Finding All Diagnoses is Redundant

Roni Stern[1] and **Meir Kalech**[2] and **Alexander Feldman**[3] and **Shelly Rogov**[2] and **Tom Zamir**[2]
[1]Harvard University, Cambridge, MA USA, e-mail: roni.stern@gmail.com
[2]Ben-Gurion University of the Negev, Beer-Sheva, Israel
e-mail: kalech@bgu.ac.il, shellyr4u@gmail.com, tom.zamir.i@gmail.com
[3] University College Cork, Cork, Ireland, e-mail: alex@llama.gs

Abstract

Consistency-based diagnosis algorithms seek for explanations to unexpected observations in systems. In particular, given a formal model of a system and an observation, a diagnosis is a set of assumables over the health of the components that is consistent with the model and the observation. Consistency-based algorithms are widely researched in the model-based diagnosis literature. A known limitation of consistency-based algorithms is that the number of diagnoses can be very large. Presenting a very large number of diagnoses to a human operator is not likely to be helpful. This raises two questions: (1) how to present the information represented by the consistency-based diagnoses to a human operator? and (2) are all diagnoses required to present this information?. We argue that a human operator requires an aggregated view of the list of partial set of diagnoses, and propose a natural way to do so called the Component Fault Probability (CFP). A CFP shows for every component the probability that it is faulty. Then, we describe a way to evaluate the quality of a CFP, and show that returning all subset minimal diagnoses is seldom required to generate an accurate CFP.

1 Introduction

A diagnosis problem arises when a system does not behave as expected. The goal in diagnosis algorithm is to find the set of system components that caused the unexpected behavior of the system. There are many real-life instances of the diagnosis problem and solving the diagnosis problem has been researched in the Artificial Intelligence community for several decades. One of the fundamental approaches to diagnosis is Model-Based Diagnosis (MBD). In MBD, a formal model of the diagnosed system is assumed to exist that specifies the expected behavior of the system. This model, along with the observed behavior is then used to deduce candidate *diagnoses*. Consistency-based MBD algorithms, a predominant class of MBD algorithms, aim to return all candidate diagnoses that are logically consistent with the system model and the observed system behavior [Reiter, 1987b; de Kleer and Williams, 1987a]. Such candidates are called *diagnoses*.

If there is only a single diagnosis, then this diagnosis is returned by the diagnosis algorithm and the user can then replace or fix the faulty components. Unfortunately, in systems that are not very small it is often the case that the number of diagnoses is very large, potentially exponential in the number of system components. Prior work have proposed algorithms for suggesting additional tests to narrow the list of diagnoses [Shakeri *et al.*, 2000; Feldman *et al.*, 2010b], or for positioning probes to view the output of internal system components [de Kleer and Williams, 1987a]. We consider a different presentation, where the output of the diagnosis algorithm should be displayed to a human operator that will decide on subsequent actions.

Clearly, a human operator cannot intelligently consider a list of hundreds of possible diagnoses. Given the probability of each diagnosis to be correct, one may consider displaying only the most probable diagnoses. In the absence of probabilities, one common approach is to display only diagnoses with the smallest number of components, known as the minimal cardinality diagnoses. Another common approach is to return only a diagnosis that is not a superset of another diagnosis. A set of such diagnoses is known as minimal subset diagnoses. Unfortunately, for systems with hundreds of components or more, the number of even only the minimal cardinality diagnoses becomes so large that even enumerating them is time consuming [Siddiqi, 2011]. Beyond the computational problems, there is an additional user interface challenge: how is a human operator expected to reason about a long list of possible diagnoses?. Furthermore, it is unknown how much information is lost by considering only the highest probable/minimal diagnoses. Figure 2, which is explained in greater detail later in this paper, shows an example where returning the most probable diagnosis is very misleading.

Based on recent research [Feldman *et al.*, 2013; Stern *et al.*, 2012], we consider a natural aggregation of the set of diagnoses by mapping it to a probability of fault for every component. We call the assignment of fault probability to each component the Component Fault Probability (CFP). CFPs can be evaluated by measuring the vector distance between a CFP and the "optimal" CFP produced by the real faults. Among the contributions of this paper is the definition of a CFP, describing how to generate a CFP from a set of diagnoses and how to evaluate a given CFP.

