Structural Abstraction for Strong Fault Models Diagnosis

Orel Elimelech, Roni Stern and Meir Kalech
The Department of Information System Engineering, Ben Gurion University of the Negev

Abstract
Cones are a special form of structural abstraction that were shown to be very effective in speeding up model-based diagnosis (MBD) algorithms. Past work, either explicitly or implicitly, assumed a weak fault-model (WFM), where only the proper behavior of the system is specified. In many cases, some information is known about how faulty components behave, and thus a more informed strong fault-model (SFM) exists. Several challenges arise when applying, and we propose several methods to address them. Each method is analyzed and evaluated. Empirical results show that cones can be extremely helpful to diagnosis algorithms when handled using the methods we proposed.

1 Introduction
Automated diagnosis is a well-established field in Artificial Intelligence, in which the goal is to automatically diagnose system failures, i.e., to identify what caused a system to fail. Model-based diagnosis (MBD) is a popular approach to automated diagnosis in which diagnoses are inferred from the relation between the observed behavior of the system and a model of the systems expected behavior. MBD algorithms have been successfully applied in the aerospace industry [Williams and Nayak, 1996], the automotive industry [Niggemann et al., 2010], and even to identify software bugs [Abreu et al., 2011; Zamir et al., 2014].

In general, the computational complexity of MBD algorithms is exponential in the number of modeled components. As a result, many MBD algorithms do not scale well to large systems. The hierarchical diagnosis paradigm [Mozetic, 1991; Siddiqi and Huang, 2011] provides a possible solution to this scaling problem. Hierarchical diagnosis starts by diagnosing an abstraction of the diagnosed system in which groups of components are considered as a single abstract component, which is treated as a black box. Diagnosing this abstract system is expected to be easier than diagnosing the original system as it has less components than the original system. The diagnoses found for this abstract system, referred to as “abstract diagnoses”, are used to produce diagnoses of the original system by recursively diagnosing abstract components that are assumed faulty by the abstract diagnosis. Generating diagnoses from abstract diagnoses is called grounding.

The success of hierarchical diagnosis algorithms depends on the system abstraction used. One form of structural abstraction that was successfully applied in state-of-the-art hierarchical diagnosis algorithms is based on partitioning the system into cones [Siddiqi and Huang, 2011; Metodi et al., 2014]. A cone is a subsystem of the diagnosed system that has a single output. For example, in the system depicted in Figure 1 the components $A$ and $B$ form a cone $G$. The corresponding cone-based abstract system consists of components $C$, $D$, $E$, $F$, and $G$.

To diagnose the abstract system, one needs to define the behavior of the abstract components in it. Past work on cone-based abstractions defined the normal behavior of an abstract component by the behavior of its constituent components and, either explicitly or implicitly, did not model how faulty abstract components behave. This is suitable for system models in which no restrictions are made on the behavior of faulty components. Such system models are said to have a weak-fault model (WFM). In many real systems, however, some information is known about the behavior of faulty components. Such system models are said to have a strong fault-model (SFM), and the focus of this work is in applying a cone-based hierarchical diagnosis to such systems.

To illustrate the problem caused by not restricting the behavior of faulty abstract components when the underlying system has a SFM, consider again the system in Figure 1. The components $A$, $B$, $E$ and $F$ are pipes, whose normal behavior is input = output. Components $C$ and $D$ are closed valves,
and therefore their expected output is “0”, where “0” marks no flow of fluids and “+” marks a flow of fluids. Note that a faulty closed valve may leak if fluids are passing through it (i.e., if its input value is “+”), but a faulty valve cannot output fluid if none was passed through it (i.e., if its input value is “0”). Assume that the observed input and output values were \( in_1 = in_3 = out_1 = out_2 = + \) and \( in_2 = 0 \). This indicates that some fault has occurred, as the expected output for these input values is \( out_1 = out_2 = 0 \). Without restricting the behavior of faulty abstract components, a possible abstract diagnosis is to assume that \( G \) outputs “+” instead of the expected zero. It is easy to see that such a diagnosis cannot be grounded, since for \( G \) to output “+” since it gets as input “0”, i.e., no fluid, and pipes cannot generate fluid if they do not get it as input (even if they are faulty).

