

UC Irvine

ICS Technical Reports

Title

Empirical evaluation of diagnosis as optimization in constraint networks

Permalink

<https://escholarship.org/uc/item/7qv2q50h>

Authors

Fattah, Yousri El
Dechter, Rina

Publication Date

1992-11-20

Peer reviewed

Notice: This Material
may be protected
by Copyright Law
(Title 17 U.S.C.)

ARCHIVES
Z
699
C3
NO. 92-109
C.2

Empirical Evaluation of Diagnosis as
Optimization in Constraint Networks

Yousri El Fattah
fattah@ics.uci.edu
Rina Dechter
dechter@ics.uci.edu

Technical Report 92-109

November 20, 1992

Appeared in *Working Papers of The Third International Workshop on Principles of Diagnosis* (DX-92), October 12-14, 1992, Rosario, Washington.

This work was supported in part by NSF grant IRI-9157636, the Air Force Office of Scientific Research grant AFOSR 900136, Toshiba of America, GE Corporate R&D, and Xerox Palo Alto Research Center.

Empirical Evaluation of Diagnosis as Optimization in Constraint Networks¹

Yousri El Fattah and Rina Dechter

Department of Information and Computer Science
University of California, Irvine, CA 92717-3425

fattah@ics.uci.edu, dechter@ics.uci.edu

Abstract

In this paper we study empirically the behavior of algorithm *structure-based abduction* (SAB) which was developed in the framework of constraint networks [5], by comparing it with a model-based diagnosis algorithm MBD, similar to algorithm GDE [3]. The distinguishing features of algorithm SAB are that it exploits the structure of the constraint network and that it is most efficient when the problem contains no cycles. The performance of both algorithms is tested on a family of parametrized acyclic combinational circuits for the task of finding all minimal-cardinality diagnoses. The results show that due to its exponential complexity for large circuits MBD can run out of space and time, while SAB is able to compute the diagnoses for those same circuits. Unlike MBD, SAB appears to be insensitive to variations in the circuit types and input probabilities. For small circuits, the average time for MBD relative to SAB seems to be proportional to the number of conflicts relative to the number of diagnoses.

1 Introduction

Generally speaking, *diagnosis* is a form of abduction or inference to the best explanation. *Explanations* are those minimal sets of value instantiations that are consistent with the model and the observations, and *best* explanations are those that optimize some desirability measure. Some common measures are *minimal cardinality* [10], *parsimonious* covering theory [15], *most probable* explanation [14], and *minimal cost proofs* [1].

Cost-based abduction is a variant of weighted abduction that was devised for interpretation of natural language [11]. A well-defined relation between cost-based abduction and belief revision in probabilistic networks has been shown [1], and algorithms based on linear constraint networks have been proposed [18]. Diagnosis can be viewed as a variant of cost-based abduction formulated as an optimization task in the framework of constraint networks [5], [4], [9]. The advantage of that problem formulation is that all algorithms and heuristics developed within that framework can be exploited. In particular, when a constraint network is acyclic, an algorithm exists for finding an optimal solution in linear time [6]. Tailoring this algorithm to diagnosis results in an algorithm called *structure-based abduction* (SAB) [5], which will be investigated empirically here. This paper presents preliminary results comparing the performance of SAB with a model-based diagnosis GDE-like algorithm [3], called MBD, on the the diagnostic task of finding all minimal-cardinality diagnoses.

¹This work was supported in part by the NSF (grant IRI-9157636), the Air Force Office of Scientific Research (grant AFOSR 900136), Toshiba of America, GE corporate R&D, and Xerox Palo Alto Research Center.

Note that since SAB is applicable only to tree-like circuits while MBD is general, the comparison may seem somewhat inappropriate. However, SAB can be extended to a general sound and complete algorithm via tree-clustering algorithm with complexity bound which is exponential in the size of the largest cluster. Moreover, MBD, is normally implemented, as a sound but incomplete algorithm since it uses constraint propagation as its basic inference mechanism. For the type of circuits considered here, both algorithms are sound and complete.