The question we then raise is which and how many diagnoses a diagnosis algorithm needs to find to produce a high quality CFP. Interestingly, we show empirically that finding all minimal subset diagnoses is not needed to produce a good CFP. In fact, improvement caused by adding more

diagnoses to the CFP quality becomes negligible after finding a relatively small number of minimal subset diagnoses, as the CFP quality converges to a specific quality. We also show that the amount of diagnoses required to converge to that CFP quality greatly depends on the order by which the minimal subset diagnoses are found. Convergence of the CFP quality was faster and more stable when the diagnoses were found in order of increasing cardinality than in the reverse order. This result poses a broader question to the diagnosis community: how to search for diagnoses such as to find diverse set of diagnoses that will result in rapid convergence of the CFP quality.

2 Consistency-Based Model Based Diagnosis

Model Based Diagnosis (MBD) problems arise when the normal behavior of a system is violated due to faulty components as indicated by certain observations. We focus on *weak fault models* (WFM), which ignore the mode of abnormal behavior of components [de Kleer *et al.*, 1992].

An MBD problem is specified as a triplet $\langle SD, COMPS, OBS\rangle$ where: SD is a system description, $COMPS$ is a set of components, and OBS is an observation.

The system description takes into account that some components might be abnormal (faulty). This is specified by an unary predicate $h(\cdot)$ on components such that $h(c)$ is true when component c is healthy, while $\neg h(c)$ is true when c is faulty. Denoting the correct behavior of c as a propositional formula, φ_c, SD is given formally as

$$SD = \bigwedge_{c \in COMPS} h(c) \Rightarrow \varphi_c$$

Namely, each component which is healthy follows its correct behavior. A diagnosis problem (DP) arises when, under the assumption that none of the components are faulty, there is an inconsistency between the system description and the observations [de Kleer and Williams, 1987b; Reiter, 1987a].

Definition 1. [Diagnosis Problem]. *Given an MBD problem, $\langle SD, COMPS, OBS\rangle$, a diagnosis problem arises when*

$$SD \wedge \bigwedge_{c \in COMPS} h(c) \ \wedge OBS \vdash \bot$$

For example, a diagnosis problem arises for the MBD of Figure 1 as normal behavior would give output $E = 1$. Once there is an inconsistency, a diagnosis algorithm tries to find a subset $\Delta \subseteq COMPS$ which, if assumed faulty, explains the observation.

Definition 2. [Diagnosis] *Given an MBD problem, $\langle SD, COMPS, OBS\rangle$, the set of components $\Delta \subseteq COMPS$ is a* diagnosis *if*

$$SD \wedge \bigwedge_{c \in \Delta} \neg h(c) \ \wedge \bigwedge_{c \notin \Delta} h(c) \ \wedge \ OBS \nvdash \bot$$

We say that Δ is a minimal diagnosis *if no proper subset $\Delta' \subset \Delta$ is a diagnosis, and that Δ is a* minimal cardinality diagnosis *if no other diagnosis $\Delta' \subseteq COMPS$ exists such that $|\Delta'| < |\Delta|$.*

For the MBD of Figure 1, Δ_1=$\{X_1, X_2\}$, Δ_2=$\{O_1\}$, Δ_3=$\{A_2\}$ are minimal diagnoses, and Δ_2, Δ_3 are minimal cardinality diagnoses, as there is no smaller diagnosis.

Minimal subset diagnoses are especially in the interest of an MBD engine since based on Definition 2 every superset

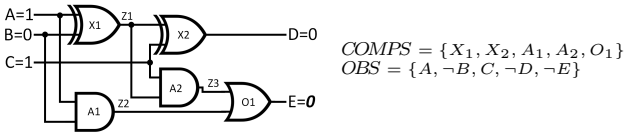

Figure 1: MBD: A full adder.

of a minimal subset diagnosis is a diagnosis too. Therefore, when focusing on the minimal subset diagnoses we actually represent the superset space of those diagnoses. The number of subset minimal diagnoses is typically huge, and therefore a reduced set that contains only the minimal cardinality diagnoses is in the interest of an MBD engine too. Those diagnoses are considered since with no information about the fault probability of the components we can assume identical probability for all the components and thus minimal cardinality diagnoses represent actually those sets with the highest probability.