As noted by Torta and Torasso [2013], previous cone-based hierarchical diagnosis algorithm did not specify what do in such a case. In this paper, we propose and evaluate several techniques to handle the possibility of ungroundable abstract diagnoses. Ungroundable abstract diagnoses may also occur in system models that specify “physical impossibility”, which state how a faulty component cannot behave [Friedrich et al., 1990]. Our methods also apply for such system models.

We propose three methods for handling ungroundable abstract diagnoses: pessimistic, weak-optimistic, and strong-optimistic. In the pessimistic method, the abstract system is analyzed offline and only cones that can be grounded for any observation are used. This results in a weaker abstraction with more components than the original cone abstraction, and is thus harder to diagnose. In the strong optimistic method all cones are used and abstract diagnosis that cannot be grounded are discarded. The weak optimistic method serves as a middle ground: all cones are initially used, but cones are “broken” when an abstract diagnosis fails to be grounded. Breaking a cone means replacing it with its constituent components. The resulting abstract system contains more components and is thus harder to diagnose, but diagnosing it is more likely to result in abstract diagnoses that can be grounded.

The tradeoffs between these approaches are discussed analytically and compared them experimentally. The results clearly show that using each of these approaches is substantially better than not using cones at all. As expected, there is no clear winner between the proposed approaches, suggesting that hybrid approaches are needed. However, the optimistic approach seems to be better for larger systems.

2 Problem Definition

An MBD problem is defined by a tuple \((SD, COMPS, OBS)\), where \(SD\) is a model of the diagnosed system, \(COMPS\) is the set of system components, and \(OBS\) is the observed behavior of the system (e.g., the observed inputs and outputs of the system). The system model \(SD\) specifies the normal behavior of every component \(c \in COMPS\). We say that a component \(c \in COMPS\) has a strong fault-model if \(SD\) contains some information about how \(c\) behaves when faulty. Otherwise we say that \(c\) has a weak fault-model. In this work we address \(SDs\) that have components with a strong fault-model.

Every component \(c \in COMPS\) has a state (also known as a “behavior mode”) specifying how \(c\) behaves. For example, the valves in Figure 1 may be in a “leaking”, “clogged”, or “normal” state, where “leaking” indicates that fluid passing through will leak, “clogged” indicates that no fluid can pass though it, and “normal” indicates its nominal behavior. A state assignment is an assumption about the state of one or more components, and a system state is a state assignment over all the components in the system. For a state assignment \(\omega\), let \(\text{state}_\omega(c)\) denote the state assumed for component \(c\) by \(\omega\). We denote by \(\omega(+)\) and \(\omega(-)\) the set of components assumed to be in a healthy state and in a faulty state, respectively. A diagnosis problem (DP) arises if the system state in which all components are healthy is inconsistent with \(SD\) and \(OBS\). A solution to a DP is a diagnosis.

Definition 1 (Diagnosis). A system state \(\omega\) is called a diagnosis if \(\omega \land OBS \land SD\) is consistent.

A DP can have many solutions (=diagnoses). Some diagnosis algorithms (DAs) are designed to find a single diagnosis. Other DAs are complete, in the sense that they return all possible diagnoses.\(^1\) Since the entire set of possible diagnoses may be very large, other DAs return only the most likely diagnoses [Williams and Ragno, 2007]. Under some assumptions one can argue that the most likely diagnoses are those that assume the smallest number of faults. Such diagnoses are called minimal cardinality (MC) diagnoses. Many DAs are specifically tuned to find a single or all MC diagnoses [Siddiqi and Huang, 2011; Metodi et al., 2014]. In this paper, we show how our approach which uses cones to diagnose systems with a SFM can find a single MC diagnosis, all MC diagnosis, and all possible diagnoses.

