# Improving Abductive Diagnosis Through Structural Features: A Meta-Approach

**Roxane Koitz**[1] and **Franz Wotawa**[2]

**Abstract.** While abductive reasoning provides an intuitive approach to diagnosis, its computational complexity remains an obstacle. Even though certain model representations are tractable, computing solutions for instances of reasonable size and complexity persists to pose a challenge. Hence, the discovery of efficient methods to derive abductive explanations presents itself as appealing research area. In this paper, we investigate the structural properties inherent to formalizations suitable for abductive failure localization. Based on the features extracted we construct a meta-approach exploiting a machine learning classifier to predict the abductive reasoning technique yielding the "best" performance on a specific diagnosis scenario. To assess whether the proposed attributes are in fact sufficient for forecasting the appropriate abduction procedure and to evaluate the efficiency of our algorithm selection in comparison to traditional abductive reasoning approaches, we conducted an empirical experiment. The results obtained indicate that the trained model is capable of predicting the most efficient algorithm and further, we can show that the meta-approach is capable of outperforming each single abductive reasoning method investigated.

## 1 Introduction

Being able to accurately identify the source of an unintended system behavior is an essential objective in various application domains. Abductive reasoning appears to be a natural approach to diagnosis as it infers consistent explanations from background knowledge and observed symptoms based on the notion of entailment. Usually a set of constraints, such as minimality, are placed on whether a solution suffices as an abductive explanation or not.

While diagnosis is the most prevalent application area for abductive reasoning, abduction has been applied to a diverse set of problems such as planning [32], natural language processing [29], and image interpretation [39]. A large body of literature has investigated approaches to mechanizing abductive reasoning such as consequence finding [22], proof-tree completion [23], set-covering [30], abductive model-based diagnosis [3] or abductive logic programming [15, 6].

Within this paper we concentrate on two methods, namely parsimonious set covering and abductive model-based diagnosis. The set covering theory by Peng and Reggia [31] utilizes a causal associative network recording the relations between disorders and their manifestations. In their simple model, these cause and effect sets are strictly disjoint. A diagnosis then is a set of disorders covering, i.e. explaining, the set of observed symptoms. Later the approach has been extended to incorporate probability theory and several refinements to the basic theory have been proposed such as the improvement of models with additional knowledge or the inclusion of more complex covering relations [1].

In model-based diagnosis a formalization of the system under consideration is exploited to determine causes for anomalies [34]. While the traditional approach utilizes a representation of the correct system behavior and derives diagnoses via inconsistencies, the abductive variant operates on a model describing the way faults affect measurable system variables. Through the notion of entailment abductive model-based diagnosis reasons about causes of a set of observations [3]. Considering a restricted problem space, simple set covering and abductive model-based diagnosis are equivalent [23].

Even though there exist certain subsets of logics where abduction is tractable, it generally is an NP-hard problem which grows exponentially in the size of the model [27]. Hence, within this paper we investigate algorithm selection as a means to efficiently compute abductive explanations in the context of diagnosis. First formalized by Rice [35], algorithm selection aims at identifying the "best performing" approach for a specific problem instance. The basic building blocks within this framework are a portfolio of algorithms to choose from, empirical performance data of the algorithms on representative problems and a set of features, which are used to get a notion of the difficulty of a problem instance [13]. On grounds of the empirical data and the feature vector, a predictor can be trained capable of determining the most suitable approach for a distinct sample from the problem space [17]. Machine learning has been identified as a feasible approach to use as a prediction tool. Leyton-Brown et al. [20] describe their portfolio approach to algorithm selection, where they train an empirical hardness model for each algorithm within their portfolio to forecast each approach's computation time on the instance and execute the one predicted as most efficient. Algorithm selection has been applied very successfully in the domain of SAT [38], graph coloring [26], or tree-decomposition [24].

In this paper, we restrict the problem space to propositional Horn clause models which can be automatically generated from failure assessments available in practice. These analyses hold information on faults and their effects and thus are suitable knowledge sources for abductive diagnosis. The resulting logical system descriptions are characterized by certain structural properties, which we utilize as features for the algorithm selection process. We extracted these attributes for a collection of instances and evaluated several abductive diagnosis algorithms empirically on their computation time for the entire sample set. On basis of the performance data and the features we trained a machine learning classifier to forecast the algorithm most suitable in regard to its runtime for a particular abductive diagnosis scenario. We embedded the selection process within a meta-algorithm, which generates the structural metrics for a given diagnosis problem, categorizes it on the previously trained classifier

[1] Authors are listed in alphabetical order.

[2] Graz University of Technology, email: {$rkoitz$, $wotawa$}@ist.tugraz.at

and computes the diagnoses using the algorithm chosen by the predictor.

We organize our paper as follows. The next section provides a formal introduction to abductive model-based diagnosis as well as the parsimonious set covering approach. Further, we show that in their simplest form they are equivalent. Section 3 discusses the type of logical formalizations we examine and describes the structural characteristics forming our features for the algorithm selection. Subsequently, we discuss the meta-approach in more detail and in Section 4.3 we empirically evaluate it in comparison to the algorithms in our portfolio. Lastly we summarize the paper and provide an outlook to future work.

## 2 Abductive Diagnosis

Within this section we consider two abductive diagnosis approaches, namely abductive model-based diagnosis [3] operating on propositional Horn clauses and the simple set covering approach [31]. Specifically, we show their equivalence and describe the corresponding elements within each method.

### 2.1 Model-Based Diagnosis

As model-based diagnosis requires a formal description of the system, its abductive variant utilizes a representation of the connections between failures and their manifestations. Based on the information available, the task is to search for a set of consistent causes which together with the background theory logically entail a set of observed fault indicator. Since abduction is a hard problem, research has focused on subsets of logics which allow to compute explanations in polynomial time [27]. An important restriction on the underlying formulas is the Horn property as for this fragment abduction is still tractable. Within the context of diagnosis a further syntactical restriction often imposed is a definite Horn theory since it often suffices to describe causal relations [7].

Therefore, we concentrate on propositional definite Horn descriptions and define in a similar manner as Friedrich et al. [9] a knowledge base.