Beyond the fact that the number of minimal subset and cardinality diagnoses is exponential, MBD is known to be a hard problem where algorithms have exponential runtime (in the size of the system). In terms of complexity, computing the first minimal subset diagnosis is in P, but computing the next one is NP-hard [Bylander *et al.*, 1991]. Computing the minimal cardinality is NP-hard, even for the first diagnosis [Selman and Levesque, 1990]. Both reasons, the exponential number of minimal diagnoses and the exponential computation, motivate an MBD engine to return a reduced set of diagnoses.

An additional challenge is raised as a result of computing a complete set of minimal subset or cardinality diagnoses: how to prioritize the diagnoses? One way to prioritize the diagnoses is by the prior fault probability of the components. With the independent assumption, the probability of a diagnosis can be determined by the product of the components it contains. With no prior probability the diagnoses can be prioritized based on their cardinality.

There are two methods to discriminate the actual diagnosis, the diagnosis that actually contains the faulty components, either by testing or probing [de Kleer and Williams, 1987b]. In the testing method the diagnosis process is run through additional input vectors. Under the assumption that faulty components in the system remain permanently faulty along different input vectors, we can prune diagnoses that are inconsistent with multiple observations. The probing task is similar, but instead of running the diagnosis on a new input vector, the probes are requests on the observation of the output of internal components. Probes can prune diagnoses that are not consistent with the new internal observation. Both methods can be executed iteratively until a single diagnosis is found.

The main challenge in both methods is to reduce the number of probes (tests). A common greedy approach to address this challenge is to choose a probe (test) that maximizes the information gain [Feldman *et al.*, 2010b]. Specifically, given the probability of each diagnosis in the diagnoses set we can measure the entropy of the diagnoses set. The information gain is the difference between the entropy of the diagnoses sets before and after activating a probe (test).

The testing and probing processes may be expensive in the number of probes (tests) required to focus on the actual diagnosis. Furthermore, in some cases the output of a diagnosis algorithm should be displayed to a human operator. A

	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5	Δ_6	CFP
C1	1	0	0	0	0	0	**0.2**
C2	1	0	0	0	0	0	**0.2**
C3	0	1	1	1	1	0	**0.8**
C4	0	1	0	0	0	0	**0.16**
C5	0	0	1	0	0	0	**0.16**
C6	0	0	0	1	0	0	**0.16**
C7	0	0	0	0	1	0	**0.16**
C8	0	0	0	0	0	1	**0.16**
p	0.2	0.16	0.16	0.16	0.16	0.16	

Table 1: Diagnoses and CFP for Figure 2

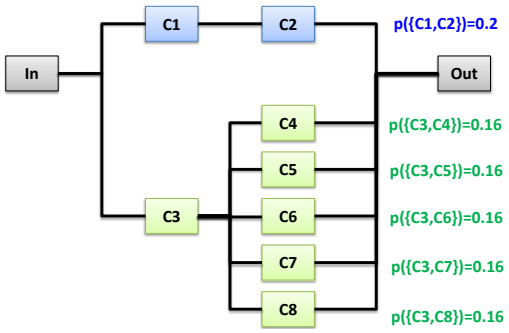

Figure 2: An example where viewing the most probable diagnosis can be more misleading than viewing the CFP. The most probable diagnosis is $\{C1, C2\}$, having a probability of 0.2. The CFP, however, would show that $C3$ has a 0.8 probability of being faulty, and is therefore more likely to be faulty than $C1$ and $C2$.

reasonable question that a user might raise is "what is the probability that a component C is faulty?". This is helpful, for example, to decide which component should be replaced first. More generally, a human diagnosis algorithm operator may wish an estimate of the probability that each component is faulty. Next, we discuss this question and its implications on how diagnosis algorithms should run.

3 Component Fault Probability (CFP)

We use the term Component Fault Probability (CFP) to denote a mapping of components to an estimate of the probability that they are faulty.

Definition 3. [Component Fault Probability (CFP)] *A CFP is a mapping* $COMPS :\rightarrow [0, 1]$ *intended to estimate the probability that a given component is faulty.*

A similar notion to the CFP was introduced in prior work [Stern *et al.*, 2012; Feldman *et al.*, 2013]. Following these prior work, a CFP can be generated from a set of diagnoses that were found by a diagnosis algorithm (DA) as follows.