3 Hierarchical Diagnoses and Cones

The type of hierarchical DAs we consider works as follows. First, an abstract version of the original DP is formulated. We use the terms abstract components to refer to the components of the abstract DP (ADP). Then, a (possibly off-the-shelf) DA is used to solve the ADP, finding abstract diagnoses. Finally, these abstract diagnoses are grounded to concrete diagnoses of the original DP. Next, we define one type of ADP that has been used by previous hierarchical DAs.

3.1 Cone Abstraction

Definition 2 (Dominator and Cone). A component \(c\) is said to be dominated by a component \(d\) if every path in the system topology from \(c\) to a system output passes through \(d\). A cone is the set of components that are all dominated by the same component. That component is called the dominator of that cone. A maximal cone is a cone that is not contained by any other cone.

\(^1\)Explicitly finding all diagnoses is not necessary to be complete, as the set of all diagnoses can be represented more compactly by the set of all kernel diagnoses [De Kleer et al., 1990].

\(^2\)We use the term system output broadly, including any observable of the system, e.g., probes.
We define that a component dominates itself, and thus every cone contains its dominator, and every single component is also a cone. Such single component cones are called trivial cones. A cone partition is a disjoint partition $A = \{C_1, \ldots, C_k\}$ of COMPS such that every $C \in A$ is a cone. For every component $c \in$ COMPS, we denote by $A(c)$ the cone $C \in A$ such that $c \in C$. A maximal cone partition is a cone partition $A$ in which every $C \in A$ is a maximal cone. Finding a maximal cone partition can be done efficiently [Metodi et al., 2014].

**Definition 3** (Constrastion for WFM). A WFM cone abstraction for a DP $(SD, COMPS, OBS)$ induced by a cone partition $A$ is an MBD problem $(SD_A, A, OBS)$, whose components are the cones in $A$ and where $SD_A$ defines the healthy behavior of each cone by the behavior of its constituent components.

Intuitively, the cone abstraction DP is the same DP as the original DP, except that each cone in the cone partition is treated as a black box. Recent successful hierarchical DAs [Siddiqi and Huang, 2011; Metodi et al., 2014] use the cone abstraction induced by the maximal cone partition as the ADP they solve.

4 Cones in Systems with a SFM

In Definition 3, $SD_A$ only describes the healthy behavior of abstract components. As discussed in the introduction, this is reasonable for weak fault models, since $SD$ also do not impose any restriction on the behavior of faulty components, but raises a design question in case of systems with strong fault model. We propose the following simple approach to define $SD_A$ for SFM: trivial cones will have the same states and behavior as their underlying component, while non-trivial cones will be assumed to have a weak fault model in $SD_A$. As shown in Figure 1 and discussed in the introduction, this WFM definition of $SD_A$ may result in some abstract diagnoses being ungroundable, as they would assume an output that is not consistent with any system state of the underlying DP. Finding such abstract diagnoses is wasteful. However, finding an abstract diagnosis that can be grounded may save an exponential amount of time. The challenge is to predict which cones to use and which cones to “break” to their constituent components, so that the resulting cone abstraction has a small number of abstract components to enable fast diagnosis but the resulting abstract diagnoses are likely to be groundable. We propose several methods to handle this.

4.1 The Pessimistic Method

The first method is pessimistic, keeping only cones that are guaranteed to be groundable for every possible observation. To achieve this, the system model $(SD)$ and its corresponding maximal cone partition $A$ are analyzed in a preprocessing stage as follows. Let $in(C)$ and $out(C)$ denote the inputs and output of a cone $C \in A$, respectively, and let $D(in(C))$ and $D(out(C))$ be the domains of the inputs and output of $C$. For example, in a Boolean circuit, a cone with 5 inputs would have a domain $[0, 1]^5$.

**Definition 4** (Always-Groundable (AG) Cones). A cone $C$ is always-groundable (AG) iff for every input $i \in D(in(C))$ and output $o \in D(out(C))$ values there exists a state assignment $\omega_C$ for the components in $C$ such that the following is consistent: $SD \land (in(C) = i) \land (out(C) = o) \land \omega_C$.

