Hierarchical Diagnosis in Strong Fault Models

Roni Stern\(^1\) and Meir Kalech\(^1\) and Orel Elimelech
\(^1\)Ben Gurion University of the Negev, Be’er Sheva, Israel

e-mail: roni.stern@gmail.com, kalech@bgu.ac.il, orel.elim@gmail.com

Abstract

Cones are a special form of structural abstraction that were shown to be very effective in speeding up model-based diagnosis algorithms. This is achieved by first finding diagnoses to the abstract system, where every component is a cone, and then grounding that abstract diagnosis to be a diagnosis for the original system. Past work, either explicitly or implicitly, assume a weak fault-model (WFM), which is a system model that includes some information about how faulty component behave. Abstract diagnoses in an SFM may not be groundable. We proposed two approaches to address this problem. A pessimistic approach, where cones are broken if they may lead to un-groundable abstract diagnoses, and an optimistic approach where all cones are initially considered, and are only broken if under the current observation they lead to abstract diagnoses that are not groundable. We also show how to tune these approaches to the diagnosis problem of finding minimal cardinality diagnoses. Empirical evaluation on a modified ISCAS-85 benchmark show that cones are useful, but that there is not universal winner between the optimistic and pessimistic approaches, suggesting further research is needed.

1 Introduction

Given an abnormal behavior of a system, model-based diagnosis (MBD) infers which components caused the abnormal behavior by reasoning over the relation between the observed behavior and a model of the diagnosed system. In general, the computational complexity of MBD algorithm is exponential in the number of modeled components. As a result, many MBD algorithm do not scale to real-sized systems.

Mozetic [1] introduced the hierarchical MBD paradigm to improve the scalability of MBD algorithms. In hierarchical MBD, a structural abstraction of the diagnosed system is assumed, that is, an abstract system is given where a single component in the abstract system is mapped to a set of components in the original system. As this abstract system has less components than the original system, it is expected to be easier to diagnose. An underlying MBD algorithm runs first on the abstract system. Diagnoses of the abstract system are then grounded to produce diagnoses of the original system. Grounding of an abstract diagnosis is usually done by recursively applying the underlying MBD algorithm.

One form of structural abstraction that was successfully applied in state-of-the-art diagnosis algorithm is based on partitioning the system into cones [2, 3]. A cone is a subsystem of the diagnosed system that has a single output. Identifying cones is easy computationally based only on the topology of he system, and grounding cones can often be done in a straightforward way.

In this paper we investigate the use of cone abstractions in hierarchical diagnosis, when the diagnosed system has a strong fault model (SFM). A system model is said to have a strong fault model if it contains information about the abnormal behavior of its components. By contrast, system model with a weak fault model (WFM) only specifies the normal behavior of its components.

The main difference between SFM and WFM with respect to using cones is that in systems that are modeled with a WFM, abstract diagnoses could always be grounded. This is not the case if the diagnosed system is modeled with a SFM, where the faulty behavior of components is constrained to follow the SFM. For example, consider the system depicted 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 ex-

Figure 1: The expected output is \(\text{out}_1 = \text{out}_2 = 0\). Components A and B form a cone G, which is a part of an abstract diagnosis \(\{G\}\), where G is assumed to be faulty, outputting “+” instead of the expected zero. This diagnosis cannot be grounded, as A and B cannot generate fluids according to their behavior modes.

\[ \text{in}_1 = + \]
\[ \text{out}_1 = + \]
\[ \text{in}_2 = 0 \]
\[ \text{out}_2 = + \]
\[ \text{in}_3 = + \]
\[ \text{C} \]
\[ \text{A} \]
\[ \text{B} \]
\[ \text{D} \]
\[ \text{E} \]
\[ \text{F} \]
\[ \text{G} \]
pected behavior is output=0, where zero marks no flow of fluids and "+" marks a flow of fluids. The observation is $i_{in1} = i_{in2} = o_{out1} = o_{out2} = +$ and $i_{in2} = 0$, which is abnormal as we expected $o_{out1} = o_{out2} = 0$. The components $A$ and $B$ form a cone $G$. The cone abstraction of this system is composed of the components $C, D, E, F$, and $G$.