**Definition 1 (Knowledge base (KB))** *A knowledge base (KB) is a tuple (A,Hyp,Th) where A denotes the set of propositional variables, Hyp $\subseteq$ A the set of hypotheses, and Th a set of Horn clause sentences over A.*

The set of hypotheses $Hyp$ comprises the propositional variables allowed to form a diagnosis, while the theory describes the relations between the variables. In this context, we further specify the propositional variables not constituting a hypothesis, i.e. $\{A \setminus Hyp\}$, as effects or symptoms. We will refer to this set of variables as $\Sigma$ within this paper. Since we aim at identifying root causes for failure manifestations, an abduction problem considers a knowledge base $KB$ as well as a set of symptoms to explain. We therefore define a Propositional Horn Clause Abduction Problem (PHCAP) as follows:

**Definition 2** ***(Propositional Horn Clause Abduction Problem (PHCAP))*** *Given a knowledge base (A,Hyp,Th) and a set of observations Obs $\subseteq$ A then the tuple (A,Hyp,Th,Obs) forms a Propositional Horn Clause Abduction Problem (PHCAP).*

**Definition 3 (Diagnosis; Solution of a PHCAP)** *Given a PHCAP (A,Hyp,Th,Obs). A set $\Delta \subseteq$ Hyp is a solution if and only if $\Delta \cup Th \models Obs$ and $\Delta \cup Th \not\models \bot$. A solution $\Delta$ is parsimonious or minimal if and only if no set $\Delta' \subset \Delta$ is a solution.*

A solution to a PHCAP is an abductive diagnosis, as it provides hypotheses consistently explaining the occurrence of a set of observations. As in practice minimal solutions are preferred, we require the diagnoses to be subset minimal.

**Example 1:** Consider the following $KB$:

$$A = \{h_1, h_2, h_3, o_1, o_2, o_3\}, Hyp = \{h_1, h_2, h_3\},$$

$$Th = \{\ h_1 \rightarrow o_1, h_2 \rightarrow o_1, h_2 \rightarrow o_2, h_3 \rightarrow o_2, h_3 \rightarrow o_3\ \}$$

Assume we can observe $o_1$ and $o_3$, i.e. $Obs = \{o_1, o_3\}$. The solutions to the PHCAP, i.e. the minimal abductive diagnoses, are $\Delta_1 = \{h_1, h_3\}$ and $\Delta_2 = \{h_2, h_3\}$.

### 2.2 Parsimonious Set Covering

Abduction by parsimonious set covering is based in its simplest form on an associative network encompassing the causal links between possible disorders and their manifestations [31]. A diagnosis problem is a 4-tuple $P =< D, M, C, M^+ >$, where $D$ is the set of disorders, $M$ comprises the manifestations, $C$ defines the causal connections, and $M^+$ represents the current set of symptoms observed. The knowledge about the causal relations is defined by two sets: *effects*$(d_i)$ and *causes*$(m_j)$. For each disorder $d_i$ we can define *effects*$(d_i) = \{m_j | < d_i, m_j > \in C\}$ as the set of manifestations cause by the disorder. Similarly, the set *causes*$(m_j) = \{d_i | < d_i, m_j > \in C\}$ holds the disorders which directly trigger manifestation $m_j$ [31]. Thus, for any subset of disorders $D_I$, we can determine the objects directly caused by it as

$$\textit{effects}(D_I) = \bigcup_{d_i \in D_I} \textit{effects}(d_i)$$

Along similar lines, we can observe that

$$\textit{causes}(M_J) = \bigcup_{m_j \in M_J} \textit{causes}(m_j)$$

As mentioned within this approach abductive explanations are defined as the causes covering the observed symptoms. A set of disorders $D_I$ is said to cover a set of manifestations $M_J \subseteq M$ if $M_J \subseteq$ *effects*$(D_I)$, i.e. the former causally infers the latter. While minimality is not a necessary condition for a cover in the original definition of Peng and Reggia [31], we introduce the further requirement that the cover is subset minimal.

**Definition 4 (Cover)** *A set $D_I \subseteq D$ is said to cover $M_J \subseteq M$ if $M_J \subseteq$ effects$(D_I)$ and there exists no $D'_I \subset D_I$ with $M_J \subseteq$ effects$(D'_I)$.*

Thus, we can define a solution to a set covering problem as a subset $D_I \subseteq D$ covering $M^+$.

**Definition 5 (Set Cover Diagnosis)** *Given a diagnosis problem $P$. A set $\Delta \subseteq D$ is said to be a diagnosis iff $\Delta$ covers $M^+$.*

In regard to the logic-based definitions discussed for model-based diagnosis, the disorders refer to the set $Hyp$ in the PHCAP framework. Their manifestations constitute the effects $\Sigma$, $M^+$ corresponds to $Obs$, and the network represents the domain theory. A causal relation $< d_i, m_j >$ is recorded in $C$ whenever there is a logical implication of the form $d_i \rightarrow m_j$, where $d_i \in Hyp$ and $m_j \in \Sigma$ within the theory $Th$. Thus, it is apparent that the simple set covering framework is equivalent to logic-based abduction with a theory

restricted to definite Horn clauses [23]. As both methods generate the explanatory causes based on the relationships between disorders and effects, they compute abductive explanations. That is a set covering diagnosis, as defined previously, corresponds to a minimal diagnosis in the PHCAP framework.

**Example 1 (cont):** Considering our previous example, the diagnosis problem $P$ can be reformalized in set covering:

$$D = \{h_1, h_2, h_3\}, M = \{o_1, o_2, o_3\}, M^+ = \{o_1, o_3\},$$

$$C = \left\{ \begin{array}{c} < h_1, o_1 >, < h_2, o_1 >, \\ < h_2, o_2 >, < h_3, o_2 >, < h_3, o_3 > \end{array} \right\}$$

We can obtain the set covering diagnoses by determining the disorder sets $D_I$ where *effects*$(D_I)$ cover $M^+$, which are *effects*$(\{h_1, h_3\}) = \{o_1, o_2, o_3\}$ and *effects*$(\{h_2, h_3\}) = \{o_1, o_2, o_3\}$. Hence, the diagnoses are $\Delta_1 = \{h_1, h_3\}$ and $\Delta_2 = \{h_2, h_3\}$.

The equivalence between set covering and the hitting set problem has been established [16], thus we can exploit the notion of hitting sets in order to define a diagnosis within the parsimonious set covering theory. In particular, we stated previously that a cover implies a causal dependency between disorders and manifestations and can be expressed through the *effects* relation. Along similar lines *causes*$(m_j)$ comprises information on all disorders responsible for $m_j$. Hence, by computing the hitting set of *causes*$(m_j)$ for a single manifestation, we derive a disjunction of all disorders possibly leading to $m_j$, i.e. each disorder constitutes a diagnosis. For multiple observations, i.e. $m_1, m_2, \ldots, m_n \in M^+$, the hitting sets of all *causes*-sets of the current manifestations form the diagnoses. This is apparent since in order to represent a solution one disorder explaining each observation has to be present within each diagnosis. To impose the parsimonious criteria, we restrict solutions to subset minimal hitting sets [30].

