Applied Intelligence (2018) 48:3976–3994
https://doi.org/10.1007/s10489-018-1171-9

Applying algorithm selection to abductive diagnostic reasoning

Roxane Koitz-Hristov[1] · **Franz Wotawa**[1]

Published online: 10 May 2018

Abstract

The complexity of technical systems requires increasingly advanced fault diagnosis methods to ensure safety and reliability during operation. Particularly in domains where maintenance constitutes an extensive portion of the entire operation cost, efficient and effective failure identification holds the potential to provide large economic value. Abduction offers an intuitive concept for diagnostic reasoning relying on the notion of logical entailment. Nevertheless, abductive reasoning is an intractable problem and computing solutions for instances of reasonable size and complexity persists to pose a challenge. In this paper, we investigate algorithm selection as a mechanism to predict the "best" performing technique for a specific abduction scenario within the framework of model-based diagnosis. Based on a set of structural attributes extracted from the system models, our meta-approach trains a machine learning classifier that forecasts the most runtime efficient abduction technique given a new diagnosis problem. To assess the predictor's selection capabilities and the suitability of the meta-approach in general, we conducted an empirical analysis featuring seven abductive reasoning approaches. The results obtained indicate that applying algorithm selection is competitive in comparison to always choosing a single abductive reasoning method.

Keywords Abductive reasoning · Model-based diagnosis · Algorithm selection

1 Introduction

Abduction is reasoning toward the "best" explanations for a set of encountered observations. While the most prevalent application area is diagnosis, where abduction is an intuitive methodology for deriving root causes, its usage ranges from test case generation [41] over text interpretation [49] to human behavior interpretation [14]. In regard to fault localization, model-based approaches have been developed for decades. These techniques aim at identifying failures from a description of the system under diagnosis and the detected abnormal behavior [8, 53]. The traditional consistency-based method exploits a formalization of the correct behavior that in conjunction with a symptom leads to inconsistencies. These inconsistencies are then used to derive explanations. In contrast, the abductive model-based diagnosis version depends on logical entailment, i.e., given the system description a diagnosis has to entail the observations. Additional restrictions, such as minimality, characterize the admissible solutions. Originally, abductive model-based approaches consider solely representations of the faulty system behavior. Such failure knowledge can be expressed as Horn theories [5].

All model-based techniques depend on the quality of the underlying representation to adequately capture the system's behavior and structure. Besides being expressive enough to formalize the artifact's characteristics, the model should allow for efficient diagnostic reasoning to be suitable in a practical application. Ideally, the system description allows additional support of the decision making process by (1) including probabilities to allow a prioritization, (2) by enabling probe selection recommendations to refine the diagnoses, and/or (3) by taking into account costs and other factors for suggesting repair or replacement activities [4]. Other approaches extend the model-based diagnosis framework with capabilities to handle uncertainties often encountered in practice [27, 38]. From a practical point of view, it

The work presented in this paper has been supported by the FFG project Applied Model Based Reasoning (AMOR) under grant 842407 and Steirische Wirtschaftsförderung project EXPERT.

✉ Roxane Koitz-Hristov
rkoitz@ist.tugraz.at

Franz Wotawa
wotawa@ist.tugraz.at

[1] Insitute for Software Technology, Graz University of Technology, Inffeldgasse 16b/II, Graz 8010, Austria

is essential to construct an appropriate system description as the quality of the entire diagnostic process relies on the model's suitability in regard to the domain; yet, generation of appropriate diagnostic models is often associated with a large initial effort. To facilitate the integration of diagnosis tools in existing work processes, research has focused on automatically extracting Horn diagnosis models from failure assessments used in practice, such as Failure Mode and Effect Analysis (FMEA) [31, 62]. While this method eliminates or reduces the manual model creation, the generated system description is only as good as its source.