The structure of the paper is as follows. Section 2 reviews basic definitions of model-based diagnosis and presents algorithm MBD. Section 3 formulates diagnosis as optimization of constraint networks and describes algorithm SAB. Section 4 presents empirical results, and section 5 provides concluding remarks.

2 Model-Based Diagnosis

Following [2] we define *model-based diagnosis* in terms of a triple $(SD, COMPS, OBS)$ where

1. SD , the system description, is a set of first-order sentences;
2. $COMPS$, the system components, is a finite set of constants;
3. OBS is a finite set of first-order sentences.

Each constant $c \in COMPS$ is associated with abnormal literals $ab(c)$ or $\neg ab(c)$, where $ab(c)$ means “ c is abnormal” while $\neg ab(c)$ means “ c is ok”. A *conflict set* for $(SD, COMPS, OBS)$ is a set $CONF \subseteq COMPS$ such that

$$SD \cup OBS \cup \left[\bigwedge_{c \in CONF} \neg ab(c) \right]$$

is inconsistent. A conflict set for $(SD, COMPS, OBS)$ is *minimal* iff no proper subset of it is a conflict set for $(SD, COMPS, OBS)$. A conflict set $CONF$ corresponds to a clause,

$$\bigvee_{c \in CONF} ab(c)$$

called a *conflict*, which is entailed by $SD \cup OBS$.

Intuitively, a diagnosis is a truth assignment for each abnormal literal (ab or $\neg ab$) which is consistent with the model and the observations. Formally, let $\Delta \subseteq COMPS$, and let $\mathcal{D}(\Delta, COMPS)$ stand for the conjunction

$$\left[\bigwedge_{c \in \Delta} ab(c) \right] \wedge \left[\bigwedge_{c \in COMPS - \Delta} \neg ab(c) \right]$$

A formula $\mathcal{D}(\Delta, COMPS)$ is a *diagnosis* for $(SD, COMPS, OBS)$ iff

$$SD \cup OBS \cup \{ \mathcal{D}(\Delta, COMPS) \}$$

is satisfiable.

We say that an assignment $V = v$ is a *prediction* with respect to a set of assumptions $\mathcal{D}(\Delta, COMPS)$ if it is a prime implicate of $SD \cup OBS_V \cup \mathcal{D}(\Delta, COMPS)$ where OBS_V is all observations excluding the one involving V .

Figure 1: Algorithm MBD

Input $(SD, COMPS, OBS)$; SD is represented as a set of value inference rules; OBS is a value assignment for a subset of variables in SD .

Output All minimal-cardinality diagnoses.

Initialization For each variable assigned a value in OBS assert a prediction with that value in all environments.

Description

1. For each $(variable, value)$ pair find all minimal environments that entail it.
 2. For each $(variable, value)$ that conflicts with the observation declare all its minimal environments as conflict sets. Call the collection of conflict sets F .
 3. Output all minimal-cardinality covers of F .
-

Algorithm MBD, presented in Fig. 1, is a variant of algorithm GDE [3] given in very high level pseudo code. It works as a three-step process: In step 1, predictions are computed by a form of constraint propagation (value inference), which although generally incomplete, is complete for restricted languages such as trees and Horn theories. In step 2, all support labels associated with predictions that conflict with the observations are identified as conflict sets (nogoods). In step 3, all minimal-cardinality diagnoses are computed as the minimal-cardinality covers of all the conflict sets.

The worst-case time and space complexity for finding all minimal supports and all minimal conflict sets is clearly exponential in $|COMPS|$. Since the task of finding a minimal cover to a set of subsets is known to be NP -complete (even when each subset has at most two elements) [8], the resulting worst-case complexity of the algorithm is likely to be (if $P \neq NP$) doubly exponential.

