Model Based Diagnosis of Multiple Observations with Implicit Hitting Sets

Alexey Ignatiev\textsuperscript{1,2}, Antonio Morgado\textsuperscript{1}, and Joao Marques-Silva\textsuperscript{1}

\textsuperscript{1} LASIGE, Faculty of Science, University of Lisbon, Portugal; \{aignatiev,ajmorgado,jpms\}@ciencias.ulisboa.pt
\textsuperscript{2} ISDCT SB RAS, Irkutsk, Russia

Abstract. Model based diagnosis finds a growing range of practical applications, and significant performance-wise improvements have been achieved in recent years. Some of these improvements result from formulating the problem with maximum satisfiability (MaxSAT). Whereas recent work focuses on analyzing failing observations separately, it is also the case that in practical settings there may exist many failing observations. This paper first investigates the drawbacks of analyzing failing observations separately. It then shows that existing solutions do not scale for large systems. Finally, the paper proposes a novel approach for diagnosing systems with many failing observations. The proposed approach is based on implicit hitting sets and so is tightly related with the original seminal work on model based diagnosis. The experimental results demonstrate not only the importance of analyzing multiple observations simultaneously, but also the significance of the implicit hitting set approach.

1 Introduction

The problem of model-based diagnosis \cite{38} (MBD) is ubiquitous in practical settings, ranging from the diagnosis of mechanical and hardware systems, to software programs, to end-user software (e.g. spreadsheets), to knowledge representation systems (e.g. ontologies, etc.), to logic programs, to production systems, to databases, to triple stores, among many others \cite{2, 15, 19, 20, 39, 47}.

The theoretical underpinnings of MBD were developed in the mid 80s \cite{12,38}, and a large body of significant work followed, covering different approaches for MBD \cite{5, 9, 11, 14, 27, 30, 31, 34, 35, 39, 40, 43–46}. In recent years, research has focused on approaches for computing minimum-size diagnoses, with MaxSAT algorithms (and variants thereof) shown to outperform other approaches in representative settings \cite{27, 30, 31}. Another line of work has been on computing (many) subset-minimal diagnoses \cite{34}. However, complete enumeration of all subset-minimal diagnosis is in general infeasible \cite{34}, which explains the interest in computing minimum-size diagnoses.

The usual formulation of MBD is \cite{38}: given a system description, composed of some components, where some of these components can be faulty, and an observation inconsistent with the system description, select a cardinality-minimal (or subset-minimal) set of components which, if declared faulty (i.e. any behavior is allowed for the component), then consistency between the model and the observation is reached. This formulation of MBD is well-suited in settings where the goal is to investigate a single (failing) observation. However, the standard formulation of MBD can be less...
adequate in recent practical instantiations of the problem, where one may need to investigate many failing observations and not only a single one. This is the case with software fault localization [20, 22, 39, 47] and spreadsheet debugging [19], among others. For example, for software fault localization, one may be faced with a few hundred (or even thousands) observations [13]. To our best knowledge, research on diagnosing multiple (failing) observations is scarce, and existing solutions are not only unrealistic in practice, but also technically problematic, as described in this paper.

This paper proposes a novel approach to diagnosing multiple failing observations concurrently, in such a way that the observed drawbacks of alternatively solutions are addressed. The proposed approach builds on recent work on implicit hitting set dualization [8, 10, 21, 23, 24, 32, 41, 45], and is shown to not only perform efficiently in practice, but also to overcome the key issue with problem representation size for large number of observations. Nevertheless, MBD represents a formidable task, and the solution we propose is part of a continued effort for developing effective solutions to aid in diagnosing practical systems.

The paper is organized as follows. Section 2 introduces the notation and definitions used throughout. Section 3 motivates the importance of MBD for multiple observations. The following two sections investigate approaches for MBD given multiple observations. Section 4 outlines existing and also straightforward solutions. In contrast, Section 5 details a novel approach based on implicit hitting set dualization. Section 6 analyzes preliminary results, intended to highlight the benefits of using implicit hitting set dualization. Section 7 concludes the paper.

# 2 Preliminaries

The paper assumes definitions that are standard in Propositional Satisfiability (SAT), Maximum Satisfiability (MaxSAT) [6], and Model-Based Diagnosis (MBD). These are reviewed in this section.

**Boolean Satisfiability & Maximum Satisfiability.** Propositional variables are taken from a set \( X = \{x_1, x_2, \ldots \} \). A Conjunctive Normal Form (CNF) formula is defined as a conjunction of disjunctions of literals, where a literal is a variable or its complement. CNF formulas can also be viewed as sets of sets of literals, and are represented with calligraphic letters, \( A, F, H \), etc. Given a formula \( F \), the set of variables is \( \text{vars}(F) \subseteq X \).