Let us consider a simple example from the wind turbine domain, where maintenance costs account for a large portion of the life time expenditure of the installations. The gearbox is a vital component converting the rotor's low-speed power to high speed power for the generator. Hence, failures in the gearbox contribute to a great extend to system downtimes. Gearbox lubrication plays an essential role in retaining a healthy system state by protecting the gears and bearings from excessive wear and overheating. A failing oil filter promotes contamination, which in conjunction with excessive oil temperature affects the lubricant film of the subcomponents. Overheating of the oil is caused by damages to the oil cooler. An insufficient lubrication of the gearbox may also be rooted in a broken oil pump, which leads to a decrease in oil pressure and thus a reduced oil flow through the gearbox. Obviously, this type of information can easily be formalized as a set of propositional Horn clauses. Suppose an inadequate lubrication of the gearbox has been detected. The goal of diagnosis is to determine the root causes of the observed symptom. For this diagnosis problem there are two possible explanations: either the oil filter is broken and there are cracks and leaks in the cooling component or the oil pump has been damaged.

Abduction for propositional Horn theories is an NP-complete problem that grows exponentially in the size of the model [10]. Even though there are practical examples of NP-complete problems that can be solved quite efficiently [2], usually there is no universally "best" algorithm operating well on all problem instances. Choosing the computation method in regard to the particular example at hand can provide better performance results [34]. Algorithm selection aims at solving exactly this issue by identifying the "best performing" procedure for a specific instance [54]. The approach requires (1) a portfolio of algorithms to choose from, (2) empirical performance data of the algorithms on representative problems, and (3) a set of problem features that are used to get a notion of the difficulty of a problem instance [20]. To determine the most suitable routine for a distinct example, a predictor is constructed taking into account the empirical data and the feature vector of the representative problem cases [34, 57].

Given the complexity results of abduction and the successful application of the portfolio approach in the domain of SAT [64], graph coloring [46], and tree-decomposition [44], algorithm selection represents an interesting strategy to compute abductive diagnoses. Hence, we investigate algorithm selection as a means to efficiently compute explanations in the context of propositional Horn abduction. Horn models possess certain structural properties, which we utilize as features in order to classify diagnosis problems in regard to the most runtime efficient algorithm. We embedded this selection process within a meta-algorithm that generates the structural metrics for a given diagnosis problem, categorizes it on a previously trained predictor, and computes the diagnoses using the selected abduction approach. The foundations for this work have been laid previously; first, we have applied a portfolio approach to abduction based on bijunctive Horn models, i.e., theories where the clauses solely contain a single negative literal representing a cause and single positive literal representing a symptom [30, 32]. Hence, in this work we extend the underlying representation to Horn clauses. Second, in Koitz and Wotawa [33], we have compared the performance of consequence finding and proof-tree completion methods in the context of propositional Horn clause abduction. We rely on the empirical performance data we have obtained from this previous research as an input to construct the predictor for our algorithm selection approach, test the resulting selection accuracy, and determine the overall performance of the meta-approach in comparison to the other abduction procedures.

This paper is structured as follows. In the upcoming sections, we give an overview of related literature and describe the basic definitions of abductive model-based diagnosis for propositional Horn theories. Section 4 focuses on the attributes we can extract from the Horn models and details our meta-approach. Before concluding the paper, we provide an empirical analysis comparing the meta-approach to consistently choosing a single abduction method.

2 Related research

There are various methods for extracting abductive diagnoses, such as the parsimonious set-covering theory [52] or probabilistic approaches [50]. In logic-based abduction there are two formulations of the problem directly connected to the relation between diagnoses, observations, and the background knowledge: consequence finding [40] and proof-tree completion [42]. (1) In consequence finding explanations are derived deductively by re-formulating abduction as the search for logical consequences. Two well known consequence finding techniques are the Assumption-based Truth Maintenance System (ATMS) [6], deriving

consequences from propositional Horn theories, and Skipping Ordered Linear (SOL) resolution allowing to extract logical consequences of interest for full clausal first-order theories [23]. (2) Proof-tree completion rewrites the diagnosis problem in such a way that refutations correspond to solutions [42]. Hence, any conflict extracting method can be utilized, e.g., Minimal Unsatisfiable Subsets (MUSes) computation of logical formulae [37]. Many MUS enumeration algorithms refrain from computing the unsatisfiable cores directly, but exploit its hitting set dual Minimal Correction Subsets (MCSes). For instance, Liffiton and Sakallah [37] present the CAMUS algorithm utilizing this hitting set duality to produce MUSes by first computing all MCSes.