3 Diagnosis as Optimization

In this section we show that a diagnosis can be formulated as an optimization task over a constraint network. Given a set of variables X_1, \dots, X_n each having a finite set of domain values $dom(X_1), \dots, dom(X_n)$, a *constraint network* CN is a set of relations $\{r_1, \dots, r_m\}$, called *constraints*, each defined on a subset of variables S_1, \dots, S_m . A *relation* over a subset of variables X_{i_1}, \dots, X_{i_j} is a subset of the cartesian products of their domains, and a *solution* to CN is an assignment of a value to each variable satisfying all the constraints. A constraint network can be associated with the set of all its solutions. Formally,

$$sol(CN) = \{t = (X_1 = x_1, \dots, X_n = x_n) \mid \forall j, t_{S_j} \in r_j\},$$

when t_S stands for the projection of a tuple t on a subset of variables S . Frequently, solutions of constraint networks can be assigned different costs as dictated by the application. A simple cost function would associate a cost $c(X = x)$ with each value of a variable and define the *cost of a*

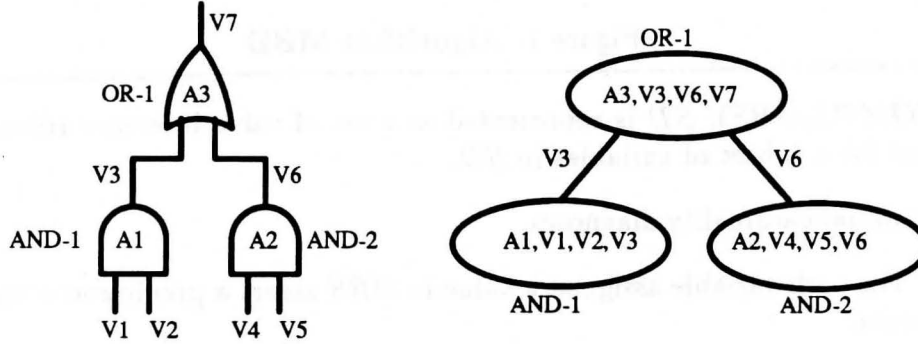


Figure 2: A circuit and its dual graph

solution to be the sum costs of all its values. Given a constraint network and a cost function, the optimization task is to find a solution having an optimal cost (largest or smallest).

To capture the diagnosis task as constraint optimization, we map the triple $(SD, COMPS, OBS)$ into a relational framework. The system description SD will be described in terms of two sets of variables: the *system variables* X_1, \dots, X_n , which are the inputs and outputs of all components, with their associated finite domain values $dom(X_1), \dots, dom(X_n)$, and the *assumption variables* $A = \{A_1, \dots, A_m\}$. The observations OBS translates to forcing value assignments for the corresponding variables. Each assumption variable A_j is associated with component c_j and describes the component's functioning status. In the simplest case, these are bi-valued variables indicating whether the component is normal (value 0) or abnormal (value 1). In the more involved case, they can index different fault models. Each component $c_j \in COMPS$ is associated with a constraint r_j describing its input-output behaviors under all its normal and abnormal conditions. Thus, the constraint r_j is defined over the set $R_j = \{A_j\} \cup S_j$, where S_j is the set of input and output variables for component c_j . Non-zero costs are associated with assumption variables only. An abnormal component is assigned a cost of 1 [i.e., $c(A = 1) = 1$], while a normal component has a zero cost [i.e., $c(A = 0) = 0$].

Given a model description and a set of observations, the diagnosis task is to construct an explanation, namely, a solution $(X_1 = x_1, \dots, X_n = x_n, A_1 = a_1, \dots, A_m = a_m)$ that is consistent with the observations. The cost of an explanation is the sum of the costs associated with the assumption variables. That is,

$$C(\{X_1 = x_1, \dots, X_n = x_n, A_1 = a_1, \dots, A_m = a_m\}) = \sum_{A_j \in A} c(A_j = a_j). \quad (1)$$

Clearly, the task of finding a minimal-cardinality diagnosis corresponds to finding a minimal solution of a constraint network as defined above.