A truth assignment \( \nu \) is a map from variables to \( \{0, 1\} \). Given a truth assignment, a clause is satisfied if at least one of its literals is assigned value 1; otherwise it is falsified. A formula is satisfied if all of its clauses are satisfied; otherwise it is falsified. If there exists no assignment that satisfies a CNF formula \( F \), then \( F \) is referred to as unsatisfiable. (Boolean) Satisfiability (SAT) is the decision problem for propositional formulas, i.e. to decide whether a given propositional formula is satisfiable. Since the paper only considers propositional formulas in CNF, throughout the paper SAT refers to the decision problem for propositional formulas in CNF. Modern SAT solvers instantiate the Conflict-Driven Clause Learning paradigm [6]. For unsatisfiable (or inconsistent) formulas, MUSes (minimal unsatisfiable subsets) represent subset-minimal subformulas that are unsatisfiable (or inconsistent), and MCSes (minimal correction subsets) represent subset-minimal subformulas such that the complement is satisfiable [6].
The (plain) MaxSAT problem is to find a truth assignment that maximizes the number of satisfied clauses. For the plain MaxSAT problem, all clauses are soft, meaning that these may not be satisfied. Variants of the MaxSAT can consider the existence of hard clauses, meaning that these must be satisfied, and also assign weights to the soft clauses, denoting the cost of falsifying the clause; this is referred as the weighted MaxSAT problem, WMaxSAT. When addressing MaxSAT problems with weights, hard clauses are assigned a large weight $\top$. The notation $(c, w)$ will be used to represent a clause $c$ with $w$ denoting the cost of falsifying $c$. The paper considers partial MaxSAT instances, with hard clauses, for which $w = \top$, and soft clauses, for which $w = 1$. The notation $\langle H, S \rangle$ is used to denote partial MaxSAT problems with sets of hard ($H$) and soft ($S$) clauses.

**Model-Based Diagnosis.** The paper considers standard model-based diagnosis (MBD) definitions, following Reiter’s seminal work [38], and which are used in most modern references [27,30,31,34,38,42]. As in recent MBD work, the weak fault model (WFM) is assumed throughout. A system description $SD$ is a set of first-order sentences [38]. The system components, Comps, are a set of constants, $\text{Comps} = \{c_1, \ldots, c_m\}$. Given a system description $SD$, composed of a set of components $\text{Comps}$, each component can be declared as healthy or unhealthy. For each component $c \in \text{Comps}$, $\text{Ab}(c) = 1$ if $c$ is declared as unhealthy (or abnormal); otherwise $\text{Ab}(c) = 0$. Similarly to earlier work [14,27,30,31], it is assumed that $SD$ is represented as a CNF formula, namely:

$$SD \triangleq \bigwedge_{c \in \text{Comps}} (\text{Ab}(c) \lor F_c)$$  \hspace{1cm} (1)

where $F_c$ denotes the CNF encoding of component $c$.

Observations are used to represent situations where the behavior of the system is not the expected one. An observation $\text{Obs}$ is defined as a finite set of first-order sentences [38]. As with the system description, it is assumed that the observation can be encoded into CNF, as a set of unit clauses, and denoted $\text{Obs}$.

**Definition 1 (Diagnosis Problem).** A system with description $SD$ is faulty if it is inconsistent with a given observation $\text{Obs}$ when all components are declared healthy:

$$SD \wedge \text{Obs} \wedge \bigwedge_{c \in \text{Comps}} \neg \text{Ab}(c) \not\models \bot$$  \hspace{1cm} (2)

The problem of diagnosis is to identify a set of components which, if declared unhealthy, make the system consistent with the observation. The problem of MBD is represented by the 3-tuple $\langle SD, \text{Comps}, \text{Obs} \rangle$.

**Definition 2 (Diagnosis).** Given an MBD problem $\langle SD, \text{Comps}, \text{Obs} \rangle$, the set of components $\Delta \subseteq \text{Comps}$ is a diagnosis if

$$SD \wedge \text{Obs} \wedge \bigwedge_{c \in \Delta} \text{Ab}(c) \wedge \bigwedge_{c \in \text{Comps} \setminus \Delta} \neg \text{Ab}(c) \not\models \bot$$  \hspace{1cm} (3)

A diagnosis $\Delta$ is minimal if no proper subset $\Delta' \subseteq \Delta$ is a diagnosis, and $\Delta$ is of minimal cardinality if there exists no other diagnosis $\Delta' \subseteq \text{Comps}$ with $|\Delta'| < |\Delta|$.
In this paper, the dual of a diagnosis will be referred to as an explanation. (These are often referred to as conflicts [38].) It is well-known that a minimal diagnosis is a minimal hitting set of the minimal explanations, and vice-versa [38].

To model MBD with MaxSAT [14, 40], SD (see (1)) represents the hard clauses, whereas the soft clauses are unit clauses ($\neg Ab(c)$), one for each component $c \in $ Comps. Different MaxSAT solving approaches can then be applied. Alternatively, the soft clauses can be replaced by a cardinality constraint and solved iteratively with a SAT solver. Combinational circuits represent the most often used vocabulary in MBD-related research [38]. Figure 1 illustrates an example circuit, example observations, and an often used encoding into CNF [14, 27, 34, 44]. An alternative model, requiring more clauses, has also been studied in recent times [30, 31]. In this paper we follow the original simpler model.

Moreover, although combinational circuits have often been used as the main vehicle to convey research ideas in MBD, other vocabularies can be used. One example is of course clauses, i.e. each component is a clause. In this paper, the key ideas will be conveyed by system descriptions where each component is a clause. Additional examples, using different vocabularies, are also analyzed throughout the paper.

The paper explores implicit (minimal) hitting sets, as applied recently in different settings [8, 10, 21, 23, 24, 32, 45]. For the concrete case of MBD, the paper also relates implicit minimal hitting sets and the duality between minimal diagnoses and minimal explanations [4, 38].