A corresponding abstract diagnosis is that $G$ is faulty, outputting "+" instead of the expected zero. In WFM, this cone is easily grounded, by assuming that the dominator component (which is $B$ in this case), is faulty and outputs "+".

Now, assume a strong fault model, where pipes have two modes: ok and blocked, which corresponds to a pipe that cannot pass fluids, i.e., output=0. In such a case the abstract diagnosis $[G]$ cannot be grounded, as $A$ and $B$ cannot generate fluids according to their behavior modes.

One way to resolve this issue is to compute behavior modes for each cone by compiling the behavior modes of its constituent components, resulting in a correct component mapping [4]. This, however, may result in a cone with an exponential number of behavior modes (the cross product over all the behavior modes of its components), and a resulting cone abstraction that is not easier to diagnose than the original system. In this work we explore a different approach, where no assumption is made on the abnormal behavior of cones that are composed of more than a single component.

We propose three alternative approaches for handling the problem of abstract diagnoses that cannot be grounded. A pessimistic approach performs an offline analysis of the cones in the system, and only uses cones that grounding them would be possible. This results in a more detailed abstraction, which is harder to diagnose.

A second approach we propose is a weak optimistic approach, where all cones are considered, but cones are "broken" dynamically online when an abstract diagnosis fails to be grounded. This allows using more cones and thus finding abstract diagnoses is simpler, but risks the possibility of finding ungroundable abstract diagnoses. A third approach we propose, referred to as a strong optimistic approach, also consider all cones, but "breaks" cones only if no abstract diagnoses was groundable.

The tradeoffs between these approaches are discussed, and we compare experimentally algorithm instances of the pessimistic and weak optimistic approaches. 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.

2 Problem Definition

The input to our problem is the standard MBD input: $(SD, COMPS, OBS)$, 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.

In some cases, $SD$ only specifies the nominal behavior of every component in COMPS. This type of models are called weak fault-models (WFM). Note that in WFM, there is no assumption as to how abnormal components behave. Models that also specify how abnormal components behave are called strong fault-models (SFM). $SD$ in SFM defines for every component $C \in COMPS$ a set of behaviors modes. A behavior mode of a component $C$ specifies a state of $C$, where that state impacts how $C$ behaves. For example, a valve in a physical system may have a behavior mode “leaking”, which indicates that some of the fluid passing through it will leak out. For every component, at least one of its modes must represent its nominal behavior.

This mode is denoted by $ok$. Note that WFM can be viewed as a SFM where every component has two modes, $ok$ and ¬$ok$, where the impact of ¬$ok$ on the system’s behavior is not specified.

A mode assignment is an assignment of modes to components, such that every component is assigned to one of its modes. For a mode assignment $\omega$, we denote by $\omega(+) \land \omega(-)$ the set of components assigned the $ok$ mode, and denote by $\omega(-)$ its complement (i.e., $\omega(-) = COMPS \setminus \omega(+)$).

$OBS$ specifies the values of the observed system inputs and outputs. The statement $mode(C) = m$ represents the assumption that the mode of $C$ was $m$ when the system was observed.

A diagnosis problem arises if assuming that the mode of every component is $ok$ is inconsistent with $SD$ and $OBS$

$$SD \land OBS\land \bigwedge_{C \in COMPS} mode(C) = ok \text{ is not consistent}$$

A solution to a diagnosis problem is a diagnosis.

Definition 1 (Diagnosis). A mode assignment $\omega$ is called a diagnosis if $\omega \land OBS \land SD$ is consistent.

We address the specific diagnosis problem of finding only minimal cardinality diagnoses (or simple MC diagnoses). MC diagnoses are diagnoses that assume the smallest number of faulty components, i.e., the smallest number of components that are assigned to modes that are not $ok$. Beyond theoretical interest, MC diagnoses are a simplistic representation of most likely diagnoses. The task of finding MC diagnoses has thus attracted much interest in the MBD literature throughout the years [5; 6; 7; 3].

3 Cones and Abstract Diagnoses

In general, finding MC diagnoses is NP-Hard [8]. Various forms of system abstractions and hierarchical diagnoses were introduced to improve the scalability of diagnosis algorithm [1; 9; 2; 4]. One of the most successfully applied system abstractions for model-based diagnosis is based on the concepts of cones and abstract diagnoses [2].

