first order sentences;*

*2.* COMPS $\overset{\text{def}}{=}$ $\{\, c_i\}_{i=1}^{n}$ *– the system's components, a finite set of constants.*

**Definition 2.2** Observations [Reiter]

OBS *– Observations comprise another set of first order sentences which describe* current *observations that are independent of the system's structure. The triple* (SD,COMPS,OBS) *denotes a system with current observations.*

**Definition 2.3** A Diagnosis

*Given* (SD,COMPS,OBS), *a* diagnosis *is a set of components whose abnormality (all other components assumed normal) explains* OBS. *Formally, let* AB *be a predicate designating abnormal components, then* $\Delta \subseteq$COMPS *is a diagnosis iff* SD$\cup$OBS$\cup\{$ AB$(c) \mid c \in \Delta\} \cup \{\neg$AB$(c) \mid c \in$ COMPS$-\Delta\}$ *is consistent. A diagnosis is* minimal *if it has no proper subset that is also a diagnosis.*

[1] I believe this not to be a final determination as it seems to me that conflict sets may prove useful in easily decomposable systems.

**Definition 2.4** The Minimal Diagnoses Problem

*Given* (SD,COMPS,OBS), *find* all *minimal diagnoses.*

## 2.2 Conflict Sets – Approaching Diagnosis Indirectly

To solve the Minimal Diagnoses problem, [Reiter 87] suggests an indirect approach: instead of dealing directly with candidate diagnoses, it introduces conflict sets – sets of components that cannot all be functioning normally. Reiter then describes the relationship between diagnoses and conflict sets and exploits this relationship to derive an algorithm for diagnosis.

**Definition 2.5** Conflict Set [Reiter]

*A conflict set is a set of components that, given* SD *and* OBS, *cannot* all *be functioning correctly. Formally,* $\mathrm{X}\subseteq\mathrm{COMPS}$ *is a conflict set iff* $\mathrm{SD}\cup\mathrm{OBS}\cup\{\neg\mathrm{AB}(c)\mid c\in\mathrm{X}\}$ *is inconsistent.*

**Definition 2.6** Hitting Set [Reiter]

*Let F be a collection of sets of components. A* hitting set *for F is a set that shares at least one component with each of F's sets. It is* minimal *if it has no proper subset that is also a hitting set.*

**Theorem 2.1** On The Relationship Conflict Sets–Diagnosis [Reiter]

$\Delta\subseteq$ COMPS *is a minimal diagnosis for* (SD,COMPS,OBS) *iff* $\Delta$ *is a minimal hitting set for the collection of conflict sets for* (SD,COMPS,OBS).

**Algorithm 2.2** Reiter's Conflict Sets Algorithm

1. *Find F, the collection of all minimal conflict sets for* (SD,COMPS,OBS).
2. *Find minimal hitting sets for F.*

Due to theorem 2.1, all minimal hitting sets are precisely all minimal diagnoses one looks for.

## 2.3 Problem Definition [Reggia, Nau and Wang]

[Reggia et al 85] casts the diagnosis problem as a set covering problem:

**Definition 2.7** *A Diagnostic Problem*

*A diagnostic problem P, is a 4-tuple* $\langle D, M, C, M^{+}\rangle$ *where*
$D = \{d_1, \ldots, d_k\}$ *is a finite set of disorders;*
$M = \{m_1, \ldots, m_n\}$ *is a finite set of manifestations;*
$C \subseteq D \times M$ *is a relation indicating manifestation and causality; and*
$M^{+} \subseteq M$ *are the observed manifestations.*

**Definition 2.8** *Manifestations*

*The set C defines each disorder's manifestations:* $man(d_i) = \{m_j \mid \langle d_i, m_j\rangle \in C\}$. *Similarly, if X is a set of disorders,* $man(X) = \cup_{d_i\in X}\ man(d_i)$.

**Definition 2.9** *Explanation*

*For any diagnostic problem* $P = \langle D, M, C, M^{+}\rangle$, $E \subseteq D$ *is an explanation for* $M^{+}$ *if* $M^{+} \subseteq man(E)$*; and E is parsimonious (i.e. minimal w.r.t. set inclusion).*

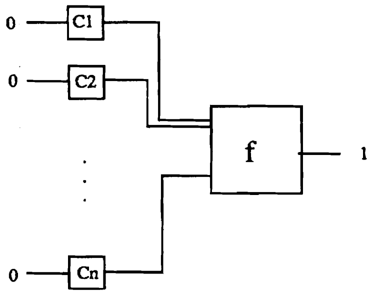
**Definition 2.10** *Solution to a Diagnostic Problem*

*A solution for* $P \overset{\mathrm{def}}{=} \langle D, M, C, M^{+}\rangle$, *denoted* $Sol(P)$ *is the set of* all *explanations for* $M^{+}$.