**Definition 6 (Abductive Hitting Set Diagnosis)** *Given a diagnosis problem $P$. A set $\Delta \subseteq D$ is said to be a minimal diagnosis iff $\Delta$ is a minimal hitting set of $S$, where $\forall m_j \in M^+$ : causes$(m_j) \in S$.*

**Example 1 (cont):** The *causes* sets for the current manifestations are *causes*$(o_1) = \{h_1, h_2\}$ and *causes*$(o_3) = \{h_3\}$, thus *causes*$(o_1) \in S$ and *causes*$(o_3) \in S$. The minimal hitting set of $S$ correspond to $\Delta_1 = \{h_1, h_3\}$ and $\Delta_2 = \{h_2, h_3\}$.

## 3 Models

An essential issue in model-based diagnosis has been the construction of system descriptions suitable for identifying faults. Thus, numerous techniques to automatically extract models have been proposed with a recent method taking advantage of Failure Mode Effect Analysis (FMEA) records [37]. This type of risk evaluation is becoming increasingly common and collects data on how faults on a component level influence system variables [12]. Table 1 depicts a simplified example of an FMEA where each row contains a component, a possible fault of said component and its corresponding effects. Since it captures the causal associations between defects and their consequences it provides information necessary for abductive reasoning.

In the straightforward mapping presented by Wotawa [37] each record of the FMEA is transformed into a set of Horn clause sentences, where the component-fault mode pair implies an effect. These formulas form the theory $Th$ of a $KB$ which we can utilize to compute abductive diagnoses based on a set of fault indicators. The

**Table 1**: Example 2: FMEA taken from the wind turbine domain.

| Component | Fault Mode | Effect |
|---|---|---|
| Fan | Corrosion | P_turbine |
| Fan | TMF | T_cabinet, P_turbine |
| IGBT | HCF | T_cabinet, T_nacelle |

set $A$ then simply encompasses all proposition variables, while the component-fault pairs compose $Hyp$.

**Example 2:** Transforming the FMEA given in Table 1 would lead to the following $KB$:

$$Hyp = \left\{ \begin{array}{c} mode(Fan, Corrosion), \\ mode(Fan, TMF), mode(IGBT, HCF) \end{array} \right\}$$

$$A = \left\{ \; mode(Fan, Corrosion), T\_cabinet, P\_turbine, \ldots \; \right\}$$

$$Th = \left\{ \begin{array}{c} mode(Fan, Corrosion) \rightarrow P\_turbine, \\ mode(Fan, TMF) \rightarrow P\_turbine, \\ mode(Fan, TMF) \rightarrow T\_cabinet, \\ mode(IGBT, HCF) \rightarrow T\_cabinet, \\ mode(IGBT, HCF) \rightarrow T\_nacelle \end{array} \right\}$$

### 3.1 Diagnosing the Models

As is apparent from the example, the result of the mapping is a model consisting of bijunctive definite Horn clauses. Thus, we can simply apply either model-based or set covering abduction to compute diagnoses. However, since we would like to extend our approach to different failure assessments in practice, we allow more expressive formalizations, i.e. conjunctions of causes and disjunction of manifestations. In order to use these models within the PHCAP and simple set covering approach, we currently compile them into define Horn clauses. It is apparent that this increases the theory's cardinality, requires to reassemble the original hypotheses at the end of the diagnosis and to ensure that subset minimality is still present.

Now we explain how we can utilize the logical models obtained on basis of the FMEAs to perform abductive reasoning within the set cover approach. Given a model of this type we can easily represent it as a hypergraph $H = (V, E)$, where $V$ is the set of vertices and $E$ constitutes the set of hyperedges. The nodes of the hypergraph represent the propositional variables, while the hyperedges are determined by the theory. For each clause there exists a hyperedge containing all propositional variables of said clause, i.e. $\forall a \in A \rightarrow a \in V$ and $\forall c \in Th \rightarrow \bigcup_{l \in c} |l| \in E$ where $||$ is a function mapping literals to the underlying propositions ignoring negations, i.e., $|\neg p| = p$ and $|p| = p$ for all $p \in A$. In Figure 1 on the right hand side a hypergraph representation of *Example 1* is shown.

Following this representation we can assign a label to each vertex within a hyperedge $E$, such that:

$$label(v) = \begin{cases} \{v\} & \text{if } v \in Hyp \\ \bigcup\limits_{x \in E \wedge x \neq v} label(x) & \text{otherwise} \end{cases}$$

In case a vertex represents a manifestation, its label correspond to its *causes*-set, as it holds the hypotheses responsible for the effect. Thus, we can utilize the labels of the nodes representing the observations to compute the abductive diagnoses as hitting sets. Note that by relying on this notion we could further handle intermediate effects, which we do not discuss in more detail in this paper.

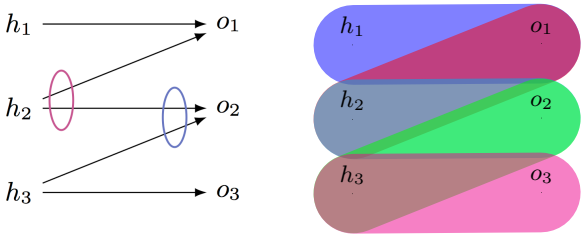



**Figure 1**: DAG and hypergraph representation of Example 1. The DAG shows shared hypotheses (left oval) and common effects (right oval) for pairs of nodes.

## 3.2 Structural Metrics

As stated the models we are considering are bijunctive definite Horn clauses. Thus, we can easily define their structural properties based on various graph representations. In the simplest case the theory is characterized as a directed acyclic graph (DAG) with two disjunctive node sets, namely the propositional variables constituting the causes and the effects, respectively. This representation is equivalent to the associative network as described by Peng and Reggia [31] in their set covering approach. Furthermore, it is apparent that by constructing an undirected graph we receive a bipartite graph.

Considering the graph representations of the model, we can extract certain characteristics of their structure which we subsequently use within the algorithm selection process. As abductive diagnosis is possibly exponential in the number of causes to consider, the cardinality of $Hyp$ is an intuitive measure complexity. In addition, we collect the number of effects and connections within the theory.

### 3.2.1 Outdegree and Indegree

Based on the DAG, we can compute for each vertice representing a hypothesis its outdegree, which specifies the number of manifestations affected by said cause. Similarly, we measure the indegree of each effect, i.e. the number of hypotheses inferring the manifestation. Considering the set covering framework we can define the degrees as follows:

$$outdegree(h_i) = |\mathit{effects}(h_i)|$$
$$indegree(m_j) = |\mathit{causes}(m_j)|$$

In regard to Example 1 we can observe $outdegree(h_2) = |\mathit{effects}(h_2)| = 2$ and $indegree(o_3) = |\mathit{causes}(o_3)| = 1$. Collected over the entire model these measures provide an intuitive metric of the basic magnitude of the theory and the connectedness of the graph.

