

# Model-Based Diagnosis with Uncertain Observations

Dean Cazes, Meir Kalech

Ben-Gurion University of the Negev, Israel  
deanc@post.bgu.ac.il, kalech@bgu.ac.il

## Abstract

Classical model-based diagnosis uses a model of the system to infer diagnoses – explanations – of a given abnormal observation. In this work, we explore how to address the case where there is uncertainty over a given observation. This can happen, for example, when the observations are collected by noisy sensors, that are known to return incorrect observations with some probability. We formally define this common scenario for consistency-based and abductive models. In addition, we analyze the complexity of two complete algorithms we propose for finding all diagnoses and correctly ranking them. Finally, we propose a third algorithm that returns the most probable diagnosis without finding all possible diagnoses. Experimental evaluation shows that this third algorithm can be very effective in cases where the number of faults is small and the uncertainty over the observations is not large. If, however, all possible diagnoses are desired, then the choice between the first two algorithms depends on whether the domain’s diagnosis form is abductive or consistent.

## 1 Introduction

Model-based diagnosis (MBD) is a principled approach for automated diagnosing the root cause of faults in systems (de Kleer and Williams 1987; Reiter 1987; Stern et al. 2012). In MBD, a model of the system is needed along with observations of the system’s behavior. These observations are checked against the given model, and inference algorithms are used to produce *diagnoses*, which are possible assumptions about which components are faulty that are consistent with the given model and the observations.

Previous works assume the observations are certain. Unfortunately, this assumption is not always correct. Experiments show that for some cases a completely different diagnosis is output when considering uncertainty.

In this paper, we address the MBD problem where there is an uncertainty over the observations. This can happen when a noisy sensor is used to collect the output. A noise can be interpreted differently across domains. In the Boolean circuits domain, for instance, the sensor collecting the output (1 or 0) might be affected by spontaneous electric pulses, in addition to the original output, causing the observed output to flip its value.

Copyright © 2020, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

We formalize the MBD problem with uncertain observations, both for abductive and consistency-based diagnosis. An abductive model is a model where for each component, all its behaviors are known and defined (Reiter 1987). For instance, if for each component in a Boolean circuit model (e.g. an *OR* gate), its output can be one of the three: (1) healthy (i.e. logical *OR*) (2) flipped (i.e. logical *NOR*) or (3) stuck at 1 (i.e. output is always 1), then we can say this model is abductive. A diagnosis in an abductive model is an assignment of a behavior mode for each component. A consistent model is a model where there exist components with an undefined behavior (Console, Dupré, and Torasso 1991). This means that given an input to this component, there can be multiple possible outputs. This is contrary to an abductive model, where a component and a behavior mode map to a single possible output.

We propose three algorithms to solve the MBD with uncertain observations problem. The first, *O2D*, reduces the uncertainty into multiple MBD problems with certainty. The second, *D2O*, saves the effort of applying an MBD solver multiple times, by iterating all behavior modes assignment combinations and for each - deriving the observation(s) it explains. Both algorithms ultimately find all of the diagnoses and their probability. If only the most probable diagnosis is desired, a third algorithm, *MstLikeDiag*, is proposed to handle this case. This algorithm is similar to the first algorithm, only that it stops after a certain condition is met. Meeting this condition ensures no other unseen diagnosis can have a probability larger than our current maximum.

Evaluation on Boolean circuit diagnosis for the abductive domain and software projects for the consistent domain, compares the runtime over these algorithms under the influence of some independent variables. The results on the abductive domain show an unequivocal preference of *D2O* to *O2D*, independently of other variables. In the consistency-based domain, in the other hand, *O2D* outperforms *D2O*. Also, we conclude that *MstLikeDiag* runs faster than the other two under certain circumstances that will be discussed later.

## 2 Related Work

MBD has been studied in a range of different settings and models (Feldman, Provan, and Van Gemund 2010; Stern et al. 2012; Metodi et al. 2014) including discrete event sys-

tems (DES) (Pencolé and Cordier 2005), qualitative models (Struss and Price 2003), incomplete causal models (Console, Dupré, and Torasso 1989) and hybrid systems (Narasimhan and Biswas 2007).

Previous work of Lamperti and Zanella (2000; 2010) also address uncertain observations. However, their work focus on DES and the uncertainty is addressed to observable events, which are the inputs to the system. Our work discusses the uncertainty of the outputs of the observations. In addition, they do not use probabilistic measures to rank the diagnoses in the new problem. A Bayesian network can also solve an MBD problem with uncertainty (Lucas 2001; Flesch, Lucas, and van der Weide 2007). In this setting, the uncertainty is not over the observations, but over the domain and behavior modes. This means that the observations are certain, but behavior modes of components are distributed and affected by the health modes of other components. Cardoso et al. (2016) consider the case where software components are not distinctly either healthy or faulty, but could be also in between. This obviously causes system tests to output a continuous value rather than a binary response of pass/fail. The outputted value could indicate to what extent the system is faulty. For example, high latencies of responses to queries may indicate that servers are overloaded. Cardoso et al. propose to model the uncertain observation with a fuzzy error function and integrate this information in spectrum-based fault localization diagnosis framework (Abreu, Zoetewij, and van Gemund 2009; Elmishali, Stern, and Kalech 2019). Notice that the uncertainty in this case is not due to uncertain observation as addressed in our paper but due to uncertainty in the faulty state of the component. To the best of our knowledge no other work has addressed the problem of MBD with uncertain observations' outputs.