If all cones are AG, then all abstract diagnoses are groundable. One way to check if a cone is AG is to enumerate all values in $D(in(C))$ and $D(out(C))$ and search for consistent state assignments. Such an exhaustive approach is exponential in the number of input and output values or in the number of components in $C$ (the smaller of the two). We propose a more efficient “lossy” approach to detect AG cones, in which a cone $C$ is identified as AG if the trivial cone containing only the dominator of $C$ is AG. Checking if a trivial cone is AG in an exhaustive manner can be done very fast, under the reasonable assumption that a single component has a relatively small number of inputs and outputs. For example, checking if a trivial cone $C = \{c\}$ is AG in a Boolean circuit means simply checking if $c$ has either the “flip” state, or both states “stuck-at-1” and “stuck-at-0”, since all inputs, internal variables, and outputs are Boolean.

4.2 Optimistic Methods

## Algorithm 1: Hierarchical Diagnosis for SFM with Cones

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Diagnose(System description SD, Components COMPS, Observation OBS)</td>
</tr>
<tr>
<td>2</td>
<td>$A \leftarrow$ maximal cone partition of SD</td>
</tr>
<tr>
<td>3</td>
<td>/* Pessimistic keeps only AG cones */</td>
</tr>
<tr>
<td>4</td>
<td><strong>if</strong> Pessimistic <strong>then</strong></td>
</tr>
<tr>
<td>5</td>
<td><strong>foreach</strong> non-trivial cone $C \in A$ <strong>do</strong></td>
</tr>
<tr>
<td>6</td>
<td><strong>if</strong> $C$ is not identified as AG <strong>then</strong></td>
</tr>
<tr>
<td>7</td>
<td>Break($C$)</td>
</tr>
<tr>
<td>8</td>
<td>$\omega_A \leftarrow$ Diagnose($SD_A, A, OBS$)</td>
</tr>
<tr>
<td>9</td>
<td><strong>if</strong> $\omega_A$ is not null <strong>then</strong></td>
</tr>
<tr>
<td>10</td>
<td>/* Try to ground $\omega_A$ */</td>
</tr>
<tr>
<td>11</td>
<td>Ungroundable $\leftarrow \emptyset$</td>
</tr>
<tr>
<td>12</td>
<td><strong>foreach</strong> non-trivial cone $C$ in $\omega_A$ <strong>do</strong></td>
</tr>
<tr>
<td>13</td>
<td><strong>if</strong> $C$ cannot be grounded <strong>then</strong></td>
</tr>
<tr>
<td>14</td>
<td>Ungroundable $\leftarrow C$</td>
</tr>
<tr>
<td>15</td>
<td>Goto line 16</td>
</tr>
<tr>
<td>16</td>
<td><strong>if</strong> Ungroundable is empty <strong>then</strong></td>
</tr>
<tr>
<td>17</td>
<td>Add all possible groundings of $\omega_A$ to $\Omega$</td>
</tr>
<tr>
<td>18</td>
<td><strong>else</strong></td>
</tr>
<tr>
<td>19</td>
<td><strong>if</strong> Weak Optimistic <strong>then</strong></td>
</tr>
<tr>
<td>20</td>
<td>Break(Ungroundable)</td>
</tr>
<tr>
<td>21</td>
<td>$\Omega_A \leftarrow \emptyset$</td>
</tr>
<tr>
<td>22</td>
<td><strong>else</strong></td>
</tr>
<tr>
<td>23</td>
<td>/* No more abstract diagnoses */</td>
</tr>
<tr>
<td>24</td>
<td>return $\Omega$</td>
</tr>
</tbody>
</table>