### 3.2.2 Covering and Overlap

Several disorders may cover the same effect, i.e. a manifestation can be explained by multiple causes. On basis of this we can define a covering metric for each pair of hypotheses as the ratio between the number of common effects and the total number of symptoms induced by the hypotheses:

$$\mathit{covering}(h_i, h_j) = \frac{|\mathit{effects}(h_i) \cap \mathit{effects}(h_j)|}{|\mathit{effects}(h_i) \cup \mathit{effects}(h_j)|}$$

Figure 1 depicts on the right hand side of the DAG the shared observation $o_2$ between $h_2$ and $h_3$ as a blue oval. Thus, we can see that $\mathit{covering}(h_2, h_3) = \frac{1}{3}$.

In a similar manner, we define the overlap of two effects as their common sources in relation to all their causes:

$$\mathit{overlap}(o_i, o_j) = \frac{|\mathit{causes}(o_i) \cap \mathit{causes}(o_j)|}{|\mathit{causes}(o_i) \cup \mathit{causes}(o_j)|}$$

The overlap of $o_1$ and $o_2$ at $h_2$ is shown as a red oval on the left side of the DAG in Figure 1. In turn we can compute $\mathit{overlap}(o_1, o_3) = 0$. Peng and Reggia [31] define a pathognomonic effect as an observation with a single cause. Thus, whenever a pathognomonic symptom is involved, we do not compute an overlap relation. By collecting these measures for any pair of hypotheses or effects, we can compute a value over the entire model.

### 3.2.3 Independent Diagnosis Subproblem

Whenever there exist several subproblems in our theory we refer to them as independent diagnosis subproblems. If several subproblems exist, the graphs representing the model are disconnected and each independent diagnosis subproblem itself is a connected subgraph. In case all effects are pathognomonic, then each cause-effect relation represents its own independent diagnosis subproblem and thus we can observe that the model is orthogonal. Imagine the clause $h_2 \rightarrow o_2$ missing from the theory of Example 1. In this case we would have two independent diagnosis subproblems, namely one including $h_1, h_2$ and $o_1$ and the other one consisting of $h_3, o_2$ and $o_3$. As an additional measure to the number of subproblems we further compute the average size over all independent diagnosis subproblems in case several exist.

### 3.2.4 Path Length

Another measure of connectedness within the model is the minimal path length between any two nodes on the hypergraph. In particular, we measure the length of the minimal path between nodes representing hypotheses, thus we compute the minimal number of hyperedges to be traversed between each pair of hypothesis vertices. Note that for a single model there are possibly several hypergraphs depending on the number of independent diagnosis subproblems, thus we disregard paths between nodes belonging to different subproblems. Considering the hypergraph in Figure 1, we can observe $\mathit{path}(h_1, h_2) = 2$.

### 3.2.5 Clustering Coefficient

The clustering coefficient is a known measure of node clusters within a graph. It is evident that we cannot compute a clustering coefficient from the graph representations used so far, i.e. the DAG, bipartite graph and hypergraph, due to the two disjoint node classes. Therefore, we transform the bipartite graph by projection [18]. In particular, we remove all nodes corresponding to manifestations and link two cause vertices $v_{h_i}$ and $v_{h_k}$ whenever they imply the same effect, i.e. $\mathit{effects}(h_i) \cap \mathit{effects}(h_k) \neq \emptyset$. Based on the resulting undirected graph featuring only the nodes corresponding to hypotheses, we compute for each node the local clustering coefficient as $c = \frac{2n}{k_i(k_i-1)}$, where $n$ is the number of neighbors of the node and $k_i$ the number of edges between the neighbors of $n$. While in network analysis the projection of bipartite graphs results in coefficients differentiating from typical one-mode networks, this does not pose an issue in our case as we are solely interested in the models in our problem space. Thus, the clustering coefficient provides for our models another measure of covering between hypotheses.

#### 3.2.6 Kolmogorov Complexity

A simple encoding-based measure on a graph is its Kolmogorov complexity, which defines a value equal to the length of the word necessary to encode the graph. A straightforward manner in this context is to compute the complexity based on the adjacency matrix of the undirected graph [25].

#### 3.2.7 Observation Dependent Metrics

Since not only the topology of the model is of interest, but also the structure of the current diagnosis problem, we measure the indegree and the overlap among the elements of $Obs$ as well as the number of diagnosis subproblems involving variables of $Obs$, in case several exist.

## 4 Meta-Approach

Algorithm selection aims at identifying the most appropriate method out of a portfolio of techniques for a given problem instance in regard to its performance [35]. Performance in this context is most commonly associated with the computation time but could also refer to accuracy or simplicity. The model as described by Rice [35] advocates for the use of features inherent to the problems within the problem space in order to accurately map a new sample to its most effective or efficient algorithm. This mapping is based on empirical performance data on representative samples of the approaches present in the algorithm space [17]. On basis of the features extracted and the execution records, a predictor can be trained which can determine aspects of the problem influencing the performance of an algorithm. Thereby each problem can be described by a set of attributes which together with execution data allows a predictor to forecast the most valuable algorithm on an instance. Machine learning has been identified as a feasible approach to use as a prediction tool.

Generally, there are two possible objectives; either one algorithm of the portfolio is to be selected based on a single predictor or for each approach within the portfolio the performance metric should be determined as a basis of the selection. The latter requires a distinct empirical hardness model for each method within the portfolio and thus whenever the algorithm for a new instance is to be selected, for each approach a prediction has to be made [13, 17]. SATzilla [38] is an example of such a portfolio approach within the domain of SAT solvers. For our meta-technique, however, we consider the first variant, where we train a single classifier for all abductive reasoning methods to select a single approach for execution.

We consider a 1-of $n$ portfolio [38], where there are $n$ algorithms to choose from but only one is selected and executed to solve the diagnosis problem. Within the context of diagnosis our meta-approach works the following way: As mentioned the foundation of model-based diagnosis is a description of the system to be diagnosed. Thus, the majority of the features can be computed offline on the diagnosis models present. Further, within this phase the empirical data on computation times of the various abductive reasoning approaches can be collected and on basis of the metrics and the runtime information a machine learning classifier is trained. Whenever the diagnosis process is triggered by a detected anomaly, we retrieve our previously learned machine learning classifier as well as the offline determined metrics of the diagnosis model. Algorithm 1 describes the online portion of the meta-approach, which is executed whenever new diagnoses are to be computed. Online we have to collect the current PHCAP's instance-based features such as $|Obs|$ or the number of independent diagnosis subproblems comprising the current observations. Based on the online and offline generated attributes we supply the feature vector $\phi$ with the measurements of the current diagnosis problem. By providing all features to the machine learning algorithm, we in turn retrieve a predicted best abduction method out of our portfolio for this specific scenario based on the trained classifier and the instance's features. Subsequently, we can instantiate the diagnosis engine with the corresponding abduction method as well as diagnosis problem and compute the set of abductive explanations, i.e. $\Delta - Set$.