**Relating MBD with MCSes & MUSes.** It is important to highlight that there is a close relationship between diagnoses and MCSes, and between explanations and MUSes [4, 7, 27, 38]. Indeed, given the inconsistent formula (1), a minimal diagnosis $\Delta$ is such that (3) is consistent. Thus, $\Delta$ is an MCS of (1). Similarly, an explanation is a minimal hitting set of the diagnoses, and so it corresponds to an MUS of (1). As a result, enumeration of diagnoses can be obtained by enumeration of MCSes [26], and enumeration of explanations by enumeration of MUSes [24]. Given the above, and throughout this paper, the term MCS is used interchangeably with minimal diagnosis, and the term MUS is used interchangeably with minimal explanation.

**Multiple Observations.** The MBD problem can be generalized to the situation where multiple inconsistent observations exist. In the presence of multiple observations, (3) is
modified as follows for observation $i$, $\text{Obs}_i$:

$$ SD_i \land \text{Obs}_i \land \bigwedge_{c \in \Delta} \text{Ab}(c) \land \bigwedge_{c \in \text{Comps} \setminus \Delta} \neg \text{Ab}(c) \not\perp $$

(4)

We assume that the system remains unchanged given different observations, and so $SD_i$ is solely a replica of the system description $SD$. A more general setting, in which the system considered also changes with the observation, could be considered, but would not change the main results in the paper. Observe that we need a distinct replica for each observation, since the actual values that result given the observation may differ.

**Definition 3 (MBD with Multiple Observations).** We assume a sequence of observations $\text{Obs}_i$, with $1 \leq i \leq r$. With each observation we associate a replica of the system $SD_i$, but such that the abnormal variables are shared by the different replicas. A minimal diagnosis $\Delta \subseteq \text{Comps}$ is a minimal set such that,

$$ \bigwedge_{i=1}^{r} (SD_i \land \text{Obs}_i) \land \bigwedge_{c \in \Delta} \text{Ab}(c) \land \bigwedge_{c \in \text{Comps} \setminus \Delta} \neg \text{Ab}(c) \not\perp $$

(5)

holds.

Thus, the goal is to find a subset-minimal (or possibly a cardinality-minimal) diagnosis $\Delta \subseteq \text{Comps}$ that makes the system consistent with any of the observations $\text{Obs}_i$, $1 \leq i \leq r$.

### 3 The Need for Multiple Observations

The formalization of model-based diagnosis presented in the previous section reveals fundamental challenges for MBD in practical settings. Concretely, MBD can be viewed as the process of identifying plausible (usually subset- or cardinality-minimal) guesses of which components in a system must be declared abnormal for consistency to be attained, when the abnormal components are allowed any behaviors. However, such guesses may not represent actual faulty components. The goal of MBD is to achieve consistency, but this is often possible by declaring faulty components that are unrelated with the actual bug. More importantly, complete enumeration of all diagnoses is infeasible in practice [34]. As a result, computed diagnoses should be as accurate as possible, given the information about failing observations.

This section investigates the importance of considering multiple observations to increase the accuracy of computed diagnoses. As shown below, it is simple to find failing observations which mislead the diagnosis tool, such that an exponentially large number of inaccurate and so irrelevant diagnoses are generated before the correct diagnosis is computed. More, importantly, this situation occurs in essentially endless situations. By considering multiple observations the accuracy of MBD can only improve. Whereas it is in general difficult to identify a test which will reveal as the source of the bug only the bug location, one can expect multiple tests to elicit cooperation such that only actual fault locations are reported.
A Buggy CNF Encoder. A well known example for SAT researchers is the debugging of CNF encoders. Suppose one implements an encoder to represent in CNF some problem. Suppose further that this encoder is composed on some modules (say $M_1$, $M_2$, $M_3$ and $M_4$), and that some have been tested and used before (e.g. $M_1$ and $M_3$) and that some others have recently been implemented (e.g. $M_2$ and $M_4$). Let us consider the example formula below.

$$\{ \neg x_{11} \lor y_{2a}, 1, \ldots, \neg x_{1r-1} \lor y_{2a}, 1, \neg y_{2a} \lor \neg t_{21a} \lor y_{21b}, 1, \ldots, \neg y_{21c} \lor \neg t_{21a} \lor y_{21d}, 1, \neg y_{21d} \lor \neg t_{21a} \lor w_{42a}, 1, \ldots \}$$

Some clauses are marked as hard since these result from modules of the encoder we know not buggy, e.g. $M_1$ and $M_3$. Some other clauses are generated by modules which we do not know whether they are buggy, e.g. $M_2$ and $M_4$. For the example shown, this is the case with the clauses containing variables from the sets of $\{y, t\}$ variables, or from the sets of $\{y, t, w\}$ variables, or from the sets of $\{t, u, z\}$ variables. Concretely, for $M_4$, the clauses only have literals in the set of $\{t, u, z\}$ variables (i.e. these are the last two clauses above). All the other soft clauses are produced by $M_2$. Let us assume further that the expected behavior of the CNF encoder is summarized in (7) below. Each line can be viewed as an observation (or as a failing test), since the assignments reported in each line are inconsistent with the model of the system given by (6).