## 3 Minimal Diagnoses and Intractability

In this section I show that the Minimal Diagnoses (MD) problem (both definitions) are exactly exponential. Moreover, I show that even finding one diagnosis, of minimal cardinality, is still NP-Hard (though within NP). Interestingly, even though the problem definitions are quite different[2], they still share same complexity.

**Theorem 3.1** *The Minimal Diagnoses problem [Reiter 87] is at least exponential.*



$$f(k_1, k_2, \cdots k_n) = \begin{cases} 1 & \textit{if}\; k_1 + k_2 + \cdots + k_n \geq \frac{n}{2} \\ 0 & otherwise \end{cases}$$

Figure 1: An example circuit.

**Proof.** Consider the circuit in figure 1: $n$ transmitters ($c_1 \cdots c_n$) are linked to a single $f$-box. When operating normally, a transmitter's output is identical to its input and $f$-box's function is as defined above.

Minimal diagnoses for this circuit are:
(a) $f$-box alone fails; and
(b) any set of *half* the $n$ transmitters fail.

By induction, one can show that there are over $\frac{2^n}{n+1}$ diagnoses of the latter form. Thus for an algorithm to display them (disregarding the need to compute them) requires at least exponential run time. $\boxed{\text{QED}}$

**Theorem 3.2** *The Minimal Diagnosis problem [Reiter 87] is at most exponential.*

**Proof.** Any algorithm that generates-and-tests the whole power set of COMPS will do. $\boxed{\text{QED}}$

**Corollary 3.3** *The Minimal Diagnosis problem [Reiter 87] is* exactly *exponential.*

At this point, one may argue for relaxing the problem to finding a single diagnosis. I next show that if one looks for a diagnosis of minimal cardinality, one remains with an NP-Hard problem.

**Definition 3.1** The Minimum Cardinality Diagnosis Problem

*Given* (SC,COMPS,OBS)*, find a single diagnosis* $\Delta$ *of least cardinality.*

**Theorem 3.4** *The Minimum Cardinality Diagnosis problem is NP-Hard (but within NP).*

[2] In fact, Reiter shows that his formulation is more expressive in the sense that all diagnoses captured by the formalism of [Reggia et al 85] will also be captured by his formalism.

**Proof.** Let $\pi(MCD)$ be the corresponding decision problem. It is easy to show that $\pi(MCD)$ is in NP, and that a well-known NP-Complete problem called Vertex Cover can be reduced to it. $\boxed{\text{Q.E.D}}$

Now let us examine the formulation of [Reggia et al 85]:

**Theorem 3.5** *The Minimal Diagnosis problem [Reggia et al 85] is at least exponential.*

**Proof.** Consider a problem in which each manifestation is linked to a set of two disorders, and all these sets are pairwise disjoint. Such a problem has exactly $2^n$ solutions, corresponding to any selection of one out of every pair of disorders. $\boxed{\text{Q.E.D}}$

**Theorem 3.6** *The Minimal Diagnosis problem [Reggia et al 85] is at most exponential.*

**Proof.** Again obvious, by enumeration. $\boxed{\text{Q.E.D}}$

**Corollary 3.7** *The Minimal Diagnosis problem [Reggia et al 85] is exactly exponential.*

## 4 The Conflict Sets Algorithm and Intractability

Having shown that the Minimal Diagnoses problem has *no* worst-case polynomial-time algorithm, I can now show that Reiter's conflict sets algorithm may be especially troublesome as it gives rise to *two* new problems that have to be solved sequentially: first, the generation of *all* minimal conflict sets and only then identification of minimal hitting sets for this collection of minimal conflict sets[3]. Both problems, I show to be *exponential* in the worst case, making each, individually, just as hard as the original MD problem. Unless shown differently, having to solve these two subproblems *sequentially*, the conflict sets algorithm should be treated as putting one in great disadvantage.

The exponential complexity of the two subproblems will be proved first. Then, I will show that computing minimal hitting sets may be exponential even in cases where the number of conflict sets is polynomial.