In the remainder of this section we describe first our portfolio which currently includes five abductive diagnosis methods and second we list the metrics we have used within the meta-approach.

### 4.1 Portfolio

We employ a 1-of 5 portfolio, i.e. we select one approach from the static algorithm space containing five methods which can be utilized for abductive reasoning based on a propositional logic model. For each technique, we give a brief description of the underlying notion. In particular, we utilize an Assumption-Based Truth Maintenance Systems (ATMS) [4, 5] as a general abduction engine for propositional Horn clauses [19]. Besides the ATMS our portfolio holds various hitting set algorithms, which are capable of computing minimal diagnoses as shown in Section 2.2. Thus, we simply compute for each $o_i \in Obs$ the set $causes(o_i)$, store them in the set $S$ and derive the minimal hitting sets for $S$. The hitting set routines we included in our meta-approach are the following: Binary Hitting Set Tree (BHSTree) [21], HS-DAG [34, 10], HST [36], and Berge's algorithm [2, 8].

#### 4.1.1 ATMS

The ATMS operates on a graph representation of the logical theory, where propositional variables are represented as nodes and the relations within the theory determine the directed edges. By utilizing a label for each node, the ATMS determines the subset minimal set of hypotheses implying each vertex and thereby allows to directly record abductive explanations. Furthermore, by recording contradictions it retains consistency. In order to generate the diagnoses for a given PHCAP, a clause is added such that $o_1 \wedge o_2 \wedge \ldots \wedge o_n \rightarrow obs$, where $o_1, o_2, \ldots, o_n \in Obs$ and $obs \notin A$. The label of $obs$ then contains the solution to the PHCAP.

#### 4.1.2 HS-DAG

Reiter's [34] minimal hitting set approach exploits the structure of a tree. To compute the hitting sets an initial set out of $S$ is the dedicated root node. The tree is then iteratively extended in a breadth first manner, where each node $n$ is labeled by a set $s \in S$ in case $s$ is disjoint to the set of edge labels of the current path. If this is the case an outgoing edge $h(n)$ is generated for each $\sigma \in s$ and after all $s \in S$ have been processed each leaf represents a hitting set. To ensure minimality various techniques on pruning the tree have been developed. Some inadequacies of Reiter's algorithm were corrected by Greiner et al. [10] and they further devised their HS-DAG as a version of Reiter's approach performed on a DAG instead of a tree.

#### 4.1.3 HST

The HST variant of HS-DAG operates on a tree instead of a graph and avoids the construction of unnecessary nodes and costly subset

**Algorithm 1** MetAB

**procedure** METAB ($A, Hyp, Th, Obs$)
  $m \leftarrow$ `retrieveClassifier()` ▷ Retrieves trained model
  $\phi_{offline} \leftarrow$ `retrieveMetrics`$(A, Hyp, Th)$ ▷ Retrieves the previously computed model metrics
  $\phi_{online} \leftarrow$ `computeMetrics`$(A, Hyp, Th, Obs)$ ▷ Computes the instance-based features
  $\phi = \phi_{offline} \cup \phi_{online}$
  $algorithm \leftarrow$ `predict`$(\phi, m)$ ▷ Forecasts the best performing algorithm for the diagnosis problem
  $\Delta - Set \leftarrow$ `diagnose`$(algorithm, A, Hyp, Th, Obs)$ ▷ Computes diagnoses based on the predicted algorithm for the PHCAP
  **return** $\Delta - Set$
**end procedure**

checks [36]. Based on an ordered list of the elements within $S$ the algorithm limits the number of outgoing edges for each node to a specific range within the ordered list. By checking the current path and the tree already constructed, the algorithm decides to whether a node has to be generated or the corresponding hitting set will be constructed later during the computation.

#### *4.1.4 BHSTree*

Lin and Jiang [21] propose the Binary Hitting Set Tree. First the tree is constructed by splitting input sets on particular elements and recursively adapting the sets and building the tree. During the bottom up traversal the hitting sets are constructed by merging the data of the child nodes. The minimization is performed by a minimization function $\mu$, which is not specified in detail in their paper.

#### *4.1.5 Berge's Algorithm*

This minimal hitting set algorithm, sometimes referred to as Berge's algorithm, uses an intuitive notion to compute hitting sets [8, 28]. By definition a hitting set intersected with any set of $S$ is not empty, i.e. each set of $S$ has to contribute to each hitting set. In Berge's algorithm the minimal hitting sets are constructed and modified incrementally. Initially, the set of hitting sets $H$ is empty. Whenever a new $s \in S$ is to consider, each $\eta \in H$ is checked whether $s \cap \eta = \emptyset$. In case it is not, i.e. $\eta$ already hits $s$, $\eta$ remains unchanged, otherwise it is removed from $H$ and for each $\sigma \in s$ a new set is created containing $\eta$ and $\sigma$. By checking whether subsets are present within $H$ the algorithm ensures to derive minimal hitting sets.

### 4.2 Features

In Section 3 we discussed the metrics we extract from our logical problem instances. Here we simply list the properties we compute in our meta-approach.

1. Logic model specific
   - Number of hypotheses
   - Number of effects
   - Number of causal relations, i.e. clauses in the theory
   - Number of independent diagnosis subproblems
   - Average size of independent diagnosis subproblems
2. DAG
   - Outdegree of hypothesis nodes (maximum, average, standard deviation)
   - Indegree of effect nodes (maximum, average, standard deviation)
   - Covering (maximum, average, standard deviation)
   - Overlap (maximum, average, standard deviation)
3. Undirected graph
   - Kolmogorov complexity based on adjacency matrix
   - Local clustering coefficient (maximum, average, standard deviation)[3]
4. Hypergraph
   - Path length (maximum, average, standard deviation)[4]
5. Instance specific/Observation dependent
   - Number of observations
   - Indegree current observation nodes (maximum, average, standard deviation)
   - Overlap current observation (maximum, average, standard deviation)
   - Number of independent diagnosis subproblems including current observations

While Hutter et al. [14] state that the feature extraction method should be efficient, in our framework only the computation of a subset of these attributes has to be performed online, namely the computation of the instance specific metrics. The creation of the diagnosis models, the computation of the basic model features, i.e. feature sets *1* to *4*, and the training of the classifier are performed offline. Thus, online the observation specific metrics have to be extracted, i.e. feature set *5*, the algorithm appropriate for the problem instance has to be predicted and the diagnoses have to be calculated.