### 3 Background

A *system* is composed of a set of components  $COMPS$ . Every component  $c \in COMPS$  has a set of input and output variables, denoted  $in(c)$  and  $out(c)$ . Some of the components inputs and outputs are *observables*, that is, it is possible to observe their values when monitoring the system. Let  $O$  denote this set of values. An *observation*  $OBS$  is an assignment of values for each  $o \in O$ . For an observation  $OBS$  and an observable variable  $o \in O$  we denote by  $OBS(o)$  the value assigned to  $o$  in  $OBS$ .

The *behavior* of component  $c$  represents a relation between values set to  $in(c)$  and the values to  $out(c)$ . Every component has a fixed set of possible *behavior modes*, where every behavior mode represents a different behavior. Every component has exactly one *healthy* behavior mode, which corresponds to the normal behavior of that component. All other behavior modes are referred to as *faulty* behavior modes. A *system description* ( $SD$ ) is a formal model that represents the possible behaviors of every component in the system. A *behavior mode assignment* is a function mapping a component to one of its behavior mode. If in a behavior mode assignment  $\omega$  a component  $c$  is mapped to its healthy behavior mode then we say that  $c$  is healthy w.r.t  $\omega$ .

MBD is designed for cases where an observation is inconsistent with the assumption that all components are healthy. More formally, an MBD problem is defined by  $\langle COMPS, SD, OBS \rangle$  where  $COMPS$  is the set of system components,  $SD$  is a system description, and  $OBS$  is an observation, for which it holds that

$$(SD \wedge OBS \wedge \bigwedge_{c \in COMPS} h(c)) \vdash \perp \quad (1)$$

where  $h(c)$  denotes that the behavior mode of  $c$  was healthy when  $OBS$  was collected.

A *diagnosis* is a behavior mode assignment *explaining* the observation. The literature on MBD proposes several definition of what it means to explain an observation. In the *consistency-based* diagnosis literature, a behavior mode assignment  $\omega$  explains an observation  $OBS$ , i.e.,  $\omega$  is a diagnosis, iff  $SD \wedge OBS \wedge \omega$  is consistent.

In the *abductive diagnosis* literature, a more detailed  $SD$  is assumed. First, every behavior mode of a component  $c$  defines a *function* that maps values assigned to the variables in  $in(c)$  to the values assigned to the variables in  $out(c)$ . Note that in consistency-based diagnosis, a behavior mode of a component  $c$  defines a *relation* between  $in(c)$  and  $out(c)$ , including non-deterministic relations in which for some input values for  $in(c)$  there may be more than one possible output values. Such non-determinism is not allowed in abductive diagnosis.

A second difference between consistency-based diagnosis and abductive diagnosis is that in abductive diagnosis, additional information is available about the state of the system at the time when the observation was collected. This information is referred to as the *context* and denoted  $Ctx$ . We limit our scope to cases where the context includes all the *system inputs*. The system inputs, denoted  $SYSIN$ , are all the components' inputs that are not outputs of any component in the system. That is,  $Ctx = SYSIN = \bigcup_{c \in COMPS} in(c) \setminus$

$\bigcup_{c \in COMPS} out(c)$ . The system outputs, denoted  $SYSOUT$ , are all the observables that are not system inputs.

In abductive diagnosis, a behavior mode assignment ( $\omega$ ) entails the observation, i.e.,  $\omega$  is a diagnosis, iff

$$SD \wedge Ctx \wedge \omega \rightarrow OBS \quad (2)$$

For a comprehensive discussion about the relation between abductive diagnosis and consistent diagnosis, see Torasso et al. (Console and Torasso 1991).

#### 3.1 Diagnosis Likelihood

When there is more than one diagnosis (either abductive or consistent) for a given MBD problem, one needs a way to compute the likelihood that a given diagnosis is *correct*. A diagnosis is correct if it assigns to every component the behavior mode that it had when the system was observed. We denote by  $P(\omega|OBS)$  the likelihood that the diagnosis  $\omega$  is correct given that  $OBS$  was observed. Many diagnosis algorithms output, in addition to a set of diagnoses, an estimate of their likelihoods.

One way to estimate the likelihood of a diagnosis  $\omega$  is as follows. Assume diagnosis  $\omega$ , and let  $\omega^+$  be the set of components assigned a healthy behavior mode, and  $\omega^-$  the complementing set of components. Let  $p(c)$  be the a-priori probability that component  $c$  is healthy. If we assume that components fail independently, then the a-priori probability that a diagnosis  $\omega$  is correct, denoted  $P(\omega)$ , is given by:  $P(\omega) = \prod_{c \in \omega^+} p(c) \cdot \prod_{c \in \omega^-} (1 - p(c))$ .

Let  $\Omega$  be the set of diagnoses, if we assume that for every pair of diagnoses  $\omega, \omega' \in \Omega$  the observation  $OBS$  is not more likely for one diagnosis over another, i.e.,  $P(OBS|\omega) = P(OBS|\omega')$ , then the posterior probability  $P(\omega|OBS)$  is given by:  $P(\omega|OBS) = \frac{P(\omega)}{\sum_{\omega' \in \Omega} P(\omega')}$ .

We use the above to compute the *diagnosis likelihood*, as was done by many prior works (de Kleer and Williams 1987; Stern et al. 2017).

## 4 Uncertainty over Observations

In this work, we assume that a subset of the observables  $O' \subseteq O$  cannot be observed accurately. For example, if an observable corresponds to reading the value of some sensor, then the sensor manufacturer may provide information about that sensor's expected accuracy.

We formalize this by defining the notion of an *uncertain observation*, as follows. Let  $dom(o)$  be the domain of an observable  $o \in O$ , i.e., the set of values that may be assigned to  $o$  in an observation. We assume throughout the paper that each observable can have discrete values only, thus  $dom(o)$  is finite.