The topology of a constraint network can be depicted by using the notion of its *dual graph*. A dual graph (called an "intersection graph" in database theory [12]) represents each constraint by a node (called a *c-variable*) and associates a labeled arc with any two nodes that share variables. The arcs are labeled by the shared variables. If the dual constraint graph is a tree (called a join tree) or can be transformed into a tree by removing redundant arcs (in linear time), then the constraint network is said to be acyclic [12]. In that case, a consistent solution or an optimal solution can be assembled in linear time. It is also known that a cyclic constraint network can be transformed into an acyclic constraint network by forming larger clusters of c-variables, but because the efficiency of

Table 1: Constraint AND-1

| A1 | V1 | V2 | V3 | Cost |
|----|----|----|----|------|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | X | X | X | 1 |

this procedure is exponential in the size of the largest cluster generated, it is appropriate only for problems having a small induced width (for details see [7]).

Example 3.1 Consider the circuit in Fig. 2, which has two AND-gates AND-1, AND-2 connected to an OR-gate OR-1. We model this circuit using three bi-valued assumption variables, A_1, A_2, A_3 , whose values indicate whether AND-1, AND-2, and OR-1, respectively, are working normally or abnormally. Constraint AND-1 is given in Table 1, where the values marked with an X correspond to “don’t care” conditions (either 0 or 1). Given the set of observations $\{V1 = 0, V2 = 1, V4 = 1, V5 = 1, V7 = 0\}$, there are two best explanations, both having cost 1.

$$\{V1 = 0, V2 = 1, V3 = 0, V4 = 1, V5 = 1, V6 = 0, V7 = 0, A1 = 0, A2 = 1, A3 = 0\}$$

corresponding to a single-fault AND-2, and

$$\{V1 = 0, V2 = 1, V3 = 0, V4 = 1, V5 = 1, V6 = 1, V7 = 0, A1 = 0, A2 = 0, A3 = 1\}$$

corresponding to a single-fault OR-1. Notice that multiple-fault explanations also exist, but they have higher costs.

Fig. 3 gives the SAB algorithm for an acyclic constraint network. In the bottom-up step, pointers are created from each tuple t of a parent node to each of a set of minimizing tuples for each child node (see Fig. 4). The complexity of the bottom-up phase of the algorithm is $O(n \cdot t \cdot \log t)$, where n is the number of c-variables and t is the number of tuples. Details about the algorithm and arguments for correctness are given in [5], [6].

4 Empirical Evaluation

4.1 Method

We consider the parametrized circuit $b(p, k)$, shown in Fig. 5, having $n = 2^k - 1$ components and k layers numbered $1, 2, \dots, k$ from the leaves to the root. The component types are independent random variables and are either AND or OR gates. A component in an odd (even) layer is an AND (OR) with probability p , OR (AND) with probability $1 - p$.² The circuit $b(p, k)$ is acyclic but can have a worst-case exponential number of minimal conflicts.³

²A special case of the circuit, $b(1, k)$, was developed by [17].

³Although the number of conflicts can be exponential for certain inputs, this does not imply that it is the case for most inputs.

Figure 3: Algorithm **SAB**

Input An acyclic constraint network T having variables $X = \{X_1, \dots, X_n\}$, assumption variables $A = \{A_1, \dots, A_m\}$, a cost function c , and relations $\{r_1, \dots, r_m\}$ defined on the sets R_1, \dots, R_m , where $R_j = \{A_j\} \cup S_j$, and $S_j \subset X$ ($j=1, \dots, m$).

Output One (or all) minimum cost solution(s).

Description

1. (**Bottom-up**) Compute weights:
 If R_j is a leaf in T , then for each tuple $t \in r_j$ we define the weight mapping $w(t) = c(t_{A_j})$.
 If R_j is a parent in T , then for each tuple $t \in r_j$ we define the weight mapping

$$w(t) = c(t_{A_j}) + \sum_{i:child(j)} \min_{s \in r_i | s_{R_i \cap R_j} = t_{R_i \cap R_j}} w(s)$$

2. (**Top-Bottom**) Generate minimum-cost solution(s) by following the pointers of a minimum-weight tuple from root to leaves.