Abduction has also been introduced to logic programming, where additional integrity constraints and distinguished predicates restrict the admissible solution space [25]. Besides specific tools such as the $\mathcal{A}$-System [26], abductive logic programming (ALP) can be realized using Answer Set Programming (ASP) [56]. Recently, Saikko et al. [55] have proposed the derivation of one minimal-cost abductive explanations through implicit hitting set computation based on a combination of an Integer Programming and a SAT solver. An implementation of their iterative approach outperforms a state-of-the-art ASP solver with an encoding of propositional abduction. Ignatiev, Morgado, and Marques-Silva [22] report on an improvement of this method by deriving the hitting sets via a MaxSAT solver.

Even though there exist certain subsets of logics where abductive inference is tractable, abduction for the general case is hard for the second level of the polynomial hierarchy [10]. Algorithm selection has resulted in significant performance improvements on various AI problems [34]. First formalized by Rice [54], algorithm selection aims at identifying the "best performing" approach for a specific problem instance. To determine the most suitable routine for a distinct example, a predictor is constructed taking into account the empirical data and the feature vector of representative problem cases [57]. Instead of creating a single classifier to chose a method, Leyton-Brown et al. [36] train an empirical hardness model for each algorithm within the portfolio to forecast each technique's performance on the instance and execute the one predicted superior. While there are approaches that manually create models for the predictor [59], most of the time a machine learning classifier is exploited that categorizes a new problem instance at hand as one of the methods in the portfolio. Static portfolio approaches have a finite and fixed set of algorithms to chose from (though there exist approaches where the number of algorithms in the portfolio is based on the training data [63]), while a dynamic portfolio is built online and may be constructed of algorithmic blocks.

Generally, there are various strategies in the field of algorithm selection, for instance, in regard to the selection (a single algorithm used to solve the entire problem or interleaved/parallel execution of various approaches), the number of models built (a single one to predict the "best" method or one per approach in the portfolio), and the time of choosing the algorithm (at the beginning or several times during search). The interested reader is referred to Kotthoff [34], who provides a comprehensive overview of different algorithm selection techniques.

At the intersection of abduction and algorithm selection, there is the work by Guo and Hsu [16] . The authors propose algorithm selection in the context of deriving the most probable explanation (MPE) in probabilistic inference. Differencing from other work on algorithm selection, the authors first use classification to determine whether the problem is solvable and then either use clique-tree propagation to derive the exact solution or apply a second classifier to identify the "best" approximation procedure. While not directly usable for abductive reasoning, Malitsky et al. [39] apply the portfolio approach to enumerate MCSes, which can be exploited to derive explanations based on their hitting set duals [28]. Instead of choosing a single solver for the current problem instance, the authors use a technique that switches between various enumeration procedures multiple times. After a fixed timeout the next solver is used for computing the remaining solutions. The decision for the proceeding solver is computed on demand and to ensure already computed solutions are not derived again, blocking clauses are added whenever a new enumeration procedure is chosen.

3 Preliminaries

In essence, abductive inference allows the derivation of plausible explanations for a given set of observations. Logic-based abduction provides an intuitive notion of this type of reasoning: a set of abducible propositions is an explanation or diagnosis for an observed symptom in case the observation is a consistent logical consequence of the diagnosis and background theory. A conclusion ϕ is said to be a logical consequence of a set of premises ψ, if and only if for any interpretation in which ψ holds ϕ is also true. We write this relation as $\psi \models \phi$ and say ψ entails ϕ. An abductive diagnosis or explanation is composed of a set of abducible propositions taken from a sub-vocabulary of the representation language. In the context of fault identification, these propositions are usually abnormality assumptions about components, while in medical diagnosis correspond to diseases.