Definition 2 (Dominator and Cones). 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.

We assume that a component dominates itself, and thus every cone contains its dominator and every single component is a cone with a single component. Grouping components into maximal cones that contain them results in a disjoint partition of all the system components. Siddiqi and Huang [2] proposed an abstraction of the diagnosed system where every maximal cone is considered as a black box.

Definition 3 (Cone Abstraction). For a given system description $SD$, a cone abstraction $A_{SD}$ is a system description whose components are the dominators of all the maximal cones in $SD$.
For every component $C$ in $A_{SD}$, we denote by $cone(C) \subseteq COMPS$ the cone dominated by $C$ in $SD$. Diagnoses of $A_{SD}$ are called abstract diagnoses and diagnoses of $SD$ are called grounded diagnoses. For a given abstract diagnosis $\omega_A$, we use the term healthy cone and faulty cone to denote a cone that its dominator belongs to $\omega_A^+(\cdot)$ and $\omega_A^-(\cdot)$, respectively.

Diagnosis algorithms that use cone abstractions find grounded diagnoses by finding abstract diagnoses and inferring grounded diagnoses from them. To infer the grounded diagnosis from a given abstract diagnosis $\omega_A$, we first compute the possible inputs and outputs of each cone, assuming that $\omega_A$ is true. Then we recursively run a diagnosis algorithm to find grounded diagnoses for each faulty cone. This entire process is more efficient than directly searching for grounded diagnoses, because finding abstract diagnoses is expected to be faster than finding grounded diagnoses, as there are less components in $A_{SD}$ than in $SD$.

3.1 Finding MC Diagnoses

The above hierarchical approach to diagnoses can be specifically tuned to find MC diagnoses as follows. First, search for MC abstract diagnoses, i.e., MC diagnoses for the cone abstraction. Then, infer grounded diagnoses from these abstract diagnoses as described above, but limit the internal cone diagnoses to only consider single fault diagnoses.

The above approach is valid for WFM, since the output of a faulty cone can be achieved by setting the dominator component to be faulty and all other components in that cone to be healthy. Since in WFM a faulty component is unconstrained, and can output any value, then any faulty cone can be grounded by a faulty dominator. Previous work used this knowledge to efficiently find grounded diagnoses [2; 3].

In SLM, however, the faulty behavior of components is specified. Thus, it is possible that the dominator may not have suitable behavior mode to ground the cone. This can have two negative outcomes. First, the grounded diagnoses that can be inferred from a given abstract diagnoses may all require a higher cardinality than that of the abstract diagnosis. Second, an abstract diagnosis may not have any grounded diagnoses that can be inferred from it. Such an example was shown in Figure 1, where the abstract diagnosis $\{G\}$ could not be grounded, due to the behavior modes of the pipes $A$ and $B$. Below, we discuss how cones can be used in strong fault models.

4 Cones in Strong Fault-Models

Consider now a system description $SD$ with a SLM, and a cone abstraction $A_{SD}$ derived for it. Let $\omega_A$ be an abstract diagnosis found for $A_{SD}$. The grounded diagnoses inferred from it assign $ok$ to components in $\omega_A^+(\cdot)$ and abnormal modes to the components in $\omega_A^-(\cdot)$. An abstract diagnosis is said to be groundable if there exists at least one grounded diagnosis that can be inferred from it.

Finding an abstract diagnosis that is not groundable is wasteful. However, finding an abstract diagnosis that can be grounded may save an exponential amount of time. As discussed earlier, in WFM every abstract diagnosis is groundable, while this is not the case for SLM. The challenge is then to predict which cones to use and which cones to “break” to its constituent components, so that the resulting abstract diagnoses would be groundable and the resulting cone abstraction is as small as possible (to enable fast diagnosis).

4.1 Pessimistic Cones

The first approach is pessimistic, keeping only cones so that the resulting cone abstraction is guaranteed to have only abstract diagnoses that are groundable. To achieve this, the system model ($SD$) and the cones in it are analyzed in a pre-processing stage as follows. Let $in(C)$ and $out(C)$ denote the inputs and output of a cone $C$, respectively, and let $D(in(C))$ and $D(out(C))$ be the domains of the inputs and output of a cone $C$, respectively. For example, in a Boolean circuit, the domains are either zero or one. Lastly, let $SD_C$ be the part in $SD$ that describes the behavior of the components in $C$.