**Theorem 4.1** *The number of minimal conflict sets may be exponential.*

**Proof.** Consider figure 1 again. The minimal conflict sets for this circuit are exactly *all* sets containing $f$-box and a combination of half the $c_i$'s. Again, there are exponential number of such sets. $\boxed{\text{Q.E.D}}$

**Theorem 4.2** *The number of minimal hitting sets for a collection of minimal conflict sets may also be exponential in the number of components.*

**Proof.** Once again, consider the example of figure 1. The set $\{f\text{-box}\}$ is one minimal hitting set for the collection of minimal hitting sets. The other minimal hitting sets are exactly *all* sets containing any combination of $\frac{n}{2}+1$ of the $c_i$'s. Once again, there are $O(2^n)$ such sets. $\boxed{\text{Q.E.D}}$

One may argue that the latter proof relies on a very particular setting in which the number of minimal conflict sets is also exponential. I next show that even if one was lucky, and the problem at hand had only a polynomial number of minimal conflict sets, the number of minimal hitting sets may still be exponential.

**Theorem 4.3** *Even if the number of minimal conflict sets is exponential, the overall complexity of the conflict sets algorithm may still be exponential.*

[3] In [Reiter 87] it is suggested that there is no need to compute all conflict sets in advance. While this might be an advantage in practice (facilitating parallelism for instance), it is important to note that access to *all* minimal conflict sets is required before *any* diagnosis can be confirmed.

**Proof.** Consider a case in which $2n$ distinct components are evenly distributed among $n$ conflict sets. Exactly $2^n$ minimal hitting sets exist, corresponding to any choice of one component from each set. $\boxed{\text{Q.E.D}}$

To summarize, the indirect approach replaces *one* problem of worst-case exponential complexity with *two* such problems that need be solved *sequentially*. To justify that, one must come up with algorithms that at least carry some good probabilistic properties (for example, that perform well in most cases, or on average). While this cannot be found in [Reiter 87], it is precisely what will be shown for the algorithm suggested next.

# 5 A Simple, Yet Probabilistically Tractable Algorithm

As I first noted in the introduction, the bare fact that a problem is exponential in the worst case should not prohibit researchers from looking for algorithms that *usually* perform well. Such an effort may take several different forms: [Bylander et al 89] have, with some encouraging success, *constrained the problem* so as on one hand it becomes tractable, while on the other hand, remains capable of expressing problems of interest. An equally attractive approach, presented next, is to find algorithms that given some distribution on problem instances, has *high probability* to perform well (i.e finish in polynomial time).

Before introducing the conflict sets algorithm, Reiter argues against the simple approach of generating-and-testing candidate diagnoses: "There is a direct generate-and-test mechanism based upon proposition 3.4: Systematically generate subsets $\Delta$ of COMPS, generating $\Delta$s with minimal cardinality first, and test the consistency of $\text{SD} \cup \text{OBS} \cup \{\neg \text{AB}(c) \mid c \in \text{COMPS} - \Delta\}$. The obvious problem with this approach is that it is too inefficient for systems with large number of components" ([Reiter 87], p. 67). In contrast, I show that this extremely simple approach does have some attractive probabilistic properties.

[de Kleer and Williams 87] uses probability-based refinements to curb the exploration of alternative *conflict sets*. In the subsequent discussion, I use some of those refinements to further cut the number of candidate diagnoses considered before the actual failure is reached. However, I concede that a number of additional improvements (possibly even out of those suggested by [de Kleer and Williams 87]) have to be made to the algorithm if it is ever to be used.

## 5.1 Failures Are Most Often Small in Size

Practically, I argue that it rarely is the case that a large number of components all fail at the same time. In electronic boards containing hundreds if not thousands of components, failures are usually detected after only a few components fail. Failures of small cardinality are also more amenable to repair than replacement of a whole unit or device. Small diagnoses thus play a rather major role in diagnostic problems.

Unfortunately, the conflict sets algorithm cannot take advantage of this fact. While it is true that in dealing with small problems, it would only be forced to explore its Hitting-Sets tree[4] to a relatively shallow level, this does not necessarily imply savings in runtime, as the number of conflict sets that need be generated is not directly related to the depth of the tree. As I have just shown, even if the number of conflict sets is polynomial, an exponential number of minimal hitting sets may still need be discovered.

Both Reiter [de Kleer et al 90] and others [Poole and Provan 90] have recently pointed out that diagnosticians should not limit themselves to minimal diagnoses. That is *not* to say that minimal diagnoses are not a particularly interesting class of diagnoses, but rather that it is possible that a failure may not be minimal. To avoid this discussion, I use the *actual* failure as a target. I do not assume that the actual failure is known, only that it can be recognized once it has been suggested by the algorithm.