For every circuit, we generate inputs as independent random variables with each input being 1 with probability q and 0 with probability $1 - q$. Then the correct output is determined and reversed. The observation for every problem consists of the inputs and the faulty output. We feed the same batch of problems to both SAB and MBD and record the performance measures of each algorithm. For MBD limits are placed on space and time to protect against exponential demand on those resources. If the limits are reached MBD, is forced to halt before a solution is returned.

For MBD, we recorded the number of predictions and the number of conflict sets, as well as the CPU time. We also measured performance by counting basic operations, which are defined as either a value inference or a test for a set covering. For SAB, we computed the CPU time and the number of consistency checks. In addition, the number of minimal-cardinality diagnoses for each problem instance was recorded. Each experiment consists of $M(= 2)$ circuit instances with, $N(= 5)$ observations for each circuit. The performance measures are averaged over the $N \cdot M$ problem instances. Circuits considered have components ranging from 3 to 511.

4.2 Results

The results of the experiments are given in Figs. 6 and 7, which show the average CPU times used by SAB and MBD for computing all minimal-cardinality diagnoses. In Fig. 6, the input probability is varied while the type probability is fixed at 0.2. In Fig. 7 the type probability is varied while the input probability is fixed at 0.5. The figures show that MBD soon becomes inefficient for computing the diagnoses for moderately sized circuits. Points not shown for MBD correspond to circuit sizes for which MBD failed on *all* problem instances, because its space (or time) requirement exceeded available resources. For such problems, the space required by MBD's caching during the prediction

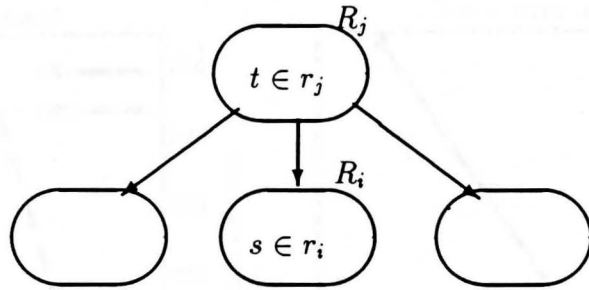


Figure 4: Each tuple t points to the minimizing tuples of the child nodes

phase had exhausted the available resource limit, set at 10,000 predictions. Other points shown for MBD give the average CPU seconds over the solved-problem instances.

To get around the space problem of the ATMS caching, a variant of MBD was implemented that determined the conflict sets one at a time while computing the diagnoses via set covering. This variant is essentially Reiter's algorithm, DIAGNOSE [16]. Although DIAGNOSE eliminated the space problem, its time performance turned out to be much worse than MBD and it became inefficient in time (exceeding limits set at 3000 seconds) on the same problems for which MBD was inefficient in space. As can be observed from Figs. 6, and 7, MBD's performance is sensitive to the type of the randomly generated circuits and to the probability of the inputs. For circuits of size 63 components and type probability 0.2, changing the input probability from 0 to 0.8 resulted in a change in the average CPU seconds from 71.5 to 19.5 for MBD and 6.95 to 8.04 for SAB. The coefficient of variation (the ratio of the standard deviation to the mean) of the CPU seconds is also an order of magnitude higher for MBD than for SAB (around 50% for MBD and 5% for SAB.) Figs. 6, and 7 also show that for small sized circuits the MBD's average performance is comparable to SAB's. Fig. 6 shows that for input probability close to 0 or 1, MBD is generally slower than SAB. On the other hand, as Fig. 7 shows, for input probability 0.5, MBD is very close to SAB. Based on these empirical results, it appears that MBD's performance is closely related to the number of conflicts, while SAB's is closely related to the number of diagnoses. On average, the time performance of MBD relative to SAB degrades as the number of conflicts relative to the