| test | $x_{11}$ | $x_{12}$ | $x_{1r-1}$ | $x_{1r-1}$ | $t_{21a}$ | $t_{41a}$ | $z_{41a}$ | $z_{42a}$ |
|------|----------|----------|------------|------------|-----------|-----------|-----------|-----------|
| $t_1$ | 1        | 0        | $\cdots$  | 0          | 0         | 1         | 1         | 0         |
| $t_3$ | 0        | 1        | $\cdots$  | 0          | 0         | 1         | 1         | 0         |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$  | $\vdots$  | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$  |
| $t_{r-1}$ | 0        | 0        | $\cdots$  | 1          | 0         | 1         | 1         | 0         |
| $t_r$  | 0        | 0        | $\cdots$  | 0          | 1         | 1         | 1         | 0         |

More importantly, a quick inspection suggests that by declaring the clauses with the $w$, $u$ and $z$ variables as faulty, i.e. the last two clauses, we can make the rest of the formula consistent with all the tests. Unfortunately, if we decide to analyze each test separately, starting with $t_1$, then each of the first $r-1$ tests will produce $4^k + 4^{k/2} + 4^{k/2} + 1$ diagnoses, of which $4^k + 4^{k/2} + 4^{k/2}$ do not represent a valid diagnosis for all the tests, given $t_r$.

Clearly, there exists a single MCS, involving 2 clauses which, if removed, causes (6) to become consistent with all tests. This is the MCS we are interested in. However, if we opt to analyze each test separately, and for large enough $k$, it will be unrealistic to compute the correct MCS.
Detailed Analysis. For completeness, we derive the number of diagnoses indicated above for the “buggy” formula (6), given the set of tests (7).

To make the system consistent we can declare faulty the last two clauses:

\[
\{ (\neg u_{41a} \lor \neg t_{41a} \lor z_{41a}, 1), (\neg u_{42a} \lor \neg t_{41a} \lor z_{42a}, 1) \}
\]

(8)

This represents the diagnosis common to all tests. However, for the first \( r - 1 \) tests, we also need to consider the groups of four clauses:

\[
\{ (\neg y_{2a} \lor \neg t_{21a} \lor y_{2jb}, 1), (\neg y_{2jb} \lor \neg t_{21a} \lor y_{2pc}, 1), \\
(\neg y_{2pc} \lor \neg t_{21a} \lor y_{2pd}, 1), (\neg y_{2pd} \lor \neg t_{21a} \lor w_{4qa}, 1) \}
\]

(9)

with \( 1 \leq p \leq k \) and \( 1 \leq q \leq 2 \), such that \( q = 1 \) for \( p \) odd, and \( q = 2 \) for \( p \) even. Thus, for the first \( r - 1 \) tests there is a diagnosis between each group of four clauses (i.e., each value of \( p \)) for the two values of \( q \). Moreover, a diagnosis can be obtained by picking one group of four clauses (for one value of \( q \)) and one of the final clauses (for the other value of \( q \)). To get the total number of diagnoses for each of the first \( r - 1 \) tests, we just need to aggregate the different contributions. For the first set of groups of 4 clauses, the contribution is \( 4^k \) diagnoses. Between each group of four clauses and the corresponding final single clause, the contribution is \( 4^{\lfloor k/2 \rfloor} + 4^{\lceil k/2 \rceil} \) diagnoses. Finally, the last pair of clauses corresponds to a single diagnosis.

An Example with Lists of Rules. Figure 2a shows the rules of a production system\(^1\) intended to serve as an assistant in the holidays of some user. The rules are organized by layers, to simplify the analysis, and these layers are reflected in the numbers used for the rules. Suppose some user is spending her holidays at a Luxury Resort in the idyllic place of Andaman, and with more time available to engage on activities at the resort. As can be concluded, application of the rules in the production system would conclude that the user must see the Doctor and call the holidays off. Manual inspection of the rules in the production system reveals that the problem lies in rules R41a and R42a which derive an unlikely conclusion given the premises. Unfortunately, if the production system is much larger, the manual analysis of the rules will be unrealistic, and so automatic analysis needs to be considered. A number of test cases can be envisioned, a propositional formula can be derived, and MCSes of this formula can be computed. As highlighted earlier for the CNF encoder example, it may happen that an exponentially large number of MCSes is computed, most of which not reflecting actual MCSes given the complete set of failing observations. Perhaps not surprisingly, assuming the right set of productions are known not to be buggy, we can map this example into (6), and so the same analysis applies.

An Example with Datalog. Figure 2b proposes a different instantiation of the same problem, where Datalog is used instead of a production system. If the facts atAndaman, LuxuryResort and moreTime are added to the Datalog program, we will infer the same facts as above, i.e., we must see the doctor and must call the holidays off. Once more, inspection will reveal the problem in some specific rules, whereas model based diagnosis may yield an exponentially large number of possible diagnoses given a suitable set

\(^{1}\) A simplified propositional version of a production system is considered, to illustrate the key points.
of failing tests. Once again, assuming the right set of productions are known not to be buggy, we can map this example into (6), and so the same analysis applies.