Similar to Friedrich, Gottlob, and Nejdl [11] we focus on Horn abduction and define a propositional knowledge

base (KB) that formalizes a propositional Horn theory *Th* over a finite set of propositional variables *A*. A Horn clause is defined as a disjunction of literals featuring at most one positive literal and can be described by a rule, i.e., $\{\neg a_1, \ldots, \neg a_n, a_{n+1}\}$ can be written as $a_1 \wedge \ldots \wedge a_n \rightarrow a_{n+1}$. A clause without any negated propositions, i.e., $\{p\}$, characterizes a fact. Many diagnostic reasoning engines restrict the model to Horn clause sentences, which are usually expressive enough in the context of fault identification to convey the necessary relations [5]. As the complexity of logic-based abduction is connected to the characteristics of the underlying model [3, 48] restricting the input language to Horn clauses is beneficial from a computation point of view. In particular, while abduction for general propositional theories is located in the second level of the polynomial hierarchy, for Horn clauses the complexity is mitigated to the first level [10]. A model or knowledge base is formally defined as a tuple *(A,Hyp,Th)*, where the set *Hyp* contains all hypotheses, i.e., abducible variables.

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* the set of Horn clause sentences over *A*.

A Propositional Horn Clause Abduction Problem (PHCAP) is characterized by a KB and a set of observations for which root causes are to be derived. In our definition, observations may only be a conjunction of propositions and not an arbitrary logical sentence.

Abduction within the logic-based framework is defined via derivability and consistency, i.e., the observed manifestation must be derivable from *Th* augmented with the diagnosis Δ, while $\Delta \cup Th$ is consistent. In addition, a solution to a PHCAP must consist of propositions from the set of hypotheses.

Definition 2 (**Propositional Horn Clause Abduction Problem (PHCAP)**) Given a KB(*A,Hyp,Th*) and a set of observations $Obs \subseteq A$, 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 iff $\Delta \cup Th \models Obs$ and $\Delta \cup Th \not\models \bot$.

Since generally the goal of abduction is to derive the "best" explanations, some preference criteria must be present to characterize the notion of optimality. A common principle is to only consider subset-minimal diagnoses. Especially in practical applications explanation supersets provide no additional information useful in a real-world context. Thus, we define Δ-Set as the set containing all parsimonious explanations.

Definition 4 (**Parsimonious Diagnosis**) A solution Δ is parsimonious or minimal iff no set $\Delta' \subset \Delta$ is a solution.

Example 1 In the introduction we have discussed the example of gearbox lubrication in the industrial wind turbine domain, where damages to the cooling component lead to oil overheating. This overheating in conjunction with a broken oil filter alters the lubricant's condition and hence negatively affects gearbox lubrication. Moreover, a malfunctioning pump triggers a reduced pressure and subsequently influences the dispensation of fluid throughout the system. These circumstances can be easily described by the following KB:

$$A = \left\{ \begin{array}{c} Broken\_filter, Cooler\_leaks, \\ Cooler\_cracks, Damaged\_pump, reduced\_pressure, poor\_lubrication, overheating \end{array} \right\}$$

$$Hyp = \left\{ Broken\_filter, Cooler\_leaks, Cooler\_cracks, Damaged\_pump, \right\}$$

$$Th = \left\{ \begin{array}{c} Broken\_filter \wedge overheating \rightarrow poor\_lubrication, \\ Cooler\_leaks \wedge Cooler\_cracks \rightarrow overheating, \\ Damaged\_pump \rightarrow reduced\_pressure, reduced\_pressure \rightarrow poor\_lubrication \end{array} \right\}$$

Presume we determine there is an insufficient greasing of the gearbox, i.e., $Obs = \{poor\_lubrication\}$. Given the theory Th and Obs, we can determine two parsimonious explanations; (1) either leaks and cracks in the cooler cause the oil to overheat in addition to a filter failure or the oil pump is damaged, i.e., Δ-Set $= \{\{Broken\_filter, Cooler\_leaks, Cooler\_cracks\}, \{Damaged\_pump\}\}$.