**Definition 4 (Correct Cones).** A cone $C$ is called correct if for every input value $i \in D(in(C))$ and output value $o \in D(out(C))$ there exist a mode assignment $\omega_C$ for the components in $C$ such that the following term is consistent

$$SD_C \land (in(C) = i) \land (out(C) = o) \land \omega_C$$

If all cones are correct, then all abstract diagnoses are groundable.

One way to check if a cone is correct is to enumerate all values in $D(in(C))$ and $D(out(C))$ and search for consistent mode assignments. Searching for a consistent mode assignment is in the worst case exponential in the number of components in $SD_C$. Therefore, the worst case runtime of checking if a cone is correct is exponential in the number of inputs (due to checking all values in $D(in(C))$) and the number of components in the cone (due to the search for a consistent mode assignment).

Alternatively, one can use a “lossy” approach to detect correct cones, where only some of the correct cones are detected. For example, consider a Boolean circuit, where the inputs, outputs and internal variables of a cone can only have Boolean values. It is possible to detect some of the correct cones by only considering the behavior modes of the dominator component. If the dominator component has a “flip” behavior mode, i.e., a mode where the component outputs the opposite of its input, then its cone is correct. This does not mean, of course, that cones with a dominator that does not have a “flip” mode is not correct.

For example, consider the Boolean circuit given in Figure 2. Components $A$ and $B$ are buffer components, i.e., where the normal behavior is input-output, and components $C$ and $D$ are AND gates. Components $A$ and $B$ for a cone denoted by $E$. The possible input/output pairs for cone $E$ are $0/0, 0/1, 1/0$ and $1/1$. If $B$ has a “flipâŽ† mode,
then $E$ could ground each of these input/output pairs: always assume $A$ is healthy, and set $B$ to flip or not according to the desired output.

Now, assume that that $B$ has only a “stuck-at-one” mode but $A$ has a “flip” mode. Cone $E$ is still correct, but it would not be identified as such by only considering the modes of the dominator. Thus, detecting correct cones in such a way (according to the modes of the dominator), is fast but may miss some correct cones.

### 4.2 Optimistic Approach

The second approach we consider is optimistic, keeping all the cones and taking the risk of finding abstract diagnoses that are not groundable. To use the optimistic approach, we slightly modify the hierarchical MBD paradigm to handle cases where abstract diagnoses cannot be grounded.

If an abstract diagnosis cannot be grounded, this means that at least one cone was found to be not true. Thus, one approach is to break that cone into its constituent components, and then search again for abstract diagnoses. We call this approach the weak optimistic approach.

Alternatively, one can continue to search for other abstract diagnoses without breaking the incorrect cones, in hope that one of these abstract diagnoses would be groundable. We call this approach the strong optimistic approach.

Note that even though a cone is not correct, it is still possible to find an abstract diagnosis that can be grounded. This can happen for example if in that abstract diagnosis the incorrect cone is assumed to output its nominal behavior.

### 5 MC-Diagnosis with Cones in SFM

All the above hold for finding a diagnosis, regardless of its cardinality. If the task is to find MC diagnoses, one can further tune the above approaches as follows. The high-level logic we propose is to consider first only diagnoses that involve a single faulty component per cone.

#### 5.1 Pessimistic

In the pessimistic approach for finding MC diagnoses, we use a cone abstraction that contains only cones that can be grounded by assuming a single fault.

**Definition 5** (Single Fault Correct Cones). *A cone $C$ is said to be single-fault correct if for every possible input to $C$ there is a mode assignment for the components in $C$ for outputting every possible output value, and that mode assignment consists of at most one faulty component.*

The pessimistic approach for finding MC diagnoses considers a cone abstraction with only cones that are single-fault correct. Then, every minimal cardinality abstract diagnosis can be grounded to a grounded diagnosis of exactly the same cardinality, and that cardinality is indeed minimal for both $SD$ and $A_{SD}$.