An *uncertain observation* is a function that maps every observable  $o$  and value  $v \in dom(o)$  to the probability that the value of  $o$  is  $v$ . That is, an uncertain observation defines for every observable  $o \in O$  a distribution over its values. For an uncertain observation  $UOBS$ , an observable  $o \in O$ , and a value  $v \in dom(o)$ , we denote by  $UOBS(o, v)$  the probability that the value of  $o$  is  $v$ . Note that there may be an observable  $o$  for which there is no uncertainty over its values. This can be expressed by setting  $UOBS(o, v) = 1$ .

In this work, we make the simplifying assumption that distributions of values for observables are independent. This means the probability that an observable  $o$  has a value  $v \in dom(o)$  is independent from the probability that a different observable  $o'$  has some value  $v' \in dom(o')$ . Therefore, for a given uncertain observation  $UOBS$  we can compute the probability that the observation is in fact  $OBS$ , denoted  $UOBS(OBS)$  as follows:  $UOBS(OBS) = \prod_{o \in O} UOBS(o, OBS(o))$ . An observation  $OBS$  is called *feasible* w.r.t an uncertain observation  $UOBS$  iff  $UOBS(OBS) > 0$ . Let  $F(UOBS)$  be the set of feasible observations for  $UOBS$ . We define the notion of abductive diagnosis and consistent diagnosis for an uncertain observation as follows.

**Definition 1** (Diagnosis for an Uncertain Observation). *A behavior mode assignment  $\omega$  is a consistent diagnosis for an uncertain observation  $UOBS$  iff*

$$\exists OBS \in F(UOBS) \text{ s.t. : } SD \wedge \omega \wedge OBS \not\perp \quad (3)$$

*A behavior mode assignment  $\omega$  is an abductive diagnosis for*

*an uncertain observation  $UOBS$  iff*

$$\exists OBS \in F(UOBS) \text{ s.t. : } SD \wedge Ctx \wedge \omega \rightarrow OBS \quad (4)$$

Given an uncertain observation  $UOBS$  and a diagnosis  $\omega$  for this respective MBD problem, the likelihood that  $\omega$  is correct is given by:

$$P(\omega|UOBS) = \sum_{OBS \in F(UOBS)} P(\omega|OBS) \cdot P(OBS) \quad (5)$$

We refer to the problem of finding diagnoses and their likelihoods for a given  $UOBS$  as the *UO-MBD problem*.

### 4.1 The Relation Between Observations and Diagnoses

For a given uncertain observation  $UOBS$ , it is interesting to consider the relation between its feasible observations and its set of diagnoses.

By definition (Eq. 3 and 4), for every diagnosis  $\omega$  of  $UOBS$  there exists at least one feasible observation  $OBS$  for which  $\omega$  is a diagnosis for. But can there be other diagnoses that are also diagnoses of  $OBS$ ? and vice versa, can  $\omega$  be a diagnosis to other observations? As we discuss later in this paper, these questions have an impact on how one can compute diagnosis likelihoods.

For consistency-based diagnosis, a single diagnosis may be consistent with multiple feasible observations. Similarly, a single observation may have multiple consistent diagnoses. So, there is a many-to-many relationship between diagnoses and feasible observations.

This is not the case for abductive diagnosis. There, a diagnosis entails exactly one feasible observation. On the other hand, more than one diagnosis may entail the same feasible observation. Hence, there is a many-to-one relation between diagnoses to feasible observations. This is useful, for example, to compute diagnosis likelihood (Eq. 5), since once a single observation is found for a given diagnosis, we have the exact likelihood.

## 5 Solving UO-MBD

### 5.1 From Observations to Diagnoses

The first approach we propose for solving a *UO-MBD* problem is straightforward: create an *MBD* problem for every feasible observation, solve it with an off-the-shelf *MBD* algorithm, and aggregate the diagnoses likelihoods obtained for each feasible observation using Eq. 5. We call this approach *observations to diagnoses (O2D)*. Algorithm 1 lists a pseudo code for *O2D*.

*O2D* maintains two data structures  $\Omega$  and  $p$ .  $\Omega$  is the set of diagnoses that were found so far.  $p$  is a dictionary, mapping every found diagnosis to the current estimate of its likelihood. Initially, both  $\Omega$  and  $p$  are empty (line 1 in Alg. 1). For every feasible observation  $OBS \in F(UOBS)$ , *O2D* executes an off-the-shelf *MBD* algorithm to obtain the set of diagnoses and their likelihoods with respect to  $OBS$  (line 3). The former is denoted by  $\Omega_{OBS}$  and latter by  $p_{OBS}$ . Every diagnosis in  $\Omega_{OBS}$  is added to  $\Omega$ , since by definition it is also a diagnosis for  $UOBS$ . For every such diagnosis

---

**Algorithm 1:** Pseudo-code for *O2D*.

---

**Input:** UOBS, an uncertain observation  
**Input:** *SD*, the system description  
**Input:** *COMPS*, the system components  
**Output:**  $\Omega$ , a set of all diagnoses  
**Output:**  $p$ , a dictionary mapping diagnoses to their likelihoods

```
1  $\Omega \leftarrow \emptyset$ ;  $p \leftarrow$  empty dictionary
2 for  $OBS \in F(\text{UOBS})$  do
3    $(\Omega_{OBS}, p_{OBS}) \leftarrow \text{DIAGNOSE}(SD, COMPS, OBS)$ 
4   for  $\omega \in \Omega_{OBS}$  do
5     if  $\omega \in \Omega$  then
6        $p(\omega) \leftarrow p(\omega) + \text{UOBS}(OBS) \cdot p_{OBS}(\omega)$ 
7     else
8        $p(\omega) \leftarrow \text{UOBS}(OBS) \cdot p_{OBS}(\omega)$ 
9        $\Omega \leftarrow \Omega \cup \{\omega\}$ 
10 return  $(\Omega, p)$ 
```

---

$\omega \in \Omega_{OBS}$  that was not added before to  $\Omega$ , we set its likelihood in  $p$  to  $\text{UOBS}(OBS) \cdot p_{OBS}(\omega)$  (line 8). Otherwise, we add  $\text{UOBS}(OBS) \cdot p_{OBS}(\omega)$  to the current likelihood estimate for  $\omega$  in  $p$  (line 6). This way, when *O2D* returns  $p$  (line 10), the likelihood of every diagnosis is exactly the summation defined in Equation 5. This *O2D* algorithm works exactly the same for consistency-based diagnosis and for abductive diagnosis.