By rewriting the relations between the theory, a diagnosis, and the observations, we can reformulate the abduction problem in two different fashions [33]: proof-tree completion [42] and consequence finding [40]. In the former, abduction is described as the search for a refutation proof consisting of abducible propositions, i.e., $\Delta \cup Th \cup \{\neg Obs\} \models \bot$, where $\{\neg Obs\}$ is the disjunction containing a negation of each observation, i.e., $\bigvee_{o_i \in Obs} \neg o_i$. To extract explanations, the derived conflicts have to be propositions from Hyp and again no inconsistency may arise from the solutions. In the latter, abductive explanations are obtained in a deductive manner. By further recasting the conditions stated in Definition 3, we can construct $Th \cup \{\neg Obs\} \models \{\neg\Delta\}$, where $\{\neg\Delta\} = \bigvee_{\delta_j \in \Delta} \neg\delta_j$. In this scenario, we seek the consistent logical consequences of the theory and the negated observations comprising negations of hypotheses, which constitute the diagnoses.

4 Algorithm selection for abductive model-based diagnosis

In practice, we observe that different approaches perform noticeably better on certain problem instances than others. Yet, often no single solver computes solutions optimally on every example [36, 39, 64]. The objective of algorithm selection is to identify the most appropriate method out of a portfolio of techniques for a given problem instance in regard to a certain performance metric [54]. A common target is to select the method minimizing the overall computation time. However, other measures such as accuracy or simplicity of solutions may be applied. To chose the appropriate algorithm a mapping between features characterizing a problem instance and the preferable method to solve it is necessary. This mapping between the algorithms and the problem attributes requires empirical performance data on representative samples of all approaches within the portfolio. On basis of the features extracted and the execution records, a predictor is built which should forecast the "best" algorithm on the instance at hand given the problem's attributes. To achieve this, the predictor has to be able to uncover aspects of the problem which influence the performance of the methods. Rice [54] advocates for the use of features inherent to the problems within the problem space to ensure an accurate selection. In order to be beneficial running the predictor may not be more expensive than solving the problem itself [34].

4.1 Structural features

Some intuitive measures of the complexity of a diagnosis problem are the number of abducible variables, i.e., hypotheses, the number of manifestations, i.e., propositions from $A \setminus Hyp$, and connections within the theory. As the models we are studying consist of Horn clauses, we can easily define their structural properties based on various graph representations. In the simplest case, the theory is characterized as a directed graph $G = (V, E)$, where V is the set of vertices and E constitutes the set of directed edges. The nodes of the graph represent the propositional variables, while the edges are determined by the theory. For each clause $a_1 \wedge \ldots \wedge a_n \rightarrow a_{n+1}$ there exists a set of edges, such that $\forall a_i \in \{a_1, \ldots, a_n\} : (a_i, a_{n+1}) \in A$, i.e., for each proposition in a clause's body, there is a directed edge from the proposition's node to the node representing the clause's head. Considering the graph representations of the model, we can extract certain characteristics of their structure, which we subsequently use within the algorithm selection process. Note that we do not account for the logical connectives of the Horn clauses' bodies in G.

Similarly to Peng and Reggia's [52] set-covering theory, we define two sets in regard to knowledge about the relations in the Horn theory: $effects(p_i)$ and $causes(p_i)$. The first contains variables inferred by p_i. That is, assuming a cause-effect relation between propositions, the set comprises all effects or manifestations of p_i. The latter holds the variables which directly trigger the proposition p_i, i.e., the causes of p_i.

4.1.1 Outdegree and indegree

Based on the directed graph, we can compute for each vertex representing a hypothesis its outdegree. This specifies the number of variables affected by said proposition. Similarly, we record the indegree of each node, i.e. the number of propositional variables inferring the variable. Collected over the entire model these measures provide an intuitive metric of the basic magnitude of the theory and the connectedness of G.