#### 5.2 Optimistic

We start with the weak optimistic approach for finding MC diagnoses. First, an MC abstract diagnosis is found. Then, every faulty cone is tested, to see if it can be grounded by assuming that only one if its constituent components is faulty. If all faulty cones can be grounded using only a single faulty component per cone, then a grounded MC diagnosis is found. Otherwise, the search for abstract MC diagnoses is restarted on the original cone abstraction except of that faulty cone which is broken to its constituent components. This is repeated until either all cones are broken or a groundable abstract MC diagnosis is found.

The strong optimistic approach is slightly more elaborated. If an abstract MC diagnosis cannot be grounded using a single faulty component, then the search continues for other abstract MC diagnoses (without breaking any cone). This continues until either a groundable MC diagnoses is found with a single fault per component, or until all abstract MC diagnoses are considered. In this case, it is clear that the minimal cardinality for $SD$ is larger than $A_{SD}$. Then, a single cone is selected and broken, and the process restarts with the updated cone abstraction.

### 6 Theoretical Analysis

In this section we analyze theoretically the proposed approaches to use cones in SFM: pessimistic, weak optimistic, and strong optimistic. First, we consider the soundness and completeness of the general approaches, and discuss their specific variants designed for finding MC diagnoses (defined in Section 5). Second we discuss the runtime analysis of each approach.

#### 6.1 Soundness and Completeness

Soundness in the context of consistency-based diagnosis algorithms means that the diagnosis algorithm indeed returns diagnoses, i.e., mode assignments that are consistent with the observation and system description. All of the approaches are sound, as the abstract diagnosis is sound, and grounding it exactly checks if there is a consistent mode assignment.

There are two senses in which a diagnosis algorithm can be complete. It can be complete in the sense that it is guaranteed to return a diagnosis, and it can be complete in the sense that it is guaranteed to return all diagnoses. Next, we discuss completeness in both senses of the proposed approaches.

The completeness of all approaches is derived from the fact that for every diagnosis $\omega$ of $SD$ there exists an abstract diagnosis $\omega_A$ such that $\omega$ is a grounded diagnosis of $\omega_A$.

In the context of finding MC diagnoses, soundness means that the returned diagnoses indeed have minimal cardinality and completeness corresponds to finding all MC diagnoses.

As any other diagnosis, an MC diagnosis is a grounding of a specific abstract diagnosis. If the cone abstraction contains only correct cones, then for every MC diagnosis $\omega$ there is an abstract MC diagnosis $\omega_A$, and therefore the pessimistic approach is complete. If the cone abstraction contains cones that are not correct, then it may be the case that an MC diagnosis would be a grounding of a non-MC diagnosis. This occurs only if all MC diagnoses are not groundable. Thus, both weak and strong optimistic approaches for finding MC diagnoses are also complete.

#### 6.2 Runtime Analysis

The runtime of all the approaches is composed of three phases: 1) a preprocessing phase which is run only based on the system description, before observing the faulty behavior of the system, 2) a diagnosis phase, where abstract diagnoses are found, and 3) a grounding phase, where abstract diagnoses are grounded.

Consider first the complexity of the preprocessing phase. All approaches find a cone abstraction using a simple polynomial cone detection algorithm [3]. The pessimistic approach requires additional runtime for this phase to detect...
which cones are correct (or single-fault correct when searching for an MC diagnosis). As discussed above, checking if a cone is correct is in the worst case exponential in the number of components in the cone and the number of inputs in the cone. Note that the runtime of the preprocessing phase can be amortized over many observations.

Next, consider the runtime of the diagnosis phase. Since some cones may not be correct, then the cone abstraction used by the pessimistic approach is potentially larger than that used by the optimistic approaches. Thus, finding a single abstract diagnoses may be exponentially harder for the pessimistic approach. However, the optimistic approaches will spend time finding abstract diagnoses that are not groundable. Thus, which approach will require an overall longer diagnosis phase depends on how many cones are not correct and how likely it is to find an abstract diagnosis that is not groundable.

The difference between the runtime of the weak and strong optimistic approaches can be analyzed in a similar way. The weak optimistic approach breaks cones if ungroundable diagnoses are found, so the runtime of its diagnosis phase grows potentially exponentially larger as more cones are broken. By contrast, the strong optimistic approach always tries to diagnose the smallest cone abstraction (compared to the weak optimistic and pessimistic approaches), but may spend much more time in trying to grounding abstract diagnoses that cannot be grounded.