## 5.2 From Diagnoses to Observations

The second approach we propose for solving a UO-MBD problem can be viewed as the reversed version of *O2D*. Instead of computing for every observation its set of diagnoses, we compute for every possible behavior mode assignment the set of observations that are consistent with it. Then, we collect the assignments that are consistent with feasible observations. We call this approach *diagnoses to observations (D2O)*.

Algorithm 2 lists a pseudo code for *D2O*. For every feasible observation  $OBS \in F(\text{UOBS})$ , we create an empty set  $\Omega_{OBS}$  (line 3). This set will later contain all the diagnoses consistent with  $OBS$ . Let  $\mathcal{B}$  be the set of all possible behavior mode assignments. For every possible behavior mode assignment  $\omega \in \mathcal{B}$ , the function *FINDOBS* returns the set of observations that are consistent with  $\omega$ . This is denoted as  $\mathcal{O}_\omega$  in the pseudo code. For every such observation  $OBS \in \mathcal{O}_\omega$ , we check if it is feasible. If so, we add  $\omega$  to  $\Omega_{OBS}$ . After going over all behavior mode assignments, we go over every feasible observation  $OBS$  and compute the likelihoods of its diagnoses. Finally, we aggregate for every diagnoses its scores over all observations.

## 5.3 Complexity Analysis

Let us define the following notations: 1.  $|\Omega|$ : the number of diagnoses. 2.  $F(\text{UOBS})$ : the set of feasible observations. 3.  $B_{max}$ : the maximal number of behavior modes for a single component. 4.  $\Omega_{max}$ : the maximal number of diag-

---

**Algorithm 2:** Pseudo-code for *D2O*.

---

**Input:** UOBS, an uncertain observation  
**Input:** *SD*, the system description  
**Input:** *COMPS*, the system components  
**Input:**  $\mathcal{B}$ , all possible behavior mode assignments  
**Output:**  $\Omega$ , a set of all diagnoses  
**Output:**  $p$ , a dictionary mapping diagnoses to their likelihoods

```
1  $\Omega \leftarrow \emptyset$ ;  $p \leftarrow$  empty dictionary
2 for  $OBS \in F(\text{UOBS})$  do
3    $\Omega_{OBS} \leftarrow \emptyset$ 
4   for  $\omega \in \mathcal{B}$  do
5      $\mathcal{O}_\omega \leftarrow \text{FINDOBS}(SD, COMPS, \omega)$ 
6     for  $OBS \in \mathcal{O}_\omega$  do
7       if  $OBS \in F(\text{UOBS})$  then
8         if  $\omega \notin \Omega$  then
9            $\Omega \leftarrow \Omega \cup \{\omega\}$ 
10         $\Omega_{OBS} \leftarrow \Omega_{OBS} \cup \{\omega\}$ 
11   for  $OBS \in F(\text{UOBS})$  do
12      $p_{SumOBS} \leftarrow \sum_{\omega \in \Omega_{OBS}} p(\omega)$ 
13     for  $\omega \in \Omega_{OBS}$  do
14        $p_{OBS}(\omega) \leftarrow Pr(\omega) / p_{SumOBS}$ 
15       if  $p$  contains  $\omega$  then
16          $p(\omega) \leftarrow p(\omega) + \text{UOBS}(OBS) \cdot p_{OBS}(\omega)$ 
17       else
18          $p(\omega) \leftarrow \text{UOBS}(OBS) \cdot p_{OBS}(\omega)$ 
19 return  $(p, \Omega)$ 
```

---

noses for a single feasible observation. 5.  $O_{max}$ : the maximal number of observations consistent with a single behavior mode assignment. 6.  $C_D$ : the computational complexity of running a diagnosis algorithm. 7.  $C_F$ : the computational complexity of finding all consistent observations.

The computational complexity of *O2D* is simply:

$$|F(\text{UOBS})| \cdot C_D \quad (6)$$

The computational complexity of *D2O* is:

$$|F(\text{UOBS})| \cdot \Omega_{max} + (B_{max})^{|\text{COMPS}|} \cdot (C_F + O_{max}) \quad (7)$$

It is reasonable to assume that  $\Omega_{max} \ll C_D$  since the complexity of finding a diagnosis is usually much larger than just enumerating them.

To provide a deeper understanding of the above complexities, let us assume that the *DIAGNOSE* call in *O2D* corresponds to a brute force search for diagnoses, and for each it performs a consistency check, whose cost is  $C_C$ , to check if the set of behavior mode assignments is a diagnosis. Thus,  $C_D = (B_{max})^{|\text{COMPS}|} \cdot C_C$ .

Similarly, let us assume that the *FINDOBS* call in *D2O* corresponds to a brute force search for all feasible observations, and for each it performs a consistency check, costs  $C_C$  as well. Thus,  $C_F = |F(\text{UOBS})| \cdot C_C$ .