$^3$The states “flip”, “stuck-at-1”, and “stuck-at-0”, correspond to faulty states in which the components’ output is the negation of the normal output, always True, and always False, respectively.
The next two methods we propose are optimistic, keeping all the maximal cones and taking the risk of finding abstract diagnoses that are not groundable. The strong-optimistic method simply discards ungroundable abstract diagnosis and continues to search for different abstract diagnoses, in hope that some of them will be groundable. This method was inspired by the DA of Feldman et al. [2009], in which a SFM system was diagnosed by first assuming that it has a WFM and then validating the found diagnoses with the underlying SFM. The weak-optimistic method also discards abstract diagnoses that could not be grounded, but also modifies the used cone abstraction to raise the chance that future abstract diagnoses will be groundable. This is done by choosing one of the cones that could not be grounded and replacing it with its constituent components, i.e., breaking that cone. Abstract diagnoses of the modified cone abstraction are expected to be easier to ground, having one less cone to ground. On the other hand, the modified cone abstraction has more components, and thus it is potentially harder to diagnose compared to the original cone abstraction. Note that both weak and strong optimistic approaches consider cones that are not AG. Cones that are not AG may still be groundable for a specific observation and abstract diagnosis.

Algorithm 1 lists the pseudo-code for all three proposed methods. Initially, the maximal cone partition is computed (line 2). Importantly, some of the cones may not be AG. If the pessimistic method is used, every cone that is not identified as AG is broken (line 5). Then, a DA finds a new abstract diagnosis ω_A (line 9), and attempts to ground it (line 13). If ω_A can be grounded, then all possible groundings of ω_A are added to the list of diagnoses Ω. Grounding is done by applying a DA recursively on the faulty abstract components. If ω_A cannot be grounded, there is an abstract component C ∈ ω_A that cannot be grounded. If the strong optimistic method is used, then the process continues by searching for a new abstract diagnosis (line 9). If the weak optimistic method is used then the ungrounded cone C is broken (line 19). When no new abstract diagnoses exist, all diagnoses have been found.

4.3 Soundness and Completeness

In the following theoretical analysis we assume that the DAs used to find abstract diagnoses (line 9) and to ground them (line 17) are sound and complete. We show that this results in all three cone-handling methods (pessimistic, weak- and strong-optimistic) being sound and complete.

Let M be an MBD problem and M_A be the cone abstraction of M induced by A. For every state assignment ω for M we define a unique state assignment ω_A for M_A in which the state for Every abstract component C ∈ A is set as follows:

\[
 state_{ω_A}(C) = \begin{cases} 
 state_{ω}(c) & \text{if } C = \{c\} \\
 Healthy & \text{if } C \cap ω_A(\neg) = \emptyset \\
 Fauly & \text{if } C \cap ω_A(\neg) \neq \emptyset 
\end{cases}
\]

where “Healthy” and “Faulty” are the assumption that C is healthy and faulty, respectively.

**Lemma 1.** For any cone partition A, if ω is a diagnosis for M then ω_A is an abstract diagnosis for M_A.

Lemma 1 directly follows from the fact that non-trivial abstract components have a WFM in SD_A and are thus consistent with any assumption about their output. It is easy to see that ω is a grounding of ω_A. Thus, every diagnosis is a grounding of an abstract diagnosis.

**Theorem 1** (Soundness and Completeness). Algorithm 1 is sound and complete (for all cone-handling methods), i.e., every state assignment returned is a diagnosis, and all diagnoses will be returned.

**Proof outline:** Since the DAs used for finding abstract diagnoses and for grounding are sound, it trivially holds that every state assignment returned is a diagnosis. Assume by negation that there exists a diagnosis ω that is not returned by Algorithm 1. The different cone-handling methods use cone abstractions induced by potentially different cone partitions: the pessimistic only uses AG cones, the strong-optimistic only uses maximal cones, and the weak-optimistic only uses maximal cones it did not break. Regardless of the cone abstraction used, there exists an abstract diagnosis ω_A for which ω is a grounding of (Lemma 1). This means that either ω_A was not found as an abstract diagnosis or that ω was not found as a grounding of ω_A. Both options contradict the assumption that the DAs used to find abstract diagnoses and to ground them are complete. □