Let Ω be a set of diagnoses found by a DA, and let $p : \Omega \rightarrow [0, 1]$ be a probability distribution over the diagnoses in Ω, corresponding to the probability that each diagnosis is correct. Many diagnosis algorithms generate this, for example, by considering a prior probability (without considering the observations) on the fault of each component and considering the probability of a diagnosis Δ (denoted $p(\Delta)$) as the product of the prior probability of the components in Δ, i.e., $p(\Delta) = \prod_{C\in\Delta} p(C)$, where $p(C)$ is the prior probability that C is faulty.[1]

A CFP can be generated from Ω and p as follows:

$$CFP(C) = \sum_{\Delta\in\Omega} p(\Delta) \cdot \mathbb{1}_{C\in\Delta} \qquad (1)$$

where $\mathbb{1}_{C\in\Delta}$ is the indicator function defined as:

$$\mathbb{1}_{C\in\Delta} = \begin{cases} 1 & C \in \Delta \\ 0 & \text{otherwise} \end{cases}$$

This way to generate CFP from Ω and p is correct if p assigns correct and independent probabilities to the diagnoses in Ω. Thus, we only consider this way to generate a CFP from Ω and p in this paper, and refer to this process simply as *generating a CFP from* Ω *and* p.

Our main argument in this paper is that presenting a human operator with a CFP is more meaningful and helpful than a long list of diagnoses. As mentioned above, consistency-based DAs may return a very large number of diagnoses. A human operator cannot reason effectively about a long list of diagnoses. One might consider presenting a short list of only the most probable diagnoses to the operator. This approach, however, may be misleading. For example, consider the system depicted in Figure 2, and assume that a DA has returned five diagnoses $\{C3, C4\}, \{C3, C5\}, \{C3, C6\}, \{C3, C7\}, \{C3, C8\}\}$ with a probability of 0.16 each, and another diagnosis $\{C1, C2\}$ with a probability of 0.2[2]. The most probable diagnosis is $\{C1, C2\}$, having a probability of 0.2. The CFP, however, would point at $C3$ as the component that is most likely to be faulty, having $CFP(C3) = 0.8$.

The CFP poses an informative aggregation of a set of diagnoses. However, a set of diagnoses contains more information than the CFP that is generated from it. The information that is lost is the dependency between the different components. For example, consider again the diagnoses in Figure 2. $C1$ only exists in a single diagnosis $\{C1, C2\}$. Thus, $C1$ is only faulty if component $C2$ is faulty as well. This relation is lost in a CFP. Automated algorithms for deciding subsequent tests or probes [Feldman *et al.*, 2010b; de Kleer and Williams, 1987b; Shakeri *et al.*, 2000] might make use of this additional relation information and might thus prefer as input a list of diagnoses over just the CFP. In this paper we focus on diagnosis algorithms whose output is displayed to a human operator. A human operator would find it difficult to reason about a list of diagnoses and would benefit from the aggregated view provided by a CFP.

3.1 Evaluating CFPs

Using Equation 1, one can generate a CFP from a list of diagnoses generated by any MBD algorithm. To evaluate the quality of a generated CFP, we consider the "optimal" CFP, denoted by CFP* and defined as follows:

$$CFP^*(C) = \begin{cases} 1 & C \text{ is faulty} \\ 0 & \text{otherwise} \end{cases}$$

CFP* can be viewed as an *offline* optimal CFP, and a CFP can be evaluated by comparing how "close" it is to CFP*. One can think of many possible distance metrics to measure

[1] To verify that p is a valid probability distribution, one is also required to normalized their sum to one.

[2] The probability of the diagnoses is normalized and therefore their sum is equal to 1.

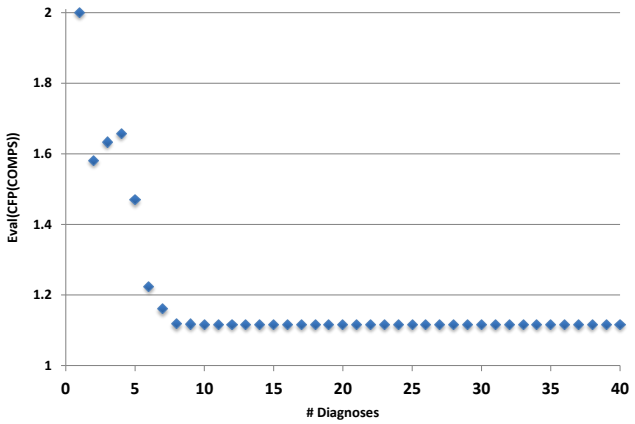

Figure 3: An example where adding more diagnoses improves the CFP quality.