Since the optimistic approach contains more cones than the pessimistic approach, its cones may also be larger, resulting in a higher runtime for grounding it. However, in general, the complexity of grounding a single abstract diagnosis is similar for both approaches.

### 7 Experimental Results

In this section we provide a preliminary experimental evaluation of the proposed approaches. As a benchmark, we used the standard ISCAS-85 benchmark on Boolean circuits. Note that all our exposition above was not specific for Boolean circuits and can apply to any qualitative model.

We generated a strong fault model for the ISCAS-85 benchmark by adding all possible modes to every component. The possible modes in a Boolean circuit is to either flip the expected output, to always output one, or to always output zero. These modes are denoted flip, stuck-at-one, and stuck-at-zero. Then, we randomly removed some of the behavior modes from each component, but always keeping every component with at least one normal and one abnormal mode. Thus, an experiment with 0% of the modes correspond to an experiment where every component has only ok and exactly one of the modes flip, stuck-at-one, or stuck-at-zero. An experiment with 100% of the modes corresponds to an experiment where every component has all three modes. In such a case, every cone is trivially single-fault correct, and thus the pessimistic and optimistic approaches should act the same. We experimented with 0%, 25%, 50%, 75%, and 100% modes.

The task we evaluated is finding the first MC diagnosis as fast as possible. A timeout of 20 seconds was set. We experimented on three systems from the ISCAS-85 suite: c880, c1355, and c2650, having 383, 546, and 1193 components, respectively. We used the observation set of Feldman et al. [10] and chose randomly a set of 10 observations per cardinality for every system. Note that since this observation set was generated for weak fault models, some of the observations were not solvable and were thus discarded. The number of observations used for each system and mode percentage is given in the “# Obs.” column Table 7. The asterisks (*) signs in the “# Obs.” column mark cases where more observation may be solvable if the timeout limit was larger. Table 7 also shows the number of components in every system and the size of the larger minimal cardinality diagnosis (column “Max MC”) for every system and mode percentage.

The diagnosis algorithm we chose to find abstract diagnoses is SATbD, a SAT-based diagnosis algorithm introduced by Metodi et al. [3]. SATbD was shown to be orders of magnitude faster than competing diagnosis algorithms for Boolean circuits models with a weak fault model. We adapted SATbD to consider a strong-fault model, and experimented with two specific instances of the approaches proposed in this work.

As an instance of the pessimistic approach, we considered only cones whose dominator has the “flip” behavior mode. It is easy to see that in Boolean circuits, if a dominator has a “flip” mode then that cone is single-fault correct. The “Correct cones” column in Table 7 shows the number of single-fault correct cones found in this way for the systems and mode percentages we experimented with. As expected, less cones are found as the number of behavior modes decreases.

We also implemented an instance of the weak optimistic approach, which is easier to implement than the strong optimistic approach. As a baseline, we also run SATbD adapted to SFI but without any cones. We denote these three algorithm instances as “opt.”, “pes.” and “no cones”.

Table 2 shows the percentage of observations where an MC diagnosis was not found under the 20 second timeout, for each algorithm, system, and % modes. For every % modes and system, we marked by bold the best performing algorithm, i.e., the algorithm that did not find an MC

<table>
<thead>
<tr>
<th>% Modes</th>
<th># Obs.</th>
<th>Max MC</th>
<th>Correct cones</th>
</tr>
</thead>
<tbody>
<tr>
<td>0%</td>
<td>54</td>
<td>19</td>
<td>13</td>
</tr>
<tr>
<td>25%</td>
<td>98</td>
<td>23</td>
<td>25</td>
</tr>
<tr>
<td>50%</td>
<td>203</td>
<td>24</td>
<td>34</td>
</tr>
<tr>
<td>75%</td>
<td>164</td>
<td>24</td>
<td>40</td>
</tr>
<tr>
<td>100%</td>
<td>203</td>
<td>23</td>
<td>46</td>
</tr>
</tbody>
</table>

Table 1: Shows the # components, # observations, # of correct cones found, and the size of the maximal cardinality of a minimal cardinality diagnosis (column “Max MC”), for the three systems we experimented with.
diagnoses the smallest number of times.