4 Diagnosis of Multiple Observations

This section investigates approaches for computing diagnoses in the presence of multiple failing observations, and considers as working examples those studied in Sec-
Algorithm 1: Enumeration of minimal diagnoses by separate analysis and posterior assemblage

**Input:** \( SD_1, \ldots, SD_r, Obs_1, \ldots, Obs_r \)

**Output:** \( D = \{ \Delta_1, \Delta_2 \ldots \} \)

1. \((\Gamma_1, \ldots, \Gamma_r) \leftarrow (\emptyset, \ldots, \emptyset)\)
2. **foreach** \( i \in \{1, \ldots, r\} \):
   3. \( \Gamma_i \leftarrow \text{AllDiagnoses}(SD_i, Obs_i) \)
   4. \( D \leftarrow \text{DiagCombine}(\Gamma_1, \ldots, \Gamma_r) \)
   5. \( \text{ReportDiagnoses}(D) \)
   6. **return**

**Separate analysis and posterior assemblage.** As shown in Algorithm 1, one solution for simultaneous diagnosis of multiple observations is to generate all the diagnoses for each observation and then compute the diagnoses that correct all observations. This approach has been investigated by a number of researchers in the recent past [20, 22].

A major drawback of this approach is that in some settings, the enumeration of all the diagnoses for a given observation may be unrealistic [34]. Indeed, as analyzed in Section 3, for some observations there may exist an exponentially large number of diagnoses, most of which will then be discarded. Observe that even when diagnoses are computed by decreasing size, it will still be possible to force this algorithm to only find the correct diagnosis after computing an exponentially large number of (useless) diagnoses. In a similar vein, tentative approximations would not necessarily be effective. For example, an approach based on computing the union of one smallest size diagnosis (i.e. the MaxSAT solution) for each different observation would not necessarily solve the problem, since the smallest size diagnosis might not be accurate as well.

**Aggregated analysis.** Another solution consists of simply generating a model that represents \( r \) copies of the system, one for each observation, and then computing cardinality-minimal or subset-minimal diagnoses. This corresponds to computing MCSes or MaxSAT solutions of (5). The model to be generated essentially encodes (5) as a MaxSAT problem, and either uses an MCS extractor to compute a subset-minimal diagnoses or a MaxSAT solver for computing a cardinality-minimal diagnosis.

Although the aggregated problem formulation will only compute the actual subset-minimal (or cardinality-minimal) diagnoses for the set of observations, as we show in Section 6, it will be impractical for all but the smaller examples (or with a small number of observations), given the number of replicas of the system that need to be considered.

It should be noted that the examples in Section 3 aim at being as simple as possible. Let such an example be denoted by \( B = \langle H, S \rangle \). In general, \( B \) will be part of a much larger system (e.g. in fault localization, or spreadsheet debugging, among other examples). Concretely, the general setting will be \( B' = \langle H \cup G_H, S \cup G_S \rangle \), with \( G = G_H \cup G_S \) denoting the additional clauses used for encoding the system, but which are not essen-
Algorithm 2: Enumeration of minimal diagnoses

input : SD, Obs₁, ..., Obsᵣ
output: D = {Δ₁, Δ₂, ...}, U = {U₁, U₂, ...}

1 (H₁, ..., Hᵣ, S) ← Encode(SD, Obs₁, ..., Obsᵣ)
2 (D, U) ← (∅, ∅)
3 while true:
4   (st, Δ) ← MinHS(U, D) # find a min HS of U s.t. D
5   if not st:
6     break
7   foreach i ∈ {1, ..., r}:
8     (st, κ) ← SAT(Hᵢ ∪ (S \ Δ))
9     if not st:
10    U ← Reduce(κ) # U is MUS of Hᵢ ∪ (S \ Δ)
11    U ← U \ {Uᵢ}
12    ReportExp(U) # report min explanation
13   break
14 else: # if the loop was not broken
15    D ← D \ {Δ} # block diagnosis Δ
16    ReportDiag(Δ) # report min diagnosis
17   foreach i ∈ {1, ..., r}:
18     if not SAT(Hᵢ ∪ D): # no more diagnoses exist
19       return
20 return

5 Iterative Hitting Set Dualization

This section proposes an alternative approach for computing diagnoses given a (possibly large) set of observations. In contrast with the approaches described in the previous section, and so in contrast with earlier work, the proposed approach is shown to scale in practice.

The proposed approach hinges on recent work on hitting set dualization, which has been investigated in different contexts in recent years [8, 10, 16–18, 21, 23, 24, 32, 36, 37, 41, 45]. (However, these ideas can be traced to the seminal work of Reiter [38], and have been studied in different settings over the years [4, 25], among others.)

The proposed approach is summarized in Algorithm 2. Each Δᵢ denotes a computed minimal diagnosis, and each Uᵢ denotes a computed minimal explanation. Although the paper focuses mainly on subset-minimal diagnosis, the same algorithm can be used for computing cardinality-minimal diagnosis. The main difference is that for subset-
minimal diagnosis, MinHS can denote subset-minimal hitting sets, and for cardinality-minimal diagnosis, MinHS must denote cardinality-minimal hitting sets. As proposed in earlier work [4, 23, 24, 45], the algorithm iteratively computes minimal diagnoses and minimal explanations, and reports one in each main iteration of the algorithm. The key objective is to find a new minimal hitting set of (all) the explanations, and so a minimal diagnosis, at each iteration of the algorithm. If the computed minimal hitting set of the explanations is not a diagnosis, then a new (missing) minimal explanation is extracted, which is then added to the set of minimal explanations. If the computed minimal hitting set is indeed a diagnosis (for all observations), then it is discarded for future iterations by blocking the same hitting set from being computed.