**Definition 5.1** The Actual Failure

*Let* FAILURE $\subseteq$ COMPS *denote the* actual *set of failing components.*

[4] This has to do with the particular way in which a hitting set is discovered in [Reiter 87]. I did not describe the hitting set tree construction algorithm as it does not have major impact on our claims.

Before introducing the generate-and-test algorithm, I will introduce some facts that help establish its probabilistic properties. I will begin with some simplifying assumptions and then show how some of those may be lifted.

**Assumption 5.1** *All components have an equal probability to fail.*

**Assumption 5.2** *Components fail independently from each other.*

The following facts can be easily inferred:

1. The cardinality of the actual failure, as a random variable, is distributed binomially $B(n, p)$, $n$ being the number of components and $p$ the probability of failure for each component.
2. The probability that k or more components are concurrently faulty is thus:

$$\text{Prob}(|\text{FAILURE}| \geq k) = \sum_{i=k}^{n} \binom{n}{i} p^i q^{n-i} \text{ , where } q \overset{\text{def}}{=} 1 - p.$$

To estimate these sums, one can use the normal approximation. By the Law of Large Numbers, $|\text{FAILURE}| \sim N(np, npq))$ and this approximation is regarded excellent when both $np$, and $nq$ are greater than 5.

Chernoff [Augluin and Valiant 79] established upper bounds for such sums. These are accurate (not an approximation), but are many times looser than those provided by the normal distribution. Applying Chernoff bounds, for any $k > np$,

$$\text{Prob}(|\text{FAILURE}| > k) \leq (\tfrac{np}{k})^k e^{k-np}$$

**Assumption 5.3** *Assume further that the* average *number of concurrently failing components is bound by some* constant *d.*

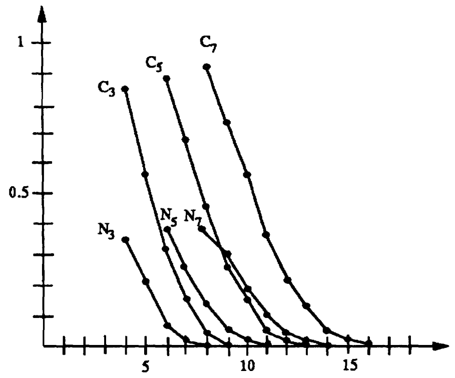
**Proposition 5.4** *Let $w > 1$ represent any chosen factor. The probability for a failure of size larger than $wd$ may be approximated by:*

*Prob*$(|\text{FAILURE}| > wd) \approx \Phi(\frac{|FAILURE| - np}{\sqrt{npq}}) = \Phi(\frac{|FAILURE| - d}{\sqrt{d - d^2/n}})$, *using the Normal approximation.*

*It can otherwise be bounded (not necessarily tightly) by:*

*Prob*$(|\text{FAILURE}| > wd) \leq (\frac{e}{w})^{wd} e^{-d}$, *using Chernoff bounds.*

**Example 5.5** *Given a circuit with n=100 components, consider average failures of 3, 5, and 7 components, the following graphs show the probability for failures of size larger than the average.*



$N_i$ – *Normal approximation for an average failure of size i.*
$C_i$ – *Chernoff bounds for an average failure of size i.*

Figure 2: Probabilities for failures of different sizes

## 5.2 Algorithm

The following algorithm takes advantage of the observations just made. The *Next-Set* routine in the algorithm is responsible for generating candidate diagnoses. As a first approximation, I assume that sets of minimal cardinality are generated first. Step 2 may be implemented directly based on the definition of a diagnosis.

**Algorithm 5.6**

```
Until no more sets, do
    1. $\Delta \leftarrow$ Next-Set ().
    2. Check if $\Delta$ is a diagnosis.
```

## 5.3 Complexity Analysis

The suggested algorithm is obviously simpler than the conflict-sets algorithm. Not only does it eliminate the need to solve two hard problems, but also consistency checking for a candidate diagnosis is much easier than for conflict sets. The reason is that in the case of a diagnosis, consistency is checked with AB literals instantiated for *all* components, while when checking inconsistency for a conflict set, only those components in the conflict set itself are instantiated, leaving all permutations of the rest as open possibilities.