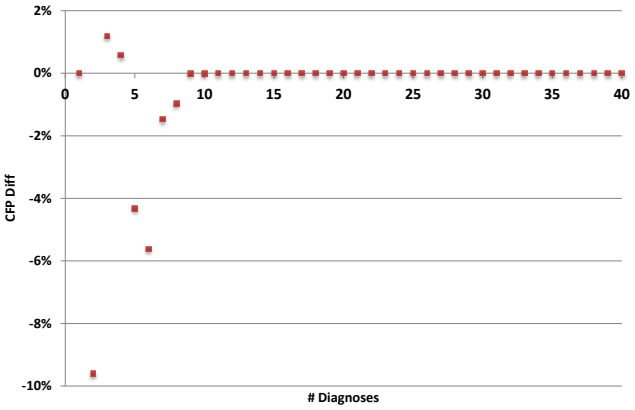

Figure 4: An example showing how the CFP diff converges to zero as more diagnoses are considered.

the "closeness" of a given CFP and CFP*. We chose to use a simple vector distance:

$$Eval(CFP(COMPS)) = \sqrt{\sum_{C \in COMPS}(CFP(C) - CFP^*(C))^2} \quad (2)$$

Lower distance indicates a better CFP. Eval(CFP(COMPS))=0 indicates a diagnosis set that contains only a single diagnosis of faulty components solely.

Introducing CFPs and a way to evaluate them (Equation 2) raises the challenge of constructing diagnosis algorithms that generate quickly high quality CFPs. Next, we evaluate the quality of the CFPs generated from a set of subset minimal diagnoses and the impact of finding more diagnoses to the resulting CFP.

4 Experimental Results

We performed experiments on the 74xxx benchmark Boolean circuits, using observations from the "synthetic track" in the annual DXC diagnosis competition of 2009.[3]

For each observation we run a breadth first search to find all minimal subset diagnoses until either all minimal subset diagnoses were found or until 40 minimal subset diagnoses were found.[4] The resulting set of minimal subset diagnoses were then ordered by cardinality, starting from the diagnoses with the minimal cardinality diagnoses. Let Γ_{inc} be this sequence of diagnoses, where $\Gamma_{inc}[i]$ denotes the i^{th} diagnosis in the sequence.

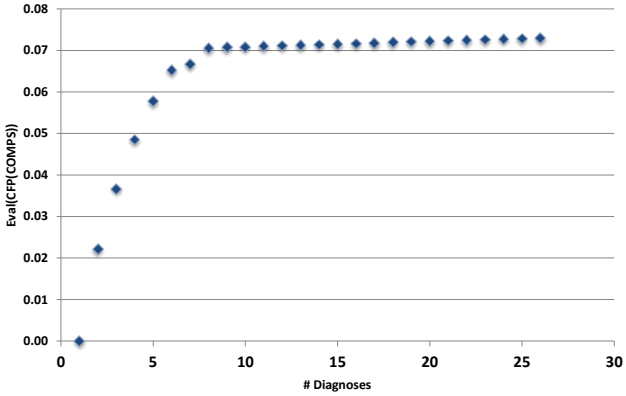

Figure 6: An example where adding more diagnoses only degrades the CFP quality.

For every diagnosis in Γ_{inc} we computed the CFP based on the union of that diagnosis and all the diagnoses that preceded it in Γ_{inc}. Thus, the first CFP was computed based on the diagnosis set that contains only a single diagnosis $\Gamma_{inc}[1]$, and then the second was computed based on a diagnosis set that contains $\{\Gamma_{inc}[1], \Gamma_{inc}[2]\}$, etc. Let $CFP_{inc}[i]$ denote the CFP generated from $\{\Gamma_{inc}[1], \Gamma_{inc}[2], ..., \Gamma_{inc}[i]\}$ and let $Eval(CFP_{inc}[i])$ denote its quality (measured using Equation 2). For every $CFP_{inc}[i]$ we measured the difference in CFP quality between two subsequent CFPs (using Equation 2) and normalize the result by dividing it with the square root of the number of components in the diagnosed system. We call this measure "CFP diff", computed for the i^{th} diagnosis in Γ_{inc} as follows:

$$\frac{Eval(CFP_{inc}[i]) - Eval(CFP_{inc}[i-1])}{\sqrt{|COMPS|}} \quad (3)$$

CFP diff is intended to measure the improvement of CFP by searching for additional diagnoses. Figures 3 and 4 show the CFP quality and the CFP diff, respectively, as a function of the number of diagnoses found, for a single observation of the 74182 system.