5 MC-Diagnosis with Cones in SFM

In many DPs there is a very large number of diagnoses and thus finding all diagnoses is too time consuming and not needed [Stern et al., 2015]. A common approach is to only return MC diagnoses, i.e., diagnoses that assume the least amount of faulty components. We adapt Algorithm 1 to only return MC diagnoses by making two main modifications. First, the DA used to find abstract diagnosis is modified to only return abstract diagnoses that are MC diagnoses of the cone abstraction being used. There are many off-the-shelf DAs that only return MC diagnoses [Siddiqi and Huang, 2007; Metodi et al., 2014]. Second, the DA used to ground the abstract diagnoses is modified to only return cardinality-preserving grounding, which are defined next.

**Definition 5** (Cardinality-preserving grounding). A diagnosis ω is a cardinality-preserving grounding of an abstract diagnosis ω_A if they have the same cardinality, i.e., |ω_A(\neg)| = |ω(\neg)|.

A diagnosis ω can only be a cardinality-preserving grounding if it assumes that exactly one component is faulty in every abstract component assumed faulty in ω_A. This greatly simplifies the task of grounding an abstract component.

**Theorem 2** (MC Soundness). If the DA used to find abstract diagnoses returns an MC diagnosis of cone abstraction being used and the DA used for grounding returns a cardinality-preserving grounding then Algorithm 1 returns an MC diagnosis.

**Proof outline:** By negation, let ω' be a diagnosis that is smaller than ω, the diagnosis returned by Algorithm 1 (i.e., |ω'(\neg)| < |ω(\neg)|). Let ω_A and ω'_A be the abstract diagnoses that ω and ω' are a grounding of, respectively (Lemma 1). By
construction of $\omega'_A$, we have that $|\omega'_A(-)| \leq |\omega(-)|$. On the other hand, since $\omega$ is a cardinality-preserving grounding, then $|\omega(-)| = |\omega(-)|$. Thus, $|\omega'_A(-)| < |\omega(-)|$, contradicting that $\omega'_A(-)$ is an MC diagnosis of the cone abstraction.

To preserve completeness of Algorithm 1, additional modifications are required in some of the cone-handling methods. For the pessimistic method, this is done by only using cones that are always-groundable with a single fault (AG-SF). A cone $C$ is AG-SF if for every possible input and output to $C$ there is a state assignment that is consistent with these input and output values and assumes at most one component is faulty. Identifying which cones are AG-SF can be done in a similar "lossy" manner proposed for identifying AG cones (see Section 4.1): a cone $C$ is identified as an AG-SF if the trivial cone that is composed of $C$’s dominator, is AG-SF.

Unlike the pessimistic method, when the optimistic methods are used it may be the case that an MC diagnosis for $M$ is not a cardinality preserving grounding of an MC diagnosis for $M_A$. For example, the MC diagnosis of $M$ may require assuming two components in a maximal cone are faulty. This is not a problem for the weak-optimistic method. When it does not find a cardinality-preserving grounding for an MC diagnosis for $M_A$, it breaks one of the cones it could not ground (with a single fault). However, the strong-optimistic in such a case would not find this MC diagnosis. Fortunately, such cases have the following unique property.

Lemma 2. For a cone abstraction $M_A$, if there exists an MC diagnosis for $M$ that is not a cardinality-preserving grounding of an MC diagnosis for $M_A$ then there is no MC diagnoses for $M_A$ that has a cardinality-preserving grounding.

Proof outline: By negation, let $\omega_A$ be an MC diagnosis of $M_A$ that has a cardinality-preserving grounding $\omega$. Let $K$ be the cardinality of $\omega_A$ and $\omega$ (which is the same for both because $\omega$ is a cardinality-preserving grounding). Let $\omega^*$ be the MC diagnosis for $M$ that is not a cardinality-preserving grounding of an MC diagnosis for $M_A$, and let $\omega'_A$ be the abstract diagnosis mapped to $\omega^*$. Since $\omega^*$ is an MC diagnosis, its cardinality is at most $K$. As $\omega'_A$ is not a cardinality-preserving grounding of $\omega_A$, it means that the cardinality of $\omega'_A$ must be smaller than $K$. This contradicts $\omega_A$ being an MC diagnosis of $M_A$. \qed