In contrast with other enumeration approaches proposed recently [24], which can be viewed as targeting enumeration of explanations, Algorithm 2 will terminate as soon as all the diagnoses have been computed, even if some explanations have not yet been identified (see lines 17–19). Indeed, as soon as all diagnoses for some observation have been computed and blocked, one cannot find another way to recover consistency for that observation. The lines line 17–19 can in practice be made optional if the goal is to compute some number \( K \) of diagnoses.

It is important to note that, in theory Algorithm 2 can compute an exponentially large number of explanations in between computed diagnoses. However, as the experimental results demonstrate, this worst-case scenario is not observed in practice.

Example 1. Let us consider a system with \( r \) failing observations, each of which has exactly two explanations: \( \{c_1\} \) and \( \{c_2\} \). Consider some minimal hitting set that does not pick either \( c_1 \) or \( c_2 \). Then, the next computed explanation will require the missing component to be also hit (picked) in future minimal hitting sets. Thus, Algorithm 2 will compute the correct diagnosis, consisting of both \( c_1 \) and \( c_2 \) in two iterations.

One essential aspect of the solution proposed by Algorithm 2 is that a single copy of the system is used throughout. In the presence of a large number of observations, this can represent a crucial improvement.

6 Preliminary Experimental Results

The experimental evaluation was performed in Ubuntu Linux on an Intel Xeon E5-2630 2.60GHz processor with 64GByte of memory. The time limit was set to 600s and the memory limit to 10GByte for each individual instance to run. A prototype of the proposed iterative hitting set dualization (IHSD) approach referred to as DEx (Diagnosis Extractor) was implemented in C++ and consists of two interacting parts. One of them computes subset-minimal or cardinality-minimal hitting sets of the set of explanations. The other part tests consistency of the system provided that the hitting set components are disabled.

Enumeration of cardinality-minimal solutions is achieved with the use of an incremental implementation of the MaxSAT algorithm based on soft cardinality constraints [1, 33], which is the state-of-the-art MaxSAT algorithm that won several categories in the MaxSAT Evaluation 2015 and 2016. Computing subset-minimal solutions is done with the use of the LBX algorithm [29] and its further improvements [28] for enumerating MCSes for a given unsatisfiable formula. In the performed evaluation,
DEx is configured to compute subset-minimal solutions. The proposed algorithm was compared to (1) the naive approach of separate analysis and posterior assemblage and (2) the approach of aggregated analysis. Both comparisons are detailed below.

**Comparison to separate analysis.** The idea of comparing the proposed approach to separate analysis is to show that enumerating diagnoses for individual observations can be infeasible in practice. Here, we ignore the assemblage phase of the naive approach and focus only on enumerating diagnoses instead. However, it should be noted that the assemblage phase would clearly impose an additional overhead. As a test material, we constructed a family of “buggy encoder” CNF instances described in Section 3 (see (6)). For that, we considered the number of observations $r$ varying from 10 to 300 with step 10, i.e. $r \in \{10, 20, 30, \ldots, 290, 300\}$. The number $k$ varies from 2 to 9 and also from 10 to 300 with step 10. The total number of CNF formulas in the constructed family is 1140. Recall that the exact number of diagnoses for each of the $r$ observations in the considered benchmarks is $4^k + 4^{\lfloor k/2 \rfloor} + 4^{\lceil k/2 \rceil} + 1$, which makes it impractical to enumerate all diagnoses for any reasonably large $k$. Here we are aiming at confirming practically that the applicability of the separate analysis approach is rather limited with respect to the considered set of instances. The separate analysis phase of the naive approach is represented in our evaluation by the two well-known MCS enumerators: RS [3] and LBX [29].

Figure 3 shows the performance of the proposed approach compared to separate analysis utilizing solvers LBX and RS. As one can observe in Figure 3a, DEx can efficiently compute the correct diagnosis for all the 1140 benchmarks. The time spent for the largest instance, is about 0.1 second. In contrast, separate analysis is able to do a comprehensive enumeration of all diagnoses only for 177 and 185 instances when using LBX and RS, respectively. Note that both tools can successfully solve only instances with $k < 8$. Figure 3b shows how close the LBX and RS solvers are to enumerate all the diagnoses for each of the individual observations. As expected, LBX and RS compute 100% of diagnoses only for the solved instances, i.e. 177 and 185 instances,
Table 1: Comparison of the formula size and running time of the IHSD-based approach and aggregated analysis for the example of (6) for growing values of $r$ and $k$.

| $r$ | $k$ | variables | clauses | time  | variables | clauses | LBX  | RS  |
|-----|-----|-----------|---------|-------|-----------|---------|------|-----|
| 10  | 10  | 50        | 56      | 0.01  | 592       | 616     | 0.01 | 0.01|
| 50  | 50  | 210       | 256     | 0.01  | 10 912    | 13 056  | 0.02 | 0.03|
| 100 | 100 | 410       | 506     | 0.02  | 41 812    | 51 106  | 0.08 | 0.09|
| 200 | 200 | 810       | 1 006   | 0.04  | 163 612   | 202 206 | 0.27 | 0.31|
| 500 | 500 | 2 010     | 2 506   | 0.19  | 1 009 012 | 1 255 506 | 1.75 | 1.84|
| 500 | 1 000 | 3 510   | 4 506   | 0.33  | 1 762 512 | 2 257 506 | 3.24 | 3.35|
| 500 | 2 000 | 6 510   | 8 506   | 0.49  | 3 269 512 | 4 261 506 | 6.15 | 6.33|

respectively. Observe that the percentage of computed diagnoses drops very quickly and is $\approx 0\%$ for most of the instances.