### 4.3 Empirical Evaluation

In this section we evaluate the feasibility of our meta-approach. On the one hand we assess the quality of the model properties to train a machine learning classifier capable of predicting the most efficient algorithm out of the portfolio in regard to its runtime for a specific PHCAP instance. On the other hand, we examine the efficiency of the meta-approach overall in comparison to the abductive reasoning algorithms in the portfolio.

Our meta-approach itself is implemented in Java as well as the ATMS engine, BHSTree[5] and Berge's algorithm. For the remaining hitting set algorithms, i.e. HS-DAG and HST, we exploit PyMBD[6] [33], a publicly available python library of minimal hitting set algorithms. To create a predictor based on the features we utilize the

[3] Based on the projection of the undirected graph only containing hypothesis nodes.
[4] Path length between hypothesis vertices.
[5] http://www.ist.tugraz.at/modremas/index.html
[6] http://modiaforted.ist.tugraz.at/downloads/pymbd.zip

Waikato Environment for Knowledge Analysis (WEKA) library [11] which provides a vast variety of classification algorithms. The experiment was conducted on a Mac Pro (Late 2013) with a 2.7 GHz 12-Core Intel Xeon ES processor and 64GB of RAM running OS X 10.10.5.

### *4.3.1 Data*

In order to evaluate the meta-algorithm we generated a test suite of artificial samples. Note here that while there are real-world samples based on FMEAs, we do not incorporate them in this evaluation as the size of the practical models is rather limited. The synthetic samples obtained differ in their number of hypotheses, effects, connections and the covering and overlap between causes and manifestations, respectively. Conjunctions of causes and disjunctions of manifestations were created randomly within the samples and it was ensured that instances with various independent diagnosis problems were present in the test suite. A total of 195 samples were created with a varying number of hypotheses ($12 \leq |Hyp| \leq 3120$), effects ($1 \leq |M| \leq 5000$), clauses ($12 \leq |Th| \leq 5100$) and ($1 \leq |Obs| \leq 30$). With each experiment run we collected the 31 metrics described in the previous section to build our feature vector[7] for the classification. For each problem instance we executed our abductive reasoning algorithms and recorded the most efficient algorithm on each problem instance. To evaluate the classification based on the features we randomly split the test suite into a training set comprising 80% of the data and a test set holding 20% of the examples. We standardized our data before performing any classification.

### *4.3.2 Results*

Before selecting the classification method, we performed cross validation on several classification algorithms available in WEKA on the training data. Based on the accuracy obtained we decided to use WEKA's implementation of a general algorithm for locally weighted learning *LWL*. While within this experiment we did not compare different classifiers besides the initial informal evaluation, we do believe that examining various machine learning algorithms will be of interest in future research on this matter. *LWL* performs prediction by building a local model based on the weighted neighborhood data of the attribute of interest. In our case the this attribute is nominal and simply corresponds to the algorithms name. In regard to the parameters we utilized a brute force nearest neighbor search algorithm, included all neighbors in the linear weighting function and an entropy based classifier.

The classification utilizing *LWL* and based on the metrics reaches a satisfactory success rate of 71.79% (MAE=0.22, RMSE=0.31) correctly predicted instances, i.e. the selected algorithm was in fact the most efficient on the problem, on the test set. The confusion matrix in Table 2 shows the number of correctly and wrongly classified instances. From the contingency table it is apparent that within the test set Berge's algorithm was the dominant approach, thus, our predictor classified all but one instance as Berge's algorithm. A known limitation of this type of algorithm, selection where simply a single approach is chosen and executed on the instance, is that in case the prediction is incorrect the meta-approach might be rather inefficient [17]. It is to note that in case our classifier chose a slower approach, the selected algorithm was the second fastest within the portfolio. On the test set the meta-approach was on average 1.57% slower than an optimal algorithm selection (MetAB_Opt), i.e. the predictor would classify every instance correctly, would have been.

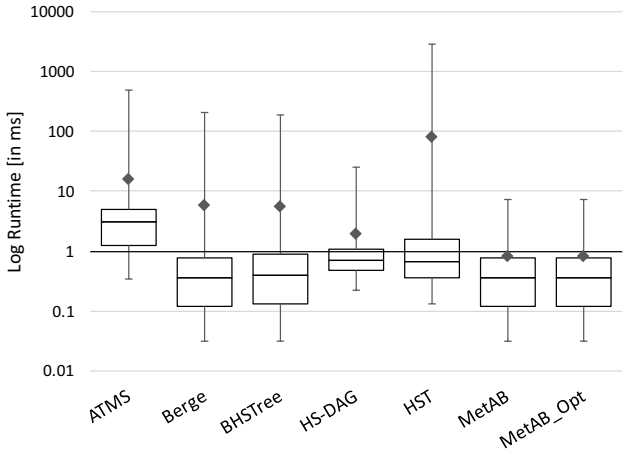



**Figure 2**: Underlying statistical distributions of the log runtimes for the test set.

We explored WEKA's attribute selection in order to determine whether we could remove certain features while achieving the same prediction accuracy. Utilizing the meta-classifier with the *LWL* classifier, we examined various selection approaches on the training data and could diminish the set of features significantly from 32 to around four[8]. The number and composition of the reduced attribute set depends highly on the performed selection process. For example utilizing the *OneR* evaluator and limiting the number of features of the reduced set to three, we receive the attribute set consisting of the number of current diagnosis problems , the number of observations and number of effects of the entire model. Attribute selection on grounds of the information gain results in number of current diagnosis problems, the number of observations, the average path length on the hypergraph and its standard deviation. Utilizing the SVM-based reduction, we receive the number of observations, the standard deviation of the indegree of the nodes representing the observations, the average current covering relation and its standard deviation as well as as the average of the covering relation over the entire model. As can be seen the size of $Obs$ plays an essential role in predicting the preferable algorithm. In regard to the remaining selected properties they provide information on the PHCAP, i.e. the current observations, and various metrics on how hypotheses are connected through effects. An in depth analysis of a reduced feature reduction would be a topic of further research.