A direct corollary from Lemma 2 is that if a cardinality-preserving grounding was found for an MC diagnosis for $M_A$, then all MC diagnoses for $M$ are cardinality-preserving groundings of a MC diagnosis for $M_A$. Thus, we modify the strong-optimistic method such that if all abstract diagnoses of $M_A$ were generated and no cardinality-preserving grounding was found, then one of the cones in $A$ is broken, and the process restarts with the new cone abstraction. The choice of which cone to break is arbitrary. In our experiments we chose heuristically to break the cones that could not be grounded in the most abstract diagnoses.

Theorem 3 (MC Completeness). For all three modified cone-handling methods, Algorithm 1 finds all MC-diagnosis.

Proof outline: The theorem trivially holds for the pessimistic method, since every abstract diagnoses of the cone abstraction it uses has a cardinality preserving grounding. The weak-optimistic method breaks a cone in cases where no cardinality-preserving grounding is found for an MC diagnosis of $M_A$. Thus, it will eventually arrive to a cone abstraction in which a cardinality-preserving grounding exists. Following Lemma 2, this means that all MC diagnoses will be found. For the strong-optimistic method, if a single abstract diagnosis has a cardinality-preserving grounding then all MC diagnoses will be found by Algorithm 1 (Lemma 2). If this is not the case, then after trying to ground all MC diagnoses of $M_A$ the strong-optimistic method will break one of the cones and restart the search. Eventually, a cone abstraction will be reached in which a cardinality-preserving grounding will be found. This known because in the worst case all cones will be broken and only trivial cones will remain, which do not require to be grounded. \qed

6 Experimental Results

We experimentally evaluated the three proposed methods and a baseline method (denoted as "no cones") in which no cones are used. As a DA, we used SATbD [Metodi et al., 2014], a state-of-the-art SAT-based DA, which we adapted to consider SFM. To the best of our knowledge, there is no standard benchmark for evaluating SFM. Therefore, we generated such a benchmark from a well-known WFM benchmark, namely, the ISCAS-85 benchmark on Boolean circuits, as follows. Initially, all components are assumed to have two states: healthy and one of the faulty states (in Boolean circuits, these are flip, stuck-at-1, and, stuck-at-0), chosen randomly. Then for every component we added each of the missing faulty states with probability $p$, where $p$ is a parameter. Setting $p = 1$ results in every component having all possible faulty states. In such a case, every component has a “flip” state and thus every cone is AG-SF. Therefore, all our methods should act the same for $p = 1$. We experimented with $p = 0.025, 0.5, 0.75, and 1$, resulting in every component having on average 1, 1.5, 2, 2.5, and 3 faulty states. Note that while our experiments are on Boolean circuits, all our exposition above applies to any qualitative model.

We experimented on three systems from the ISCAS-85 suite: c880, c1355, and c2650, having 383, 546, and 1193 components. For each system, ten random observations per cardinality was chosen from the observation set of Feldman et al. [Feldman et al., 2010]. Since this observation set was generated for WFM, some of the observations were not solvable and were thus discarded. Column “Obs.” in Table 1 lists the number of solvable observations per value of $p$. The “MC” column shows the size of the largest MC diagnosis, and the “AG-SF” column shows the average number of non-trivial AG-SF cones.

For every solvable observation we run the competing methods with a timeout of 20 seconds and measured the ratio of solvable observations not solved by each method. The results are shown in the columns “No”, “Pes.”, “Strong”, and “Weak” of Table 1, corresponding to not using cones, the pessimistic method, strong optimistic, and weak optimistic, respectively. The best results for every system and parameter $p$ are marked in bold. Note that the preprocessing time required by the pessimistic method is included in the allocated
Table 1: Experimental Results.