So, we have that the complexity of *O2D* is

$$|F(\text{UOBS})| \cdot (B_{\max})^{|\text{COMPS}|} \cdot C_C \quad (8)$$

and the complexity of *D2O* is

$$|F(\text{UOBS})| \cdot \Omega_{\max} + (B_{\max})^{|\text{COMPS}|} \cdot (|F(\text{UOBS})| \cdot C_C + O_{\max}) \quad (9)$$

It is reasonable to assume that  $\Omega_{\max} \ll (B_{\max})^{|\text{COMPS}|}$ , since the diagnoses consistent with a single observation are a subset of all possible behavior mode assignments. Similarly,  $O_{\max} \ll |F(\text{UOBS})|$  since the observations consistent with a single behavior mode assignment are a subset of all feasible observations. Thus, the main factor in the complexity of *D2O* is

$$|F(\text{UOBS})| \cdot (B_{\max})^{|\text{COMPS}|} \cdot C_C \quad (10)$$

Therefore, *D2O* and *O2D* are equally complex.

**Analysis for Abductive Diagnosis:** For abductive diagnosis,  $C_F = |\text{COMPS}|$  since we can propagate values from inputs to outputs to obtain the observation corresponding to a given diagnosis. For the same reason,  $O_{\max} = 1$ . So, the complexity of *D2O* becomes:

$$|F(\text{UOBS})| \cdot \Omega_{\max} + (B_{\max})^{|\text{COMPS}|} \cdot |\text{COMPS}| \quad (11)$$

A brute-force implementation of DIAGNOSE is to iterate over all possible behavior mode assignment, and for each to check if it is consistent with the given observation. Since  $C_C = |\text{COMPS}|$  in abductive diagnosis, we have that in this case the complexity of *O2D* is

$$|F(\text{UOBS})| \cdot (B_{\max})^{|\text{COMPS}|} \cdot |\text{COMPS}| \quad (12)$$

It is reasonable to assume that  $\Omega_{\max} \ll (B_{\max})^{|\text{COMPS}|}$ . Therefore, for this brute-force implementation of DIAGNOSE the *O2D* approach is a factor of  $|F(\text{UOBS})|$  slower than *D2O* in this setting.

## 6 Finding the Most Probable Diagnosis

The number of diagnoses can be exponential in the number of components. Even the number of minimal-cardinality diagnoses may be prohibitively large. As such, finding all diagnoses may be too time consuming. A reasonable alternative is to return the *most probable* diagnosis. A naive approach is to find all diagnoses, sort them according to their likelihood, and return the most likely one. This, however, will also be very time consuming. Algorithm 3 is an extension of *O2D*, called *most likely diagnosis (MstLikeDiag)*, that returns the most probable diagnoses without finding all diagnoses.

There are two differences between *O2D* and *MstLikeDiag*. First, it maintains an additional variable  $P$  to store the *probability mass* of the feasible observations processed so far. That is, the sum of  $\text{UOBS}(\text{OBS})$  over every feasible observation  $\text{OBS}$  for which we have computed diagnoses.  $P$  is initially zero (line 1 in Alg. 3) and updated after processing each observation (line 12).

The second difference between *O2D* and *MstLikeDiag* is that it returns only the most likely diagnosis (lines 14

---

### Algorithm 3: Pseudo-code *MstLikeDiag*.

---

**Input:** UOBS, an uncertain observation  
**Input:** SD, the system description  
**Input:** COMPS, the system components  
**Output:**  $\omega_1$ , the most likely diagnosis

```

1  $\Omega \leftarrow \emptyset$ ;  $p \leftarrow$  empty dictionary;  $P \leftarrow 0$ 
2 for  $\text{OBS} \in F(\text{UOBS})$  do
3    $(\Omega_{\text{OBS}}, p_{\text{OBS}}) \leftarrow \text{DIAGNOSE}(\text{SD}, \text{COMPS}, \text{OBS})$ 
4   for  $\omega \in \Omega_{\text{OBS}}$  do
5     if  $\omega \in \Omega$  then
6        $p(\omega) \leftarrow p(\omega) + \text{UOBS}(\text{OBS}) \cdot p_{\text{OBS}}(\omega)$ 
7     else
8        $p(\omega) \leftarrow \text{UOBS}(\text{OBS}) \cdot p_{\text{OBS}}(\omega)$ 
9        $\Omega \leftarrow \Omega \cup \{\omega\}$ 
10   $\omega_1 \leftarrow \text{argmax}_{\omega \in \Omega} p(\omega)$ 
11   $\omega_2 \leftarrow \text{argmax}_{\omega \in \Omega \setminus \{\omega_1\}} p(\omega)$ 
12   $P = P + \text{UOBS}(\text{OBS})$ 
13  if  $p(\omega_1) - p(\omega_2) \geq 1 - P$  and  $p(\omega_1) \geq 1 - P$  then
14    return  $\omega_1$ 
15 return  $\text{argmax}_{\omega \in \Omega} p(\omega)$ 
```

---

and 15). To do so, after processing every feasible observation, *MstLikeDiag* finds the diagnoses with the highest and seconds highest  $p$  values, denoted by  $\omega_1$  and  $\omega_2$ , respectively. Finally, *MstLikeDiag* halts and returns  $\omega_1$  if the difference between  $\omega_1$  and  $\omega_2$  is greater than or equal to  $1 - P$  and  $p(\omega_1) \geq 1 - P$  (line 14) or if all observations have been processed (line 15).

For every diagnosis  $\omega$ , the most an observation  $\text{OBS}$  can add to  $p(\omega)$  is  $\text{UOBS}(\text{OBS})$  (see lines 6 and 9). Thus, if  $p(\omega_1)$  is larger than  $p(\omega_2)$  by more than  $1 - P$ , then the diagnosis likelihood of  $\omega_1$  is larger than the diagnosis likelihood of any other  $\omega \in \Omega$ . Since we require also that  $p(\omega_1) \geq 1 - P$ , it also proves that any diagnosis not already in  $\Omega$  will have a lower likelihood than  $\omega_1$ .