4.1.2 Covering and overlap

Several propositions may be covered by the same cause. On basis of this we can define a direct covering metric for

each pair of propositions as the ratio between the number of common effects and the total number of symptoms induced by the propositions:

$$covering(p_i, p_j) = \frac{|effects(p_i) \cap effects(p_j)|}{|effects(p_i) \cup effects(p_j)|}$$

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

$$overlap(p_i, p_j) = \frac{|causes(p_i) \cap causes(p_j)|}{|causes(p_i) \cup causes(p_j)|}$$

Additionally, based on the directed graph we compute a label for each vertex v, which contains all hypotheses said node is (directly and indirectly) caused by:

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

Note that we do not contemplate the connectives of the variables, hence, a label in our case is a simple set. Given the definition of a label, we define an overlap based on labels:

$$overlap_{label}(p_i, p_j) = \frac{|label(p_i) \cap label(p_j)|}{|label(p_i) \cup label(p_j)|}$$

While *overlap* only takes into account the direct causes of a proposition, $overlap_{label}$ uses the information of all predecessors of a node. By collecting these measures for any pair of hypotheses or effects, we can compute a value over the entire model.

4.1.3 Independent diagnosis subproblem

Whenever there exist several subproblems in our theory we refer to them as independent diagnosis subproblems. If several subproblems exist, G is disconnected and each independent diagnosis subproblem itself is a connected subgraph.

4.1.4 Path length

Another measure of connectedness within the model is the minimal path length between any two nodes in G. Yet, we only consider paths between nodes which are causing some other proposition. Therefore, we construct an undirected graph, where we only consider propositions as nodes, which are causing other propositions. An edge is created between two propositions, in case they are directly causing the same effect. In particular, we measure the length of the minimal path between these nodes.

4.1.5 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 approach in this context is to compute the complexity based on the adjacency matrix of the undirected graph [45]. Therefore, we simply replace the directed edges of G with undirected ones, create a String representation of the adjacency matrix and based on this derive the Kolmogorov complexity.

4.1.6 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 *indegree*, *overlap* and $overlap_{label}$ among the elements of Obs as well as the number of diagnosis subproblems involving variables of Obs, in case several exist.

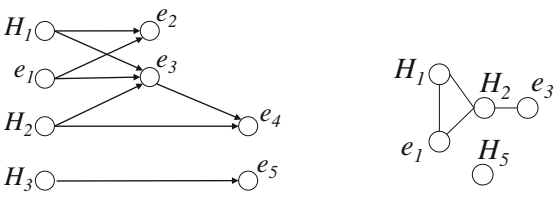
Example 2 For describing the features, we will use the Horn clause model with $Th = \{e_1 \wedge H_1 \rightarrow e_2, e_1 \wedge H_1 \rightarrow e_3, H_2 \rightarrow e_3, H_2 \rightarrow e_4, e_3 \rightarrow e_4, H_3 \rightarrow e_5\}$ as an example. Figure 1a shows G created on basis of the model. The labels are as follows:

v	H_1	H_2	H_3	e_1	e_2	e_3	e_4	e_5
$label(v)$	H_1	H_2	H_3	$\emptyset$	H_1	H_1, H_2	H_1, H_2	H_3

For instance, we can extract $outdegree(H_1) = 2$, $indegree(e_2) = 2$, $covering(H_1, H_2) = \frac{1}{3}$, $overlap(e_3, e_4) = \frac{1}{4}$ and $overlap_{label}(e_3, e_4) = 1$. We also have two independent diagnosis subproblems, namely one including H_1, H_2, e_1, e_2, e_3 and e_4 and the other one consisting of H_3 and e_5. To compute the path length between nodes functioning as causes, we construct the graph in Fig. 1b. Based on the undirected graph, we can see for instance that $path(H_1, e_3) = 2$.

4.2 Meta-approach

There are two possible variations of algorithm selection; either one algorithm of the portfolio is to be selected based on a single predictor or there is a predictor for each



(a) Directed graph G (b) Undirect graph of cause nodes

Fig. 1 Feature graphs Example 2

approach within the portfolio estimating the performance of each algorithm and choosing the method with the best result [20, 34]. For our meta-technique, we consider the first variant, where we train a single classifier for all abductive reasoning methods to select a single approach for execution. We use a 1-of n portfolio [64], i.e., we chose one of the n algorithms in our portfolio and invoke it in order to solve our current diagnosis problem. The portfolio consists of the seven abductive reasoning methods, which we describe in Section 5.1. Our performance objective is to choose the approach with the least runtime.