20 seconds, but was negligible (less than 10ms) in practice. The results show several trends. First, there is a huge advantage in using cones over not using cones. For example, in the c2670, not using cones almost always results in a time-out, while an MC diagnosis was always found with the pessimistic method. Second, the weak-optimistic method is in general the best alternative, loosing to the pessimistic method only for c2670 with p = 0. This follows our theoretical analysis, where the added cost of potentially failing to ground an abstract diagnoses is negligible compared to the added gain of considering larger cones. The weaker performance of the strong optimistic method can be explained by its behavior on cases where MC diagnoses are not a cardinality-preserving groundings of the initial maximal cone abstraction. In such cases, the strong-optimistic will go over all abstract diagnoses of this cone abstraction before breaking one of the cones and finding an MC diagnosis. As the number of abstract diagnoses can be very large, this severely limits the performance of the strong optimistic method.

While these results suggest that the weak-optimistic is the best option, this is highly dependent on the specific instance, and it is not hard to construct examples where each approach would be superior to the others. Future work would study the relation between specific features of an observation and which approach would work best.

7 Related Work

Mozetic’s foundational work on hierarchical MBD [Mozetic, 1991] has been extended by several researchers. Provan [2001] proposes an abstraction that considers domain knowledge in order to reduce the complexity. Chittaro and Ranon [2004] extend Mozetic’s hierarchical abstraction and improve its efficiency by considering the observation. They propose a way to rearrange a given hierarchy by considering the observation points and also propose a bottom-up algorithm that exploits the observations to reduce the cost of the diagnoses search. These previous works do not propose algorithms to generate an abstraction. However, the efficiency of the abstraction depends on the choice of the abstraction. Perrot and Travé-Massuyes [2007] propose a conflict-directed approach to choose the abstraction. This approach is based on the insight of [Williams and Rago, 2007] that a diagnosis cannot contain a conflict and thus finding conflicts reduces the search space of the diagnoses. In a similar way, an abstraction directed by minimal conflicts may potentially focus on the diagnoses. This can be achieved even with no observation using abstract states corresponding to minimal potential conflicts [Cordier et al., 2004].

Another common form of abstraction is used in physical systems, where continuous state variables are often abstracted to discrete values. The challenge is to determine the granularity level of these discretized component states, as there is a tradeoff between the accuracy of the model, and thus of the diagnosis task, and the complexity of the diagnosis. Determining the abstraction level is a hard problem.

Tortasso and Torta [2005] propose an automatic behavioral abstraction approach for time-varying system. This approach is composed of online and offline phases. At offline a set of reusable abstraction fragments is computed and at online the process builds a system abstraction by combining suitable abstraction fragments. Sachenbacher and Struss [2005] implemented a task-dependent qualitative domain abstraction in the context of a real-world example taken from the automotive domain. They present a framework that determines qualitative values for a device model.

Recently, Torta and Torasso [2013] proposed a unified view of structural and behavioral abstraction. They proposed several methods to reduce the number of behaviors of an abstract component, merging possible states if they are indiscernible. The complexity of identifying such cases can be very high and in the worst case no states may be merged, ending up with an exponential number of possible behaviors.

8 Conclusion and Future Work

We presented a general approach and several practical methods for using cone to speedup diagnosis of systems with a SFM. The key challenge is that in SFM, abstract diagnoses may not be grounded to diagnoses of the original DP. Three methods are proposed to address this problem: a pessimistic method, in which only some of the cones are used such that all abstract diagnoses could be grounded, and two optimistic methods, in which all maximal cones are used and different mechanisms are used for handling ungrounded abstract diagnoses. We showed how to tune all these methods to find only MC diagnoses, and evaluated the performance of these methods on a modified version of the ISCAS-85 benchmark. Results show that all the proposed methods are substantially better than not using cones. In general, the (weak) optimistic approach was better, but in a small number of cases the pessimistic approach was better. This suggests future work that combines these approaches intelligently. Moreover, we did not consider a priori knowledge about which components are more likely to fail. Considering this information in the decision of which cones to consider and which cones to break is also a promising future direction.
References