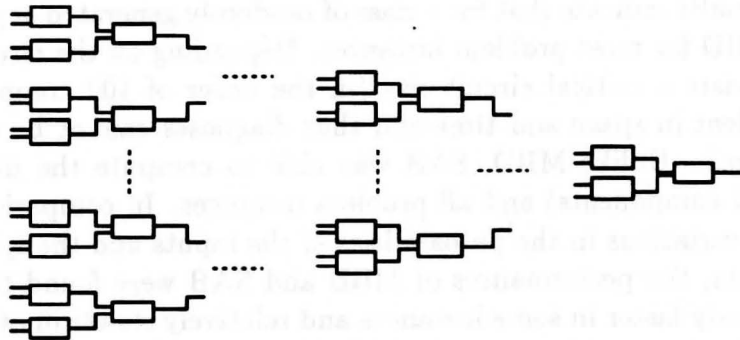


Figure 5: A parametrized circuit $b(p, k)$, which is a binary tree with k layers numbered $1, 2, \dots, k$ from the leaves to the root. Components are either AND-gates or OR-gates, with p being the probability of an AND for odd layers, OR for even layers.

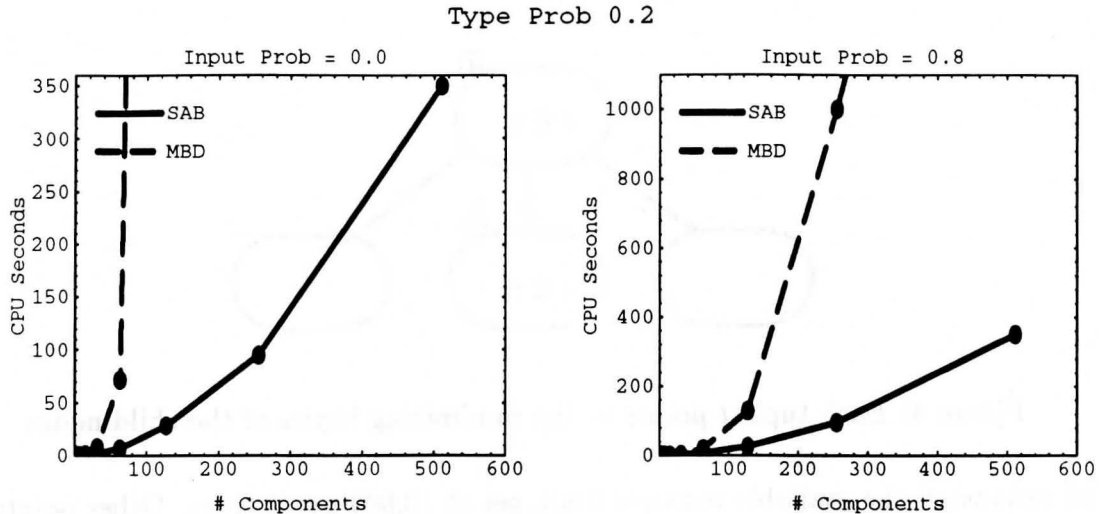


Figure 6: Average CPU seconds for different input probabilities

number of diagnoses increases. Fig. 8 shows the average number of conflicts and diagnoses for two cases: one where SAB is faster (type probability 0.2 and input probability 0), and the other where MBD is faster (type probability 0 and input probability 0.5). The figure clearly shows that MBD is at a disadvantage with respect to SAB when there is a large number of conflict sets but a small number of minimal-cardinality covers. On the other hand, MBD is at an advantage with respect to SAB when the number of conflicts is small while the number of diagnoses is relatively large.

5 Summary and Conclusions

The primary motivation for this work was to demonstrate that for some classes of problems the tasks of diagnosis and abduction can be accomplished efficiently by exploiting the structure of the problem. To that end, we have implemented algorithm SAB, which finds the minimal-cardinality diagnoses and is applicable to acyclic-problem instances. Since the performance of our algorithm is time and space linear (while the performance of the general GDE-like algorithm, MBD, is doubly exponential), we expected to show a significant performance gain.