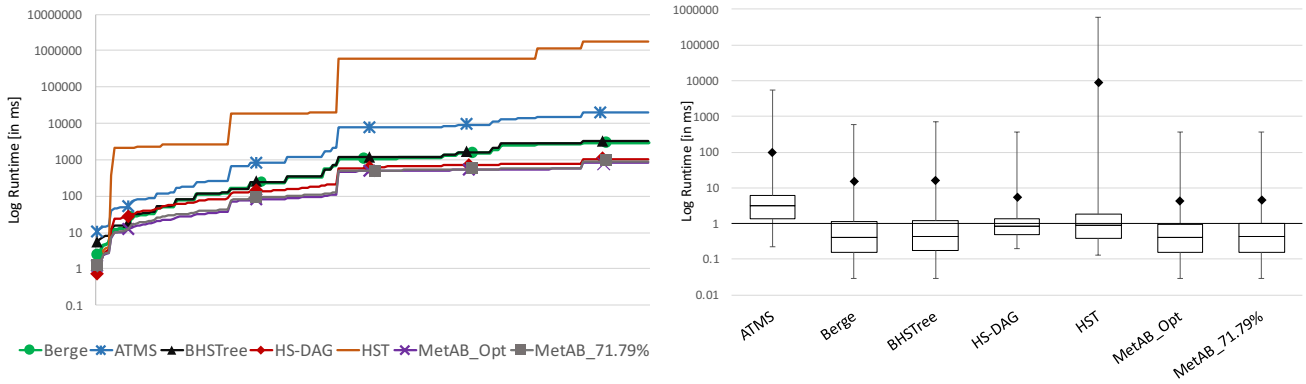
**Table 2**: Confusion Matrix for the artificial test set. The rows represent the actual number of instances within the category, while the columns show the predicted outcome.

| | **Berge** | **HS-DAG** | **ATMS** | **BHSTree** | **HST** | **Total** |
|---|---|---|---|---|---|---|
| **Berge** | 27 | 0 | 0 | 0 | 0 | 27 |
| **HS-DAG** | 1 | 0 | 0 | 0 | 0 | 0 |
| **ATMS** | 2 | 0 | 0 | 0 | 0 | 2 |
| **BHSTree** | 8 | 0 | 0 | 0 | 0 | 8 |
| **HST** | 0 | 0 | 0 | 0 | 1 | 1 |
| **Total** | 38 | 0 | 0 | 0 | 1 | 39 |

Figure 2 shows the distribution of the log runtime data for the

[7] Note that the feature vector itself holds 32 values, as one is the nominal value corresponding to the algorithm's name. This last feature is to be predicted.

[8] The feature of the most efficient algorithm remains of course within the set after selection and therefore accounts for one feature in the reduced vector.




(a) Cumulative runtimes over the entire sample set.

(b) Underlying statistical distributions of the log runtimes for the entire sample set.

**Figure 3**: Runtime plots over the entire sample set.

test set. In case of our artificial examples the meta-approach MetAB (M=0.83 ms, SD =1.54 ms) performs well, i.e. is on average the most efficient. On average its is 99% faster than HST (M=82.78 ms, SD=455.65 ms), 94.86% more efficient than the ATMS (M=16.16 ms, SD=76.76 ms), around 85% faster than BHSTree (M=5.57 ms, SD=30.43 ms) and Berge's algorithm (M=5.9 ms, SD=32.72 ms), and still computes diagnoses around 58.27% faster than the HS-DAG (M=1.99 ms, SD=4.5 ms).

The overall runtime for the meta algorithm is determined by (1) the computation of the online metrics, (2) the time it takes to create the feature vector, supply it to the classifier and predicting the best algorithm and (3) the diagnosis time of the suggested abduction procedure. In regard to the feasibility of the meta-approach overall in comparison to other abductive diagnosis methods, we like to refer back to a particular characteristic of model-based diagnosis, namely the availability of a system description offline. As the model has to be present before the computation of the diagnoses, it allows us to extract most of the metrics utilized in the algorithm selection offline. Thus, the online computation of the features which are inherent to the specific instance of the PHCAP is negligible ($< 0.1$ ms). The prediction of the algorithm for a single instance for the diagnosis models we investigated was insignificant. The third factor, the diagnosis time, is much dependent on the predictive capabilities of the classifier.

A premature analysis of the results of the test data would suggest that applying Berge's method to every instance would yield the optimal runtime for most problems. However, from the cumulative log runtimes Figure 3a we can observe that based on the entire set of problems, i.e. test and training, Berge is not the most efficient approach as on several instances its computation time is notably larger than of other algorithms. On the entire sample we observe that only considering the algorithms from the portfolio, HS-DAG is on average the best performing approach (M=5.5 ms, SD=32.31 ms) followed by Berge's algorithm (M=15.62 ms, SD=68.39 ms) and BHSTree (M=16.88 ms,SD=76.6 ms), while the ATMS (M=101.37ms, SD=563.85 ms) still outperforms HST (M=8968.04ms, SD=70873.65ms). Furthermore, based on the prediction accuracy on the test set, we have created a mock meta-approach (MetAB_71.79%) with 71.79% accuracy as we have experienced on the test data. Thus, in 71.79% of the samples we recorded for this approach the optimal time and for the remaining 28.21% the second fastest time. We choose the instances with the slower runtimes randomly. This mock meta-approach would still outperform, the other algorithms in the portfolio. In Figure 3b we have depicted the distribution of the log runtimes of the various approaches.

## 5 Conclusion

Even though abduction is an intuitive approach to diagnosis, its computational complexity remains a disadvantage. Since the complexity is inherent to the underlying diagnosis problem, we investigate algorithm selection as a method to predict the most efficient abduction approach for a particular instance. An essential part within algorithm selection is the exploration of characteristics of the problems which contribute to the computational effort. Hence, we consider various metrics characterizing the type of models generated from failure assessments available in practice. An advantage of this approach in the context of abductive diagnosis is that the majority of these features can be collected offline. Based on the attributes we form a feature vector for a machine learning classifier as part of a meta-approach. The empirical evaluation showed that the extracted properties of the instances allow to determine the "best" abduction method. Even in cases where the classification is incorrect, the approach selects the second most efficient algorithm and thus overall outperforms the other diagnosis methods. Therefore, we believe that this meta approach is a feasible alternative to continuously using a single abduction procedure. Despite the satisfactory classification results, we plan on further extending the set of problem metrics, explore the capabilities of various machine learning classifiers as well as determine different attribute combinations yielding a more accurate classification result.

## ACKNOWLEDGEMENTS

The work presented in this paper has been supported by the FFG project Applied Model Based Reasoning (AMOR) under grant 842407. We would further like to express our gratitude to our industrial partner, Uptime Engineering GmbH.

## REFERENCES

[1] Joachim Baumeister and Dietmar Seipel, 'Diagnostic reasoning with multilevel set-covering models', Technical report, DTIC Document, (2002).

[2] Claude Berge. Hypergraphs, volume 45 of north-holland mathematical library, 1989.