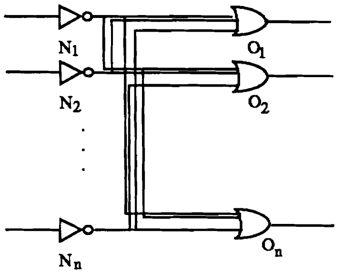
It is obvious that in the worst case, the algorithm would have to traverse all sets of $b \overset{\text{def}}{=} |\text{FAILURE}|$ or less components. Section 5.1, however, has shown that chances are very high that $b$ is no more than a small constant. The number of traversed sets is thus well bound by $O(n^b)$[5].

[5] In fact, as $b$ grows, the number of sets is significantly lower than $O(n^b)$. Recall however that to consider actual run time one has to factor the time required for each invocation of the consistency checking step.

## 5.4 Where This Algorithm Fails

The above algorithm does not solve *all* problems in a practically feasible time since, as I have shown, the problem is itself exponential. Next, I present an example in which the number of candidate diagnoses tested by algorithm 5.6 before hitting the *first* diagnosis is also exponential.

Consider the circuit in figure 3. It has a total of $2n$ gates ($n$ inverters and $n$ OR gates). The output of all inverters is connected to the input of all ORs.



For all $i, j$ Inp($\mathrm{N}_i$)=0, Out($\mathrm{O}_j$)=0, $\mathrm{Inp}_i(\mathrm{O}_j)$=Out($\mathrm{N}_i$).

Figure 3: A Bad Case

Given that all inputs and all outputs are zeros, the only minimal diagnoses are that either all inverters are faulty, or that something is wrong with all ORs. At any rate, each of the diagnoses involves at least $n$ components, forcing the generation of at least all sets of $n$-1 components before an actual failure is hit. The number of such sets is exponential.

# 6 Discussion

## 6.1 The Uniform Failing Rate Assumption

I have so far assumed that different components fail at the same rate. While this assumption is helpful in the derivation of upper bounds for the algorithm's complexity, it does not reduce the complexity of the problem. A slight change to algorithm 5.6 results in an algorithm that is even more efficient in the case of distinct failure rates.

**Definition 6.1** *A priori* Failing Rate

*Let* $\mathrm{C}_i \in \mathrm{COMPS}$ *be a component. By* $\mathrm{AP}(\mathrm{C}_i)$ *we denote the a priori probability that* $\mathrm{C}_i$ *fails, given there is a fault. Now, let* X *be a set of components. Assuming components fail independently, the* a priori *failing rate of* X *is given by:*

$$\mathrm{AP}(\mathrm{X}) \overset{\mathrm{def}}{=} \prod_{C_i \in X} \mathrm{AP}(\mathrm{C}_i) \cdot \prod_{C_i \in COMPS-X} (1 - \mathrm{AP}(\mathrm{C}_i))$$

*From now on, assume components are sorted so that whenever* $i > j$*, then* $\mathrm{AP}(\mathrm{C}_i) < \mathrm{AP}(\mathrm{C}_j)$.

An *a priori* failure rate is often available for real world diagnostic problems (e.g. from a circuit's history). Algorithm 5.6 may take advantage of this by giving high priority to sets with higher *a priori* probability to fail.

A *Next-Set* routine that does exactly that is described next. It uses a heap (initiated by *Init-Sets-Generator*) in which sets are ordered by their probability of failure. Sets with higher probability are generated and expanded first.

**Algorithm 6.1** Candidate Diagnoses Generation for Non-Uniform Failing Rate.

```
Procedure Init-Sets-Generator ()
H ← Make-Heap ({}).

Procedure Next-Set ()
if H ≠ empty-heap do
        1. S ← Delete-Max (H).
        2. Let k ← $\max_{\{C_i \in S\}} i$ ,     (0 if S={})
        3. For all $C_i$ such that i > k do Insert-Heap(H, S ∪{$C_i$})
        4. return S
```

**Proposition 6.2** *If all components have an a priori failing rate that is less than or equal to $\frac{1}{2}$, then for every set of components* S *and for every single component* C ∈ COMPS-S, AP(S) > AP(S∪{C}).

**Corollary 6.3** *If no single component has an* a priori *probability that is larger than $\frac{1}{2}$, then the* Next-Set *routine of algorithm 6.1 produces sets of monotonically increasing* a priori *probability for failure.*

The following proposition shows that in the case of non-uniform failing rate, using the altered *Next-Set* procedure, chances are of hitting FAILURE even faster than in the case of uniform failing rate.