Observe that for abductive diagnosis, every diagnosis has exactly one observation that it is consistent with. Therefore, in every iteration of Algorithm 3, for every diagnosis  $\omega \in \Omega$  it holds that  $p(\omega)$  is indeed the diagnosis likelihood of  $\omega$  w.r.t UOBS. Therefore, there is no need to compare  $\omega_1$  to  $\omega_2$ , and only the right-hand condition in line 13 needs to be checked.

## 7 Evaluation

### 7.1 Abductive Form

To evaluate our algorithms for the abductive diagnosis form, we used a system from the 74XXX benchmark Boolean circuit - an ALU called 74182, having  $|O| = R = 5$  outputs,  $\text{SYSIN} = 9$  inputs and  $\text{COMPS} = 18$  components. An abductive diagnosis states which gates are healthy and which are in a flipped mode. All observations were selected from Feldman et al.'s (2010) known benchmark, a total of 250 observations, equally divided to five minimal cardinality values of 1-5.

To adapt the observations to have uncertainty, observed

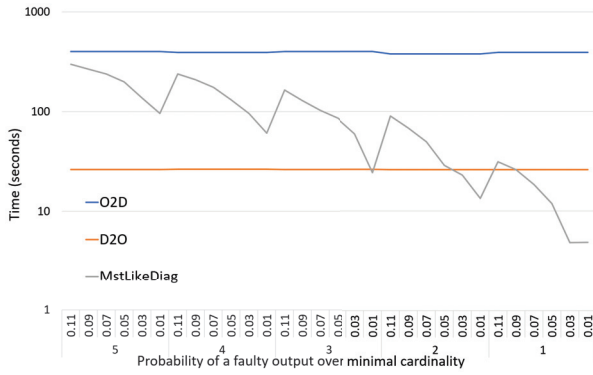


Figure 1: Average runtime over  $\mathcal{P}$  and cardinality.

outputs from the benchmark were treated as candidate outputs for each observation. Then, we set a probability  $\mathcal{P}$  for an observed output to be wrong.

For each algorithm, *O2D* (Alg. 1), *D2O* (Alg. 2) and *MstLikeDiag* (Alg. 3), observation and value of  $\mathcal{P}$ , we collected the average execution runtime. Figure 1 visualizes these results. The runtime in the  $y$ -axis is presented in logarithmic scale to show the relative differences between the algorithms. The  $x$ -axis combines 2 parameters: the value of  $\mathcal{P}$  (up) and the minimal cardinality over the most probable diagnoses (down).

As can be seen, *O2D*'s runtime is constant (i.e. independent of the minimal cardinality diagnosis and of the value of  $\mathcal{P}$ ) and runs in an average time of 392 seconds. *D2O*'s runtime is also constant. It runs in an average time of 26 seconds, performing better than *O2D* by multiple orders of magnitude. *MstLikeDiag* is the most interesting one. As we can see, its runtime is dependent on both parameters. For each cardinality the runtime decreases as  $\mathcal{P}$  decreases. In addition, for each value of  $\mathcal{P}$  the runtime decreases as the cardinality decreases.

These results support our hypothesis. First, we can see that *D2O* performs much better than *O2D* and produces the same results, regardless of the cardinality of the diagnoses nor  $\mathcal{P}$ . This is because *O2D* calls `DIAGNOSE`  $2^R$  times, each with runtime of  $2^{|\text{COMPS}|}$ , resulting in total runtime of  $2^{R+|\text{COMPS}|}$ , while *D2O* iterates only the components, resulting in total runtime of  $2^{|\text{COMPS}|}$ . Second, we can see that for low values of cardinality and  $\mathcal{P}$ , *MstLikeDiag* outperforms *D2O*.

These results show that retrieving the diagnoses by traversing on all subsets of components from *COMPS* is always better than using a naïve diagnoser for each possible outputs combination. Also, it shows that for small values of  $\mathcal{P}$  or for observations with low cardinality diagnoses, retrieving only the maximum probability diagnoses can save time while ensuring correctness. Since it is safe to assume the probability for a faulty output is small and most real observations imply a single fault, we can derive that this method is generally better.

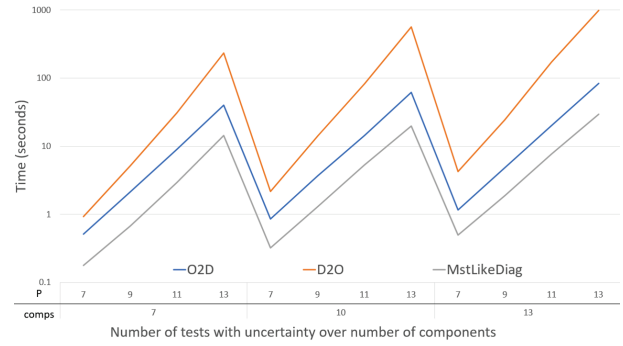


Figure 2: Algorithms average runtime over the number of components and the number of tests with uncertainty.

## 7.2 Consistent Form

To evaluate our algorithms for the consistency-based diagnosis form, we run experiments on the software diagnosis domain. Here, a diagnosis corresponds to the software components (e.g. classes, methods, blocks or lines of code) that are faulty (i.e. contain a bug), and the observation is given by the result of tests. This domain is considered consistency-based since a test can pass even though a buggy component is involved in it.

We have used Barinel (Abreu, Zoetewij, and van Gemund 2009) algorithm, which combines MBD and Spectrum-based Fault Localization (SFL). Barinel also assigns a score to every diagnosis, that roughly corresponds to its likelihood. Using Barinel as our diagnoser, we were able to implement all three algorithms.