[3] Luca Console, Daniele Theseider Dupre, and Pietro Torasso, 'On the Relationship Between Abduction and Deduction', *Journal of Logic and Computation*, **1**(5), 661–690, (1991).

[4] Johan de Kleer, 'An assumption-based TMS', *Artificial Intelligence*, **28**, 127–162, (1986).

[5] Johan de Kleer, 'Problem solving with the ATMS', *Artificial Intelligence*, **28**(2), 197–224, (1986).

[6] Marc Denecker and Antonis Kakas, 'Abduction in logic programming', in *Computational Logic: Logic Programming and Beyond*, 402–436, Springer, (2002).

[7] Thomas Eiter and Georg Gottlob, 'The complexity of logic-based abduction', *Journal of the ACM (JACM)*, **42**(1), 3–42, (1995).

[8] Thomas Eiter, Kazuhisa Makino, and Georg Gottlob, 'Computational aspects of monotone dualization: A brief survey', *Discrete Applied Mathematics*, **156**(11), 2035–2049, (2008).

[9] Gerhard Friedrich, Georg Gottlob, and Wolfgang Nejdl, 'Hypothesis classification, abductive diagnosis and therapy', in *Expert Systems in Engineering Principles and Applications*, 69–78, Springer, (1990).

[10] Russell Greiner, Barbara A Smith, and Ralph W Wilkerson, 'A correction to the algorithm in reiter's theory of diagnosis', *Artificial Intelligence*, **41**(1), 79–88, (1989).

[11] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten, 'The weka data mining software: an update', *ACM SIGKDD explorations newsletter*, **11**(1), 10–18, (2009).

[12] Peter G. Hawkins and Davis J. Woollons, 'Failure modes and effects analysis of complex engineering systems using functional models', *Artificial Intelligence in Engineering*, **12**, 375–397, (1998).

[13] Frank Hutter, Youssef Hamadi, Holger H Hoos, and Kevin Leyton-Brown, 'Performance prediction and automated tuning of randomized and parametric algorithms', in *Principles and Practice of Constraint Programming-CP 2006*, 213–228, Springer, (2006).

[14] Frank Hutter, Lin Xu, Holger H Hoos, and Kevin Leyton-Brown, 'Algorithm runtime prediction: Methods & evaluation', *Artificial Intelligence*, **206**, 79–111, (2014).

[15] Antonis C. Kakas, Robert A. Kowalski, and Francesca Toni, 'Abductive logic programming', *Journal of logic and computation*, **2**(6), 719–770, (1992).

[16] Richard M Karp, *Reducibility among combinatorial problems*, Springer, 1972.

[17] Lars Kotthoff, 'Algorithm selection for combinatorial search problems: A survey', *arXiv preprint arXiv:1210.7959*, (2012).

[18] Matthieu Latapy, Clémence Magnien, and Nathalie Del Vecchio, 'Basic notions for the analysis of large two-mode networks', *Social networks*, **30**(1), 31–48, (2008).

[19] Hector J Levesque, 'A knowledge-level account of abduction.', in *IJCAI*, pp. 1061–1067, (1989).

[20] Kevin Leyton-Brown, Eugene Nudelman, Galen Andrew, Jim McFadden, and Yoav Shoham, 'A portfolio approach to algorithm selection', in *IJCAI*, volume 1543, p. 2003, (2003).

[21] Li Lin and Yunfei Jiang, 'The computation of hitting sets: Review and new algorithms', *Information Processing Letters*, **86**(4), 177–184, (2003).

[22] Pierre Marquis, 'Consequence finding algorithms', in *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, 41–145, Springer, (2000).

[23] Sheila A McIlraith, 'Logic-based abductive inference', *Knowledge Systems Laboratory, Technical Report KSL-98-19*, (1998).

[24] Michael Morak, Nysret Musliu, Reinhard Pichler, Stefan Rümmele, and Stefan Woltran, 'Evaluating tree-decomposition based algorithms for answer set programming', in *Learning and Intelligent Optimization*, 130–144, Springer, (2012).

[25] Abbe Mowshowitz and Matthias Dehmer, 'Entropy and the complexity of graphs revisited', *Entropy*, **14**(3), 559–570, (2012).

[26] Nysret Musliu and Martin Schwengerer, 'Algorithm selection for the graph coloring problem', in *Learning and Intelligent Optimization*, 389–403, Springer, (2013).

[27] Gustav Nordh and Bruno Zanuttini, 'What makes propositional abduction tractable', *Artificial Intelligence*, **172**(10), 1245–1284, (2008).

[28] Mattias Nyberg, 'A generalized minimal hitting-set algorithm to handle diagnosis with behavioral modes', *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, **41**(1), 137–148, (2011).

[29] Ekaterina Ovchinnikova, Niloofar Montazeri, Theodore Alexandrov, Jerry R Hobbs, Michael C McCord, and Rutu Mulkar-Mehta, 'Abductive reasoning with a large knowledge base for discourse processing', in *Computing Meaning*, 107–127, Springer, (2014).

[30] Yun Peng and James A Reggia, 'Plausibility of diagnostic hypotheses: The nature of simplicity.', in *AAAI*, volume 86, pp. 140–145, (1986).

[31] Yun Peng and James A Reggia, *Abductive inference models for diagnostic problem-solving*, Springer, 1990.

[32] David Poole and Keiji Kanazawa, 'A decision-theoretic abductive basis for planning', in *AAAI Spr. Symp. on Decision-Theoretic Planning*, (1994).

[33] Thomas Quaritsch and Ingo Pill, 'Pymbd: A library of mbd algorithms and a light-weight evaluation platform', *Proceedings of Dx-2014*, (2014).

[34] Raymond Reiter, 'A theory of diagnosis from first principles', *Artificial Intelligence*, **32**(1), 57–95, (1987).

[35] John R Rice, 'The algorithm selection problem', (1975).

[36] Franz Wotawa, 'A variant of reiter's hitting-set algorithm', *Information Processing Letters*, **79**(1), 45–51, (2001).

[37] Franz Wotawa, 'Failure mode and effect analysis for abductive diagnosis', in *Proceedings of the International Workshop on Defeasible and Ampliative Reasoning (DARe-14)*, volume 1212. CEUR Workshop Proceedings, ISSN 1613-0073, (2014). http://ceur-ws.org/Vol-1212/.

[38] Lin Xu, Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown, 'Satzilla: portfolio-based algorithm selection for sat', *Journal of Artificial Intelligence Research*, 565–606, (2008).

[39] Yifan Yang, Jamal Atif, and Isabelle Bloch, 'Abductive reasoning using tableau methods for high-level image interpretation', in *KI 2015: Advances in Artificial Intelligence*, 356–365, Springer, (2015).