Within the context of abductive diagnosis our meta-approach can be split into two stages: the online and the offline phase. In the offline step, the empirical data on computation times of the various abductive reasoning approaches is collected. On basis of the metrics and the runtime information a machine learning classifier is trained. Furthermore, in the model-based diagnosis scenario the system description has to be available before the actual diagnosis computation; thus, the majority of features can be computed offline on the diagnosis model. Only a portion of attributes, namely those associated with the set of observations, has to be derived online.[1] Figure 2 lists all features extracted from the models, with the *Instance specific/Observation dependent* constituting the attributes computed online.

Algorithm METAB describes the online portion of the meta-approach, which is executed whenever new diagnosis problem is applied. The procedure takes the PHCAP, the offline-trained machine learning classifier, and the already computed metrics of the diagnosis model as input. 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 ϕ with the measurements of the current diagnosis problem. While Hutter et al. [21] state that the feature extraction method should be highly 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. By providing all features to the machine learning algorithm, we in turn retrieve a predicted best abduction method α out of our portfolio for this specific scenario based on the trained classifier and the instance's features. Subsequently, we instantiate the diagnosis engine with the corresponding abduction method as well as abduction problem and compute the set of abductive diagnoses, i.e., $\Delta - Set$.

Algorithm 1 METAB [29]

Input: A: the set of propositional variables, Hyp: the set of hypotheses, Th: the Horn theory, Obs: the set of observations, CL: trained classifier, $\phi_{offline}$: previously computed model metrics
Output: $\Delta - Set$: set of consistent minimal diagnoses

```
φ_online ← computeMetrics(A,Hyp,Th,Obs)   ▷ Computes the instance-based features
φ = φ_offline ∪ φ_online                  ▷ Creates the entire feature vector
α ← predict(φ, CL)                        ▷ Forecasts the best performing algorithm α for the diagnosis problem
Δ − Set ← diagnose(α, A,Hyp,Th,Obs)       ▷ Computes solutions with α for the diagnosis problem
return Δ − Set
```

5 Evaluation

In this section, we evaluate the feasibility of our meta-approach based on the an empirical study we have conducted previously [33]. This foregoing evaluation has comparatively analyzed conflict-driven and consequence finding techniques for abductive reasoning based on two benchmarks; one set of experiments utilizes examples stemming from practice, while the other encompasses artificially generated diagnosis problems. The objectives of the evaluation in this paper are two-fold; on the one hand, we aim at assessing the quality of the structural attributes to train a machine learning classifier to forecast the most efficient algorithm in regard to its runtime for a specific PHCAP instance. On the other hand, we want to determine the overall efficiency of the meta-approach in comparison to consistently using a single abductive diagnosis techniques in the portfolio.

In the upcoming subsections, we first report on the abductive reasoning methods utilized in the previous study which function as the algorithms for our portfolio in this paper. Second, in Section 5.2 we describe the two different types of sample sets we have utilized and then report on the empirical performance data we extracted from this previous study which we subsequently utilize to determine the quality of our meta-approach. A more detailed description of the algorithms, their implementation, the benchmarks, and the runtime data obtained can be found in Koitz and Wotawa [33]. In Section 5.4,

[1] In the case of general abductive reasoning it might not be the case that partial information of the theory is already available, hence, all attributes have to computed online

Fig. 2 Features

Logic model specific:
1. *Number of hypotheses*
2. *Number of effects*, i.e., propositions in $A \setminus Hyp$
3. *Number of causal relations*, i.e., clauses in the theory

Directed Graph:
4-6. *Outdegree* of hypothesis nodes: maximum, average, standard deviation
7-9. *Indegree* of effect nodes: maximum, average, standard deviation
10-12. *Covering*: maximum, average, standard deviation
13-15. *Overlap*: maximum, average, standard deviation
16-17. *Overlap with labels*: maximum, average, standard deviation
18. *Number of independent diagnosis subproblems*
19. *Average size of independent diagnosis subproblems*