Here we define an uncertain output as a chance for a bug in the test, causing the test to fail or pass though it should do the opposite. A bug in the test is not so rare and can happen from multiple reasons, such as (1) attempting to write a test for a legacy, barely understandable code, (2) requirements had been changed and the test was forgotten to be changed accordingly, and (3) the test was written by a less experienced developer.

Each execution trace vector is stored in a matrix file, containing: (1) all components names, (2) the components involved in each test, and (3) all tests outcomes. Since there are many independent variables affecting the algorithms' runtime in this domain, we constructed synthetic trace matrices to control as much of them as we can and evaluate their effect on the runtime. In addition, we run experiments on real buggy source codes.

To do this, we created a combination of  $n$  components and  $m$  tests, where  $7 \leq n, m \leq 13$ . For each combination we created 20 random matrices and error vectors having the combination properties. For each matrix file with  $m$  tests, we executed all three algorithms with a fixed number of  $u$  tests with uncertain observation over the result of the test, where  $7 \leq u \leq m$ . In addition, the third algorithm was executed with different faulty output probabilities ( $[0.1, 0.5]$ ). Preliminary results showed that the number of failing tests has insignificant influence on runtime, so we have not varied this parameter.

**Synthetic Experiments:** Figure 2 visualizes the effects of both the number of components and the number of tests with uncertainty ( $x$ -axis). The  $y$ -axis is the average runtime in logarithmic scale. Both parameters are in direct correlation with all three algorithms’ runtime. If we fix the number of components (tests with uncertainty), the average runtime of all algorithms becomes higher as the number of tests with uncertainty (components) becomes larger. We can also see that the number of tests with uncertainty have much more influence on the runtime than the number of components.

We evaluated also the effect of  $\mathcal{P}$  on *MstLikeDiag*’s runtime. for  $\mathcal{P}$  values of 0.1,0.2,0.3,0.4,0.5, the average runtime was 0.43,1.13,2.6,4.36,5.71, respectively. The results suggest a linear correlation between them.

A significant difference with the abductive case is the performance ratio between *O2D* and *D2O*. In this case, *O2D* seems to perform better than *D2O* by an order of magnitude. This can be explained by diving into the algorithms implementation in this domain.

In *O2D* (Algorithm 1), The DIAGNOSE function, which is Barinel in our case, is being called as the number of feasible observations (line 3), which is  $2^u$  (with  $u$  being the number of tests with uncertainty). Inside Barinel implementation, a minimal-hitting-set (MHS) algorithm is called to find sets of components hitting all failing tests. being an NP-complete problem, the asymptotic runtime of this implementation to MHS is exponential. However, using some optimizations, the real runtime of this implementation is usually almost linear in the number of components ( $n$ ). Being the dominant part of *O2D*, the runtime of this algorithm is close to the sum of runtimes of all Barinel calls, which is  $n * 2^u$ .

As for *D2O* (Algorithm 2), The FINDOBS function is being called  $2^n$  times (line 5). Here we have used a somewhat ”reverse MHS” algorithm - given a set of components  $c$ , find all sets of tests whose  $c$  is a MHS of them. The runtime of this implementation is exponential in  $u$ . Thus, the total runtime of *D2O* is close to  $2^{n+u}$ , an order of magnitude worse than *O2D*.

**Real-World Experiments:** Next, we evaluated our algorithms using real bugs from Apache Commons-Lang, an open-source software project. This project includes a test package. For every experiment, we chose a package and a known bug that occurred in this package. Then, we chose a subset of the automated tests that test this package and ran them to record their trace. The trace and error vector were then stored as a matrix file, allowing us to follow the same pipeline as before, with a minor difference. Here, as opposed to with the synthetic matrices, we assumed that only initially observed failed tests can have uncertainty, while passed tests are certain. This assumption addresses the high complexity of real-world problems which includes hundreds of components and tests. We believe this is a fair assumption, as experience shows that a bug in the test code usually causes it to fail, not pass when it should not.

Figure 3 presents the runtime of all three algorithms. We can see that *O2D* performs as good as *MstLikeDiag*. We can also clearly see that *D2O* performs significantly worse than the other two. This is due to the change discussed above,

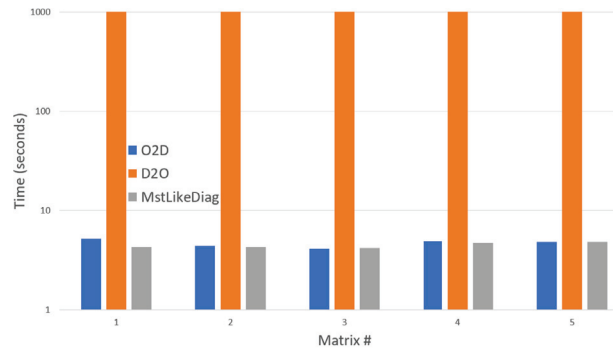


Figure 3: Algorithms runtime on several matrices.

greatly reducing the number of tests with uncertainty but still keeping a large number of components. Knowing that only *D2O*’s runtime is exponential in the number of components explains why it performs poorly, relative to the other two.

As for the outputs, *MstLikeDiag* returned a different most probable diagnosis for  $\mathcal{P} = 0.1$  and for  $\mathcal{P} > 0.1$ , for every evaluated matrix. This shows that the level of uncertainty indeed affects the algorithms output.

The results of both the real and synthetic matrices show us that if only the most probable diagnosis is desired, using a method to find it explicitly is always the best option, performance-wise, no matter the nature of software being diagnosed. If, however, all diagnoses are required, then iterating all feasible observations is probably the best choice.