**On aggregated analysis.** The purpose of the following discussion is to show how dramatically the formula size grows with the growth of the number of observations $r$ for the example CNF instance of (6). Table 1 compares the size of the original formula (in terms of the number of variables and clauses) fed to the proposed IHSD approach and the size of the formula resulted from aggregating all observations. Additionally, the table shows the running time of the proposed approach and the aggregated analysis implemented with the use of LBX and RS. Note that the formula construction time is excluded here while in practice it can noticeably contribute to the total running time of the solver. As one can see, a reasonably large number of observations results in a formula having a few orders of magnitude more variables and clauses than in the original formula. As an example, while the original formula has a few thousand variables and clauses, the size of the aggregated formula goes beyond millions of variables and clauses, which makes it hard to build and deal with. As mentioned earlier, the example of (6) is designed to be as simple as possible while illustrating the main points of the paper. However, in practice formulas can contain an arbitrarily large number of additional clauses that do not help revealing the culprits of system’s inconsistency but contribute a lot to the aggregated formula size. To confirm this, Table 2 shows the rate of aggregated formula growth given the size of some (arbitrary) original formulas. As one can observe, given a system, which is inconsistent with a few hundred observations, encoded as a formula with a few hundred of thousands of variables and a few million of clauses (which may well happen in practice), the aggregated formula contains millions of variables and clauses, which makes it hard to deal with and results in the aggregated approach being ineffective in practice.

## 7 Conclusions & Research Directions

Model based diagnosis and its many practical instantiations find a growing number of important practical applications. Recent work on model based diagnosis has focused on efficiently computing cardinality-minimal diagnoses [27, 30, 31] and on efficiently listing subset-minimal diagnoses [34, 45]. Another important problem is how to com-
Table 2: Asymptotic growth for the aggregated formulas.

| r   | Original formula | Aggregated formula |
|-----|------------------|--------------------|
|     | variables clauses| variables clauses  |
| 10  | 100              | 1 000              |
| 100 | 1 000            | 100 000            |
| 200 | 10 000           | 2 000 000          |
| 200 | 100 000          | 4 000 000          |
| 300 | 100 000          | 30 000 000         |

This paper shows that existing solutions are bound to be ineffective in practical domains, for one of two reasons: (i) approaches based on separate analysis may need to compute a tool large number of diagnoses; and (ii) approaches based on aggregate analysis may need to analyze too large formulas. As a result, the paper proposes a novel approach for computing (subset or cardinality) minimal diagnoses in the presence of multiple (failing) observations, by exploiting implicit hitting set dualization. The experimental results confirm both the efficiency and scalability of the proposed approach.

Future work will validate the performance gains also in the case of cardinality-minimal diagnoses, which can be implemented seamlessly using the proposed approach. In addition, the ideas proposed in this paper will be applied in different application domains.