Figure 5 shows the CFP diff as a function of the number of diagnoses used to generate that CFP for all the 74xxx systems and all the observation set in the DXC benchmark set. Each of the red x mark data points correspond to a CFP diff of a specific observation as a function of the number diagnoses seen so far.

As can be seen, the value of adding diagnoses converges very close to zero. This suggests that finding more diagnoses is not always helpful, and that finding all minimal diagnoses is not needed. Thus, a smart diagnosis algorithm might exploit this by finding only some of the minimal diagnoses instead of all them. This is expected to result in a better runtime and practically the same quality of CFP.

[3] See details in the DXC 09 website: `http://sites.google.com/site/dxcompetition2009/`.

[4] 40 was chosen to avoid exhausting memory or time. Other bounds are also possible.

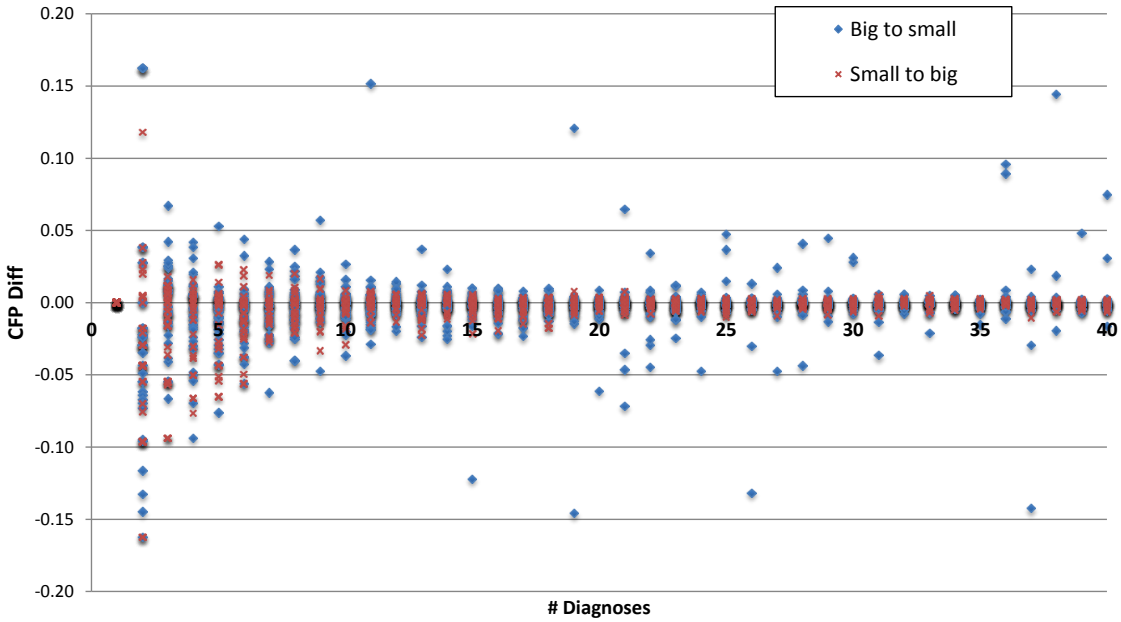

Figure 5: CFP diff as a function of the number of diagnoses found, when searching for subset minimal diagnoses. The red x marks are for experiments where the diagnoses were found in order of increasing cardinality, while the blue diamonds are for when diagnoses where found in the reverse order.

The convergence of the CFP diff to zero, observed in Figure 5, suggests that in every observation the CFP quality converges to a specific value as more consistent diagnoses are found. This can be seen clearly also in Figure 3, where the CFP quality converges to approximately 1.16 as more diagnoses are found. The converged CFP quality is not necessarily better than the CFP quality obtained by considering only a subset of the consistent diagnoses. For example, Figure 6 shows the CFP quality as a function of the number of diagnoses found for a different observation. In this observation, the CFP quality starts at the optimal value (zero distance from CFP^*). This is since the correct diagnosis happened to be the first diagnosis that was found. As more diagnoses are found, the CFP quality degrades. Thus, considering more diagnoses is not always helpful. Since when running a diagnosis algorithm one does not know CFP^*, it is a challenge to identify when adding more diagnoses does not help.