Our preliminary results indicate that for a class of randomly generated acyclic circuits, algorithm SAB is superior to MBD for most problem instances. Depending on the circuit type and the input probabilities, there exists a critical circuit size (in the order of 100 components) beyond which MBD becomes inefficient in space and time and thus diagnoses cannot be computed (for at least one generated problem). Unlike MBD, SAB was able to compute the diagnoses for all circuit sizes studied (3 to 511 components) and all problem instances. In comparison to SAB, MBD was markedly sensitive to variations in the probabilities of the inputs and the types of circuits studied. For smaller circuit sizes, the performances of MBD and SAB were found to be very comparable, with each being relatively faster in some instances and relatively slower in others. In general, MBD seems to be at a disadvantage with respect to SAB when the number of conflict sets is large while the number of minimal-cardinality covers is small. On the other hand, MBD is at an advantage with respect to SAB when the number of conflicts is small while the number of diagnoses is relatively large.

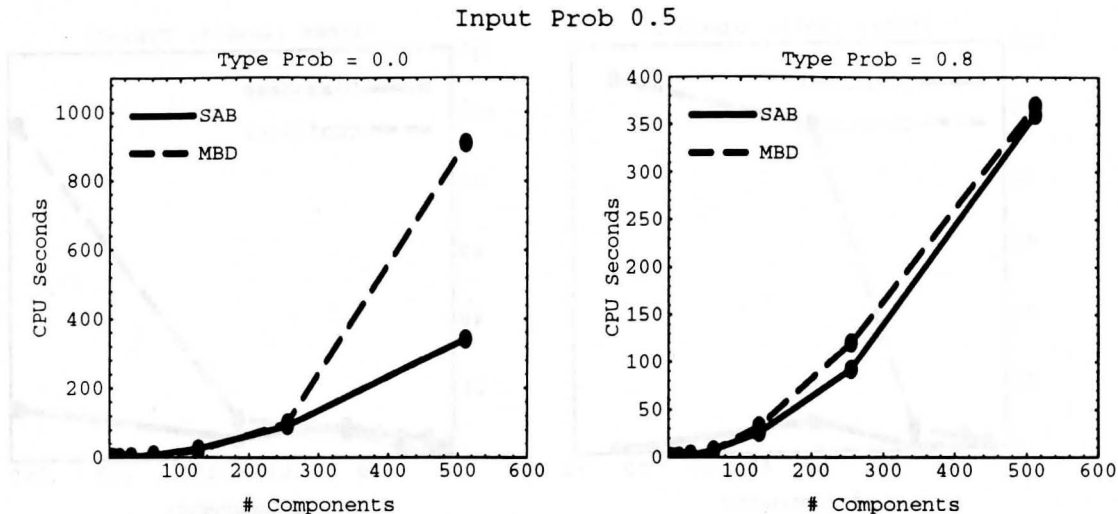


Figure 7: Average CPU seconds for different type probabilities

We regard the present results as preliminary and plan to extend this investigation in various directions. First, we will investigate other acyclic circuits with minimal-cardinality diagnoses of cardinality higher than 1. Second, we will experiment with additional MBD variants, addressing fault models and focusing tactics. In particular, we will investigate the role of the weights of assumption variables in SAB. Finally, and perhaps most importantly, we will apply the algorithm to cyclic problems using tree clustering as a preprocessing phase. Applying tree clustering to cyclic circuits is known to be exponential in the clique sizes [7]. However, even when relatively expensive, it might be worthwhile to do tree clustering once so that all future diagnosis tasks will be relatively easy. This will be tested empirically on series-parallel circuits [13] whose width is known to be 2 and for which clustering is only cubic in time and space, as well as on problems having larger clusters.

References

- [1] E. Charniak and S.E. Shimony. Probabilistic semantics for cost-based abduction. In *Proceedings, AAAI-90*, pages 106–111, 1990.
- [2] J. de Kleer, A.K. Mackworth, and R. Reiter. Characterizing diagnoses and systems. *Artificial Intelligence*, 56:197–222, 1992.
- [3] J. de Kleer and B. C. Williams. Diagnosing multiple faults. *Artificial Intelligence*, 32:97–130, 1987.
- [4] R. Dechter. Constraint networks. In *Encyclopedia of Artificial Intelligence*, pages 276–285. J. Wiley, New York, second edition, 1992.
- [5] R. Dechter and A. Dechter. Belief maintenance in dynamic constraint networks. In *Proceedings, AAAI-88*, pages 37–42, 1988.
- [6] R. Dechter, A. Dechter, and J. Pearl. Optimization in constraint networks. In R.M. Olivier and J.Q. Smith, editors, *Influence Diagrams, Belief Nets and Decision Analysis*, pages 411–425. J. Wiley, New York, 1990.