**Proposition 6.4** *Let $F$ be the collection of sets of components, sorted by increasing cardinality. Let $F'$ be the same collection, sorted by a priori failure probability. Then, for all $1 \leq i \leq 2^n$, the a priori probability that* FAILURE *is one of the first $i$ sets of $F'$ is equal or higher than that it is one of the first $i$ sets of $F$.*

**Corollary 6.5** *In the case of non-uniform failure rate, the modified algorithm does even better than the original algorithm.*

## 6.2 The Independently Failing Components Assumption

In real life components do *not* fail independently of each other. In electronic circuits, for instance, it often happens that a failing component causes the failure of adjacent components. One way to deal with such dependencies is to explicitly represent causality. Here, we would only discuss a simple extension to the basic generate-and-test algorithm that uses *empirical* data to take into account frequently failing *combinations* of components.

Suppose that we have a database of past failures. In one pass over this database, it is possible to construct a heap in which sets are ordered based on their database frequency. Theoretically, such a heap may require $O(2^n)$ nodes. In practice, however, the number of such nodes may not exceed the number of past failures recorded in the database and most often would be much smaller.

The set generation step in the algorithm would then simply hand the set currently at the root of the heap and proceed by deleting it from the heap. When the heap is empty, the sets generator will output all remaining sets (not all sets are represented in the past failures database) in some pre-agreed order (e.g. by their cardinality or by the total/product of the a priori probability of their components).

# 7 Summary

In this paper I have shown:

1. that the Minimal Diagnoses problem is exponential, and not only NP-Hard; and
2. that the plain generate-and-test approach does have some desirable properties when examined through the glasses of *expected* performance. Without advocating its use, I show that it is an equal competitor with current, more complex, algorithms for diagnosis.

The class of problems that is well-covered (i.e. solved quickly) by a given algorithm seems to be an interesting area of research. Such a determination may allow the choice of a good algorithm for the particular problem at hand. It also seems, though I have not worked this out completely, that the conflict sets algorithm does particularly well on examples in which the generate-and-test algorithm fails miserably, and vice versa. If this is the case, it may well pay to run the two in parallel so as to significantly increase the likelihood of quick diagnosis at the modest expense of doubling the run time.

**Current work:**
Being exponentially hard, it seems worthwhile to re-examine the very definition of the diagnostic problem. In current work, I redefine the problem in terms of its goals [Rymon 91].

## References

[Allemang et al 89] Allemang, D., Tanner, M. C., Bylander, T. and Josephson, J., Computational Complexity of Hypothesis Assembly. *Proc. IJCAI-89*, Detroit, MI, 1989, pp.1112-1117.

[Augluin and Valiant 79] Augluin, D. and Valiant, L. G., Fast Probabilistic Algorithms for Hamiltonian Paths and Matchings. *Journal of Computer and System Science, 18, 1979, pp. 155-193.*

[Bylander et al 89] Bylander, T., Allemang, D., Tanner, M. C., and Josephson, J. R., Some Results Concerning the Computational Complexity of Abduction. *Proc. KR-89*, pp.44-54.

[de Kleer and Williams 87] de Kleer, J. and Williams, B. C., Diagnosing Multiple Faults, *Artificial Intelligence, 32, 1987, pp. 97-130. Proceedings of AAAI-90, Boston, MA, July 1990, pp. 324-330.*

[de Kleer et al 90] de Kleer, J., Mackworth, A. K. and Reiter, R., Characterizing Diagnoses. *Proceedings of AAAI-90, Boston, MA, July 1990, pp. 324-330.*

[Garey and Johnson 79] Garey, M. R. and Johnson, D. S., Computers and Intractability. A Guide to the Theory of NP-Completeness. *W.H. Freeman and Company, New York, 1979, pp. 46, 54-56.*

[Poole and Provan 90] Poole, D. and Provan G. M., What is an Optimal Diagnosis? *Conference on Uncertainty in Artificial Intelligence, pp. 46-53, 1990.*

[Reggia et al 85] Reggia, J. A., Nau, D. S. and Wang, P. Y., A Formal Model of Diagnostic Inference. I. Problem Formulation and Decomposition. *Information Sciences 37, 1985, pp. 227-256.*

[Reiter 87] Reiter, R., A Theory of Diagnosis From First Principles. *Artificial Intelligence, 32, 1987, pp. 57-95.*

[Rymon 91] Rymon, R. Towards Goal-directed Diagnosis (Preliminary Report). Submitted to the *Second International Workshop on Principles of Diagnosis.* Milan, Italy, October 1991.