Next, consider the blue diamonds in Figure 5. These diamonds show the CFP diff for the same set of experiments, where the order of the sequence of diagnoses is reversed. This means that the first diagnoses considered is not the minimal cardinality diagnoses, but rather the maximum cardinality diagnoses. As can be seen, while a general trend of convergence to zero exists to some extent, it is much more noisy then when diagnoses were considered in order of increasing cardinality. This demonstrates the potential importance of the order by which diagnoses are found by the diagnosis algorithm.

5 Related Work

Many of the existing diagnosis techniques propose to apply a combination of deterministic reasoning and search algorithms. One classic approach involves a two stage process. First, it identifies conflict sets, each of which includes at least one fault. Then, it applies a hitting set algorithm to compute sets of multiple faults that explain the observation [de Kleer and Williams, 1987a; Williams and Ragno, 2007]. These methods guarantee sound diagnoses, and some of them are even complete. However, they tend to fail for large systems due to infeasible runtime or space requirements.

Compilation-based methods have also been proposed in the MBD context. Torasso and Torta apply BDDs to compile the model [Torasso and Torta, 2006]. Darwiche [Darwiche, 2001] compiles a system description into Decomposable Negation Normal Form (DNNF) where a minimal cardinality diagnosis can be found in time that is polynomial in the size of the DNNF. However, the size of the DNNF may grow exponentially and is shown to become a bottleneck [Siddiqi and Huang, 2007].

Feldman *et al.* [2010a] propose a stochastic diagnosis algorithm called SAFARI. In contrast to the above, SAFARI does not try to compute the set of all diagnoses and it does not guarantee to find minimal cardinality diagnoses. Its advantage is that it is very fast. Like other MBD algorithms, it tries to generate as many small diagnoses as possible and does not consider aggregating the resulting set of diagnoses. Thus, the work in our paper is orthogonal to SAFARI, as a CFP can be generated from the diagnoses found by SAFARI using Equation 1.

Keren *et al.* [2011] present an alternative approach to diagnosis that combines MBD with multi-label classification. They propose to build a classifier that maps symptoms (observations) of the system to possible faults. The major advantage of this approach is in reducing significantly the online computational complexity; The learning process of the relations between observations and the diagnoses is performed in advance offline. Afterwards (online), a diagnosis can be computed immediately by using the classifier that was learned offline. Unlike other diagnosis algorithms mentioned above, this machine learning approach to diagnosis returns a single diagnosis and not a set of diagnosis. Similar

to our approach, the output of this machine learning based diagnosis algorithm is not measured by its consistency but by its distance to the real diagnosis (the faults), using standard classification metrics such as false positives and false negatives. In our work we evaluated the CFP with a simple vector distance metric instead. In future work, however, we plan also to evaluate our results using machine learning metrics such as false positives and negatives, precision and recall.

Maier et al. [2011] pointed out that often AI problems lie at the intersection of the fields of model-based diagnosis and probabilistic reasoning and that probabilistic reasoning can be a promising alternative to the model-based diagnosis approaches. They use a Bayesian networks (BNs) results from first-order model-based diagnosis formalism for this translation to first-order probabilistic reasoning framework. Similar to our approach, they also remark that solutions to AI problems in engineering domains need to be compactly represented for the needs of engineers. However, by compactly represented they mean auto-generating low-level representation such as BNs that can be used as an input to off-the-shelf tools while we aim to find a more general and simple approach for compact representation not by finding the connections and dependencies between components but rather by addressing the logic model of the system done by aggregating the diagnoses to CFP.

6 Conclusion and Future Work

In this work we proposed an alternative form of output to diagnosis algorithm called the *component fault distribution*, or CFP in short. CFP maps every component in the diagnosed system to a probability that this component is faulty. We argue that a CFP is a more reasonable output than a single or k most probable diagnoses since it contains aggregated information over all the found diagnoses. We also argue that a CFP is more reasonable to a human operator than a long list of diagnoses. CFPs can be generated from a list of diagnoses that are returned by a diagnosis algorithm, and a way to evaluate CFPs is proposed. Empirical evaluation on the 74xxx system suggests that the quality of a CFP converges quickly as more diagnoses are found.