References

1. B. Andres, B. Kaufmann, O. Matheis, and T. Schaub. Unsatisfiability-based optimization in clasp. In ICLP, pages 211–221, 2012.
2. M. Arenas, L. E. Bertossi, and J. Chomicki. Answer sets for consistent query answering in inconsistent databases. TPLP, 3(4-5):393–424, 2003.
3. F. Bacchus, J. Davies, M. Tsimpoukelli, and G. Katsirelos. Relaxation search: A simple way of managing optional clauses. In AAAI, pages 835–841, 2014.
4. J. Bailey and P. J. Stuckey. Discovery of minimal unsatisfiable subsets of constraints using hitting set dualization. In PADL, pages 174–186, 2005.
5. A. Bauer. Simplifying diagnosis using LSAT: A propositional approach to reasoning from first principles. In CPAIOR, pages 49–63, 2005.
6. A. Biere, M. Heule, H. van Maaren, and T. Walsh, editors. Handbook of Satisfiability. IOS Press, 2009.
7. E. Birnbaum and E. L. Lozinskii. Consistent subsets of inconsistent systems: structure and behaviour. J. Exp. Theor. Artif. Intell., 15(1):25–46, 2003.
8. K. Chandrasekaran, R. M. Karp, E. Moreno-Centeno, and S. Vempala. Algorithms for implicit hitting set problems. In SODA, pages 614–629, 2011.
9. A. Darwiche. Model-based diagnosis using structured system descriptions. J. Artif. Intell. Res. (JAIR), 8:165–222, 1998.
10. J. Davies and F. Bacchus. Solving MAXSAT by solving a sequence of simpler SAT instances. In CP, pages 225–239, 2011.
11. J. de Kleer. Minimum cardinality candidate generation. In DX, pages 397–402, 2009.
12. J. de Kleer, A. K. Mackworth, and R. Reiter. Characterizing diagnoses and systems. Artif. Intell., 56(2-3):197–222, 1992.
13. H. Do, S. G. Elbaum, and G. Rothermel. Supporting controlled experimentation with testing techniques: An infrastructure and its potential impact. Empirical Software Engineering, 10(4):405–435, 2005.
14. A. Feldman, G. Provan, J. de Kleer, S. Robert, and A. van Gemund. Solving model-based diagnosis problems with Max-SAT solvers and vice versa. In DX, pages 185–192, 2010.
15. M. Gebser, T. Schaub, S. Thiele, and P. Veber. Detecting inconsistencies in large biological networks with answer set programming. TPLP, 11(2-3):323–360, 2011.
16. A. Ignatiev, A. Morgado, and J. Marques-Silva. Propositional abduction with implicit hitting sets. In ECAL, pages 1327–1335, 2016.
17. A. Ignatiev, A. Previti, M. H. Liffiton, and J. Marques-Silva. Smallest MUS extraction with minimal hitting set dualization. In CP, pages 173–182, 2015.
18. A. Ignatiev, A. Previti, and J. Marques-Silva. On finding minimum satisfying assignments. In CP, pages 287–297, 2016.
19. D. Jannach and T. Schmitz. Model-based diagnosis of spreadsheet programs: a constraint-based debugging approach. Autom. Softw. Eng., 23(1):105–144, 2016.
20. M. Jose and R. Majumdar. Cause clue clauses: error localization using maximum satisfiability. In PLDI, pages 437–446, 2011.
21. R. M. Karp. Implicit hitting set problems and multi-genome alignment. In CPM, page 151, 2010.
22. S. Lamraoui and S. Nakajima. A formula-based approach for automatic fault localization of multi-fault programs. JIP, 24(1):88–98, 2016.
23. M. H. Liffiton and A. Malik. Enumerating infeasibility: Finding multiple MUSes quickly. In CPAIOR, pages 160–175, 2013.
24. M. H. Liffiton, A. Previti, A. Malik, and J. Marques-Silva. Fast, flexible MUSes quickly. Constraints, 21(2):223–250, 2016.
25. M. H. Liffiton and K. A. Sakallah. Algorithms for computing minimal unsatisfiable subsets of constraints. J. Autom. Reasoning, 40(1):1–33, 2008.
26. J. Marques-Silva, F. Heras, M. Janota, A. Previti, and A. Belov. On computing minimal correction subsets. In IJCAI, pages 615–622, 2013.
27. J. Marques-Silva, M. Janota, A. Ignatiev, and A. Morgado. Efficient model based diagnosis with maximum satisfiability. In IJCAI, pages 1966–1972, 2015.
28. C. Mencía, A. Ignatiev, A. Previti, and J. Marques-Silva. MCS extraction with sublinear oracle queries. In SAT, pages 342–360, 2016.
29. C. Mencía, A. Previti, and J. Marques-Silva. Literal-based MCS extraction. In IJCAI, pages 1973–1979, 2015.
30. A. Metodi, R. Stern, M. Kalech, and M. Codish. Compiling model-based diagnosis to Boolean satisfaction. In AAAI, pages 793–799, 2012.
31. A. Metodi, R. Stern, M. Kalech, and M. Codish. A novel sat-based approach to model based diagnosis. J. Artif. Intell. Res. (JAIR), 51:377–411, 2014.
32. E. Moreno-Centeno and R. M. Karp. The implicit hitting set approach to solve combinatorial optimization problems with an application to multigenome alignment. OR, 61(2):453–468, 2013.
33. A. Morgado, A. Ignatiev, and J. Marques-Silva. MSCG: Robust core-guided MaxSAT solving. JSAT, 9:129–134, 2015.
34. I. Nica, I. Pill, T. Quaritsch, and F. Wotawa. The route to success - A performance comparison of diagnosis algorithms. In IJCAI, pages 1039–1045, 2013.
35. I. Nica and F. Wotawa. ConDiag - computing minimal diagnoses using a constraint solver. In DX, pages 185–192, 2012.
36. A. Previti, A. Ignatiev, A. Morgado, and J. Marques-Silva. Prime compilation of non-clausal formulae. In *IJCAI*, pages 1980–1988, 2015.
37. A. Previti and J. Marques-Silva. Partial MUS enumeration. In *AAAI*, pages 818–825, 2013.
38. R. Reiter. A theory of diagnosis from first principles. *Artif. Intell.*, 32(1):57–95, 1987.
39. A. Roychoudhury and S. Chandra. Formula-based software debugging. *Commun. ACM*, 59(7):68–77, 2016.
40. S. Safarpour, H. Mangassarian, A. G. Veneris, M. H. Lifititon, and K. A. Sakallah. Improved design debugging using maximum satisfiability. In *FMCAD*, pages 13–19, 2007.
41. P. Saikko, J. Berg, and M. Järvisalo. LMHS: A SAT-IP hybrid MaxSAT solver. In *SAT*, pages 539–546, 2016.
42. S. A. Siddiqi. Computing minimum-cardinality diagnoses by model relaxation. In *IJCAI*, pages 1087–1092, 2011.
43. S. A. Siddiqi and J. Huang. Hierarchical diagnosis of multiple faults. In *IJCAI*, pages 581–586, 2007.
44. A. Smith, A. G. Veneris, M. F. Ali, and A. Viglas. Fault diagnosis and logic debugging using Boolean satisfiability. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 24(10):1606–1621, 2005.
45. R. T. Stern, M. Kalech, A. Feldman, and G. M. Provan. Exploring the duality in conflict-directed model-based diagnosis. In *AAAI*, pages 828–834, 2012.
46. B. C. Williams and R. J. Ragno. Conflict-directed A* and its role in model-based embedded systems. *Discrete Applied Mathematics*, 155(12):1562–1595, 2007.
47. W. E. Wong, R. Gao, Y. Li, R. Abreu, and F. Wotawa. A survey on software fault localization. *IEEE Trans. Software Eng.*, 42(8):707–740, 2016.