## 8 Discussion

One may consider solving the UO-MBD problem using a reduction to (certain) MBD. We demonstrate the reduction in the Boolean circuits domain, but a similar reduction can be made in any domain: The reduction will create a new system s.t. for each output  $o \in O$ , a Buffer gate  $b_o$  is placed s.t.  $in(b_o) = in(o)$ ,  $out(b_o) = o$ . We define the a-priori faulty likelihood for this buffer as the faulty probability of its attached output. Informally speaking, we separate the uncertainty of an output from the output itself, and turning it into a buffer component which can be faulty. This way we get a new system where all outputs are certain, and we can use any off-the-shelf standard diagnoser.

Unfortunately, this solution does not quite solve our problem. A diagnosis in the new system is a union of components from the initial system and buffers created by the reduction algorithm. In the consistent form, since a diagnosis can explain several observations, different diagnoses can constitute the same diagnosis after filtering the buffers. As adding new components might cause the likelihood computing (e.g. Barinel) to act differently, a future work is needed to research this area.

Even in the abductive form its efficiency is doubtful. Using a brute-force diagnoser for a Boolean circuit domain for example, *O2D*’s main factor is executing this diagnoser for each feasible observation, resulting in a runtime of  $2^{|COMPS|+|O|}$ . Using the reduction, we can execute the diagnoser one time only but the system now has  $|COMPS| + |O|$ ,

resulting also in a runtime of  $2^{|COMPS|+|O|}$ . *D2O*'s runtime, however, is  $2^{|COMPS|}$ , performing better than the reduction.

## 9 Conclusion and Future Work

In this paper, we addressed an extension of the classic MBD problem, where there is an uncertainty over the observed values. We proposed two algorithms for solving this problem: (1) *O2D* - for each possible observation compute the consistent diagnoses, and (2) *D2O* - for each possible diagnosis compute the observation(s). We proved their correctness and analyzed their runtime. In addition, we focused on a third algorithm, *MstLikeDiag*, that provides the most probable diagnosis.

Followed experiments were conducted to compare the algorithms runtime in abductive and consistent forms. We concluded that finding the most probable diagnosis has the best performance in both forms, assuming the probability for a faulty observed output or the diagnosis's cardinality are likely to be small. In the abductive form, we concluded that it is always preferable to find all diagnoses by *D2O* than *O2D*. In the consistent form however we concluded the opposite, as iterating all observations always gave us better results, runtime-wise.

## Acknowledgement

We would like to thank Roni Stern. This research was funded by ISF grant No. 1716/17.

## References

- Abreu, R.; Zoetewij, P.; and van Gemund, A. J. C. 2009. Spectrum-based multiple fault localization. In *Automated Software Engineering (ASE)*, 88–99. IEEE.
- Cardoso, N.; Abreu, R.; Feldman, A.; and Kleer, J. d. 2016. A framework for automatic debugging of functional and degradation failures. In *Proceedings of the Twenty-second European Conference on Artificial Intelligence*, 569–576. IOS Press.
- Console, L., and Torasso, P. 1991. A spectrum of logical definitions of model-based diagnosis 1. *Computational intelligence* 7(3):133–141.
- Console, L.; Dupré, D. T.; and Torasso, P. 1989. A theory of diagnosis for incomplete causal models. In *IJCAI*, 1311–1317.
- Console, L.; Dupré, D. T.; and Torasso, P. 1991. On the relationship between abduction and deduction. *Journal of Logic and Computation* 1(5):661–690.
- de Kleer, J., and Williams, B. C. 1987. Diagnosing multiple faults. *Artif. Intell.* 32(1):97–130.
- Elmishali, A.; Stern, R.; and Kalech, M. 2019. Debuguer: A tool for bug prediction and diagnosis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 9446–9451.
- Feldman, A.; Provan, G.; and Van Gemund, A. 2010. Approximate model-based diagnosis using greedy stochastic search. *Journal of Artificial Intelligence Research* 38:371–413.
- Flesch, I.; Lucas, P. J.; and van der Weide, T. P. 2007. Conflict-based diagnosis: Adding uncertainty to model-based diagnosis. In *IJCAI*, volume 2007, 380–385.
- Lamperti, G., and Zanella, M. 2000. Uncertain temporal observations in diagnosis. In *ECAI*, 151–155.
- Lamperti, G., and Zanella, M. 2010. Monitoring of active systems with stratified uncertain observations. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* 41(2):356–369.
- Lucas, P. J. 2001. Bayesian model-based diagnosis. *International Journal of Approximate Reasoning* 27(2):99–119.
- Metodi, A.; Stern, R.; Kalech, M.; and Codish, M. 2014. A novel sat-based approach to model based diagnosis. *Journal of Artificial Intelligence Research* 51:377–411.
- Narasimhan, S., and Biswas, G. 2007. Model-based diagnosis of hybrid systems. *IEEE Transactions on systems, man, and cybernetics-Part A: Systems and humans* 37(3):348–361.
- Pencolé, Y., and Cordier, M.-O. 2005. A formal framework for the decentralised diagnosis of large scale discrete event systems and its application to telecommunication networks. *Artificial Intelligence* 164(1-2):121–170.
- Reiter, R. 1987. A theory of diagnosis from first principles. *Artif. Intell.* 32(1):57–95.
- Stern, R.; Kalech, M.; Feldman, A.; and Provan, G. M. 2012. Exploring the duality in conflict-directed model-based diagnosis. In *AAAI*.
- Stern, R.; Kalech, M.; Rogov, S.; and Feldman, A. 2017. How many diagnoses do we need? *Artificial Intelligence*.
- Struss, P., and Price, C. 2003. Model-based systems in the automotive industry. *AI magazine* 24(4):17–34.