This observation leads to considering, in future work, how to develop a diagnosis algorithm that finds a small but representative set of diagnoses that will generate a high quality CFP. One might even consider generating a CFP from "almost" consistent diagnoses - set of components that if assumed to be faulty explains almost all the observed system outputs.

References

[Bylander *et al.*, 1991] Tom Bylander, Dean Allemang, Michael C. Tanner, and John R. Josephson. The computational complexity of abduction. *Artif. Intell.*, 49(1-3):25–60, 1991.

[Darwiche, 2001] Adnan Darwiche. Decomposable negation normal form. *Journal of the ACM*, 48(4):608–647, 2001.

[de Kleer and Williams, 1987a] J. de Kleer and B. C. Williams. Diagnosing multiple faults. *Artif. Intell.*, 32(1):97–130, 1987.

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

[de Kleer *et al.*, 1992] Johan de Kleer, Alan K. Mackworth, and Raymond Reiter. Characterizing diagnoses and systems. *Artificial Intelligence*, 56(2-3):197–222, 1992.

[Feldman *et al.*, 2010a] Alexander Feldman, Gregory Provan, and Arjan van Gemund. Approximate model-based diagnosis using greedy stochastic search. *J. Artif. Int. Res.*, 38:371–413, May 2010.

[Feldman *et al.*, 2010b] Alexander Feldman, Gregory Provan, and Arjan van Gemund. A model-based active testing approach to sequential diagnosis. *J. Artif. Int. Res.*, 39:301–334, September 2010.

[Feldman *et al.*, 2013] Alexander Feldman, Helena Vicente de Castro, Arjan van Gemund, and Gregory Provan. Model-based diagnostic decision-support system for satellites. In *Aerospace Conference, 2013 IEEE*, pages 1–14. IEEE, 2013.

[Keren *et al.*, 2011] Betty Keren, Meir Kalech, and Lior Rokach. Model-based diagnosis with multi-label classification. In *The 22nd International Workshop Principles of Diagnosis (DX-11)*, 2011.

[Maier *et al.*, 2011] Paul Maier, Dominik Jain, and Martin Sachenbacher. Diagnostic hypothesis enumeration vs. probabilistic inference for hierarchical automata models. In *the International Workshop on Principles of Diagnosis (DX)*, Murnau, Germany, 2011.

[Reiter, 1987a] R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32(1):57–96, 1987.

[Reiter, 1987b] Raymond Reiter. A theory of diagnosis from first principles. *Artif. Intell.*, 32(1):57–95, 1987.

[Selman and Levesque, 1990] Bart Selman and Hector J. Levesque. Abductive and default reasoning: A computational core. In *AAAI*, pages 343–348, 1990.

[Shakeri *et al.*, 2000] Mojdeh Shakeri, Vijaya Raghavan, Krishna R Pattipati, and Ann Patterson-Hine. Sequential testing algorithms for multiple fault diagnosis. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 30(1):1–14, 2000.

[Siddiqi and Huang, 2007] Sajjad Ahmed Siddiqi and Jinbo Huang. Hierarchical diagnosis of multiple faults. In *IJCAI*, pages 581–586, 2007.

[Siddiqi, 2011] Sajjad Ahmed Siddiqi. Computing minimum-cardinality diagnoses by model relaxation. In *IJCAI*, pages 1087–1092, 2011.

[Stern *et al.*, 2012] Roni Stern, Meir Kalech, Niv Gafni, Yair Ofir, and Eliav Ben-Zaken. Using model-based diagnosis to improve software testing. In *the International Workshop on the Principles of Diagnosis (DX)*, 2012.

[Torasso and Torta, 2006] Pietro Torasso and Gianluca Torta. Model-based diagnosis through obdd compilation: A complexity analysis. In *Reasoning, Action and Interaction in AI Theories and Systems*, pages 287–305, 2006.

[Williams and Ragno, 2007] Brian C. Williams and Robert J. Ragno. Conflict-directed A* and its role in model-based embedded systems. *Discrete Appl. Math.*, 155(12):1562–1595, 2007.