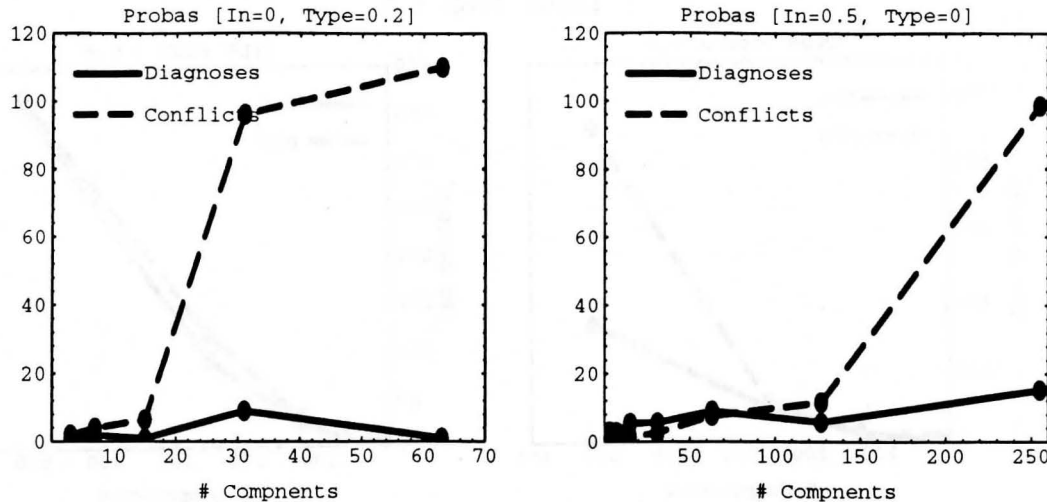


Figure 8: Average number of conflicts and diagnoses versus number of components

- [7] R. Dechter and J. Pearl. Tree clustering for constraint networks. *Artificial Intelligence*, 38:353–366, 1989.
- [8] M.R. Garey and D.S. Johnson. *Computers and Tractability*. W.H. Freeman, New York, 1979.
- [9] H. Geffner and J. Pearl. An improved constraint propagation algorithm for diagnosis. In *Proceedings, IJCAI-87*, pages 1105–1111, 1987.
- [10] M. R. Genesereth. The use of design descriptions in automated diagnosis. *Artificial Intelligence*, 24:411–436, 1984.
- [11] J.R. Hobbs, M. Stickel, P. Martin, and D. Edwards. Interpretation as abduction. In *Proceedings, 26th Annual Meeting of the Association for Computing Linguistics*, pages 95–103, 1988.
- [12] D. Maier. *The Theory of Relational Databases*. Computer Science Press, Rockville, MD, 1983.
- [13] U. Montanari. Networks of constraints: Fundamental properties and applications to picture processing. *Inform. Sci.*, 7:95–132, 1974.
- [14] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufman, San Mateo, CA, 1988.
- [15] Y. Peng and J.A. Reggia. *Abductive Inference Models for Diagnostic Problem Solving*. Springer-Verlag, Berlin, 1990.
- [16] R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32:57–95, 1987.
- [17] P. Resnick. Generalizing on multiple grounds: Performance learning in model-based troubleshooting. Technical Report AI-TR 1052, MIT Artificial Intelligence Laboratory, February 1989.
- [18] E. Santos, Jr. On the generation of alternative explanations with implications for belief revision. In *Proceedings, Uncertainty in Artificial Intelligence*, pages 339–347, 1991.