The advantage of using cones over not using cones is very clear from the results. For example, in the c2670, not using cones almost always results in a timeout, while with the pessimistic algorithm an MC diagnosis was always found.

The difference between the pessimistic and optimistic algorithms is less clear. Overall, both algorithms behave similarly in terms of the number of instances solved under the timeout. For the smaller systems, c880 and c1355, the optimistic algorithm is better, while for the larger system c2650 the pessimistic approach is better. Thus, there is no universal winner and deeper study of these algorithms and perhaps a hybrid algorithm that combines them, is needed. In addition, future work would also consider an instance of the strong optimistic approach, as well as other instances of the pessimistic approach, such as an algorithm that exhaustively finds all correct cones.

Among the instances solved under the timeout by both the optimistic and pessimistic algorithms, we observed that the pessimistic algorithm was faster than the optimistic algorithm. We believe that this is due to the overhead required by the optimistic to ground found abstract diagnoses. The pessimistic algorithm can simply ground a found diagnosis by assigning a “flip” mode to the dominator component.

### 8 Related work

Mozetic’s foundational work on hierarchical MBD [1] has been extended by several researchers. Provan [11] proposes an abstraction which considers also the domain knowledge in order to reduce the complexity. Chittaro and Ranon [9] extend the hierarchical abstraction of Mozetic and improves its efficiency by considering the observation. First they propose a way to rearrange a given hierarchy by considering the observation points. Second, they propose a bottom-up algorithm which 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 [12] propose a conflict-directed approach to choose the abstraction. This approach is based on the insight of [6] 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 [13].

A complementary approach to the cone abstraction, and to structural abstraction in general, is to consider abstractions of behavioral models. For example, in physical systems the components may have continuous values and a common approach is to abstract these continuous values to discrete values. The challenge is then to determine the granularity level of the behavior modes of the components. 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.

Torasso and Torta [15] 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 [16] implemented a task-dependent qualitative domain abstraction in the context of a real-world example taken from the automotive domain. They present a framework which automatically determines qualitative values for a device model.

### 9 Conclusion and Future Work

In this work we discussed how cone abstractions can be used to diagnose systems with a strong fault model. In a strong fault model, abstract diagnoses derived from cone abstractions may not correspond to fully grounded diagnosis. We proposed two approaches to address this: a pessimistic approach and an optimistic approach. In the pessimistic approach, cones are broken if they may lead to the creation of an abstract diagnosis that is not groundable, under some possible observation. In the optimistic approach, all cones are initially considered, and are only broken if under the current observation they lead to abstract diagnoses that are not groundable. Tuning the proposed approach to the task of finding MC diagnoses was also discussed.

An empirical evaluation of specific instances of the proposed approach was performed on a modified version of the ISCAS-85 benchmark. The results show that both approaches are substantially better than not using cones. In some systems, one approach was better, in others another approach. This suggests future work that will combine these approaches intelligently.

For example, one can compute offline (i.e., just considering the system description) for every cone that is not correct the ratio of possible inputs and outputs that it could not satisfy. Then, consider a cone abstraction with only the cones above a specific threshold. Alternatively, one can run an optimistic approach, but use this data to decide which abstract diagnosis to try to ground first, or to decide if an ungrounded cone should be broken or not (thus having a middle ground between the strong and weak optimistic approaches).

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.

### Table 2: The % of instances solved under the 20 seconds timeout.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>22</td>
<td>14</td>
<td>62</td>
<td>24</td>
<td>0</td>
<td>81</td>
</tr>
<tr>
<td>0.25</td>
<td>8</td>
<td>20</td>
<td>52</td>
<td>0</td>
<td>0</td>
<td>67</td>
<td>11</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>0.50</td>
<td>0</td>
<td>0</td>
<td>58</td>
<td>0</td>
<td>0</td>
<td>67</td>
<td>0</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>1.00</td>
<td>0</td>
<td>0</td>
<td>59</td>
<td>0</td>
<td>0</td>
<td>56</td>
<td>0</td>
<td>1</td>
<td>100</td>
</tr>
</tbody>
</table>
References