Undirected Graph:
20. *Shortest path between causes*: maximum, average, standard deviation
21. *Kolmogorov complexity* based on adjacency matrix

Instance specific/Observation dependent:
22. *Number of observations*
23-25. *Indegree current observation nodes*: maximum, average, standard deviation
26-28. *Overlap current observation*: maximum, average, standard deviation
29-31. *Overlap with labels current observation*: maximum, average, standard deviation
32. *Number of independent diagnosis subproblems* including current observations

we give an overview of the evaluation set-up for the meta-approach and subsequently present the results of this evaluation.

5.1 Portfolio

In this subsection, we give a short description of each abductive reasoning method as well as details on the implementations used in the empirical evaluation. For a more in depth discussion we refer to our previous work [33].

5.1.1 Abduction with the ATMS

As discussed in the preliminaries, explanations can be derived via consequence finding; this type of reasoning infers logical consequences from a background theory while restricting solutions to a target language referred to as a production field [40]. The production field specifies the conditions applying to the clauses generated, e.g., their cardinality or literals admissible. The ATMS is a reasoner for propositional Horn clauses, allowing to derive logical consequences given a query w.r.t. the underlying background theory [35, 42]. The constructed consequences only contain assumptions, i.e., hypotheses. Coupled with a problem solver, the ATMS determines the belief of data, while the problem solver performs the inference steps. Internally, the ATMS constructs a directed graph in which propositions are represented as nodes and the edges are based on the relations between the variables. Each node stores a label that contains all hypotheses allowing to infer the node. de Kleer [7] provides an algorithm that ensures the labels are sound, complete, minimal, and consistent. Given a PHCAP and an ATMS that has already processed the background theory, one simply has to add a clause $o_1 \wedge o_2 \wedge \ldots \wedge \ldots o_n \rightarrow explain$, where $o_1, o_2, \ldots, o_n \in Obs$ and $explain \notin A$. The label of the node *explain* then contains all subset minimal sets of hypotheses which are consistent with the theory and allow to infer the observations; hence, the label of *explain* holds all parsimonious diagnoses. We realized the ATMS[2] within Java and refer to the approach as *ATMS* within our experiments.

5.1.2 Abduction as consequence finding via SOL-resolution

Inoue [23] proposes SOL-resolution as a sound and complete strategy to derive subset minimal logical consequences belonging to the production field. By defining an additional skip rule, SOL-resolution bypasses literals belonging to the production field instead of resolving them. To obtain abductive explanations as logical consequences using SOL-resolution, the background theory has to be extended by the negation of observations, i.e., a clause $\bigvee_{o_i \in Obs} \neg o_i$ is added to the knowledge base. The production field contains the negation of the hypotheses, i.e., $\forall h \in Hyp : \neg h$, to ensure only abducibles are considered in the derivation. After the computation, hypotheses contained within the consequences, have to be converted back their positive counterpart to obtain abductive diagnoses. In our evaluation, we exploit the consequence finding tool SOLAR[3] [47], which is a Java implementation of SOL-resolution within the framework of tableau calculus for first order full clausal theories. Consequence finding with SOLAR is denoted *CF* within the evaluation.

5.1.3 Conflict-driven search via HS-DAG

In proof-tree completion-style abduction, contradictions appear given the background theory and the negated observations. These conflicts constitute the explanations.

[2] Note this is an unfocused version of the ATMS restricted to Horn clauses.

[3] Used version: SOLAR 2 `(Build 315)`

Reiter [53] has developed an approach to derive diagnosis within the consistency-based framework via conflicts arising from the manifestation contradicting the normal system behavior. The method exploits the hitting set relation between conflicts and consistency-based diagnoses.[4] Reiter's technique operates on a tree who's nodes are either labeled by a conflict, which is returned by an external theorem prover, or constitute a minimal hitting set, i.e., a consistency-based diagnosis. The tree structure implicitly removes already computed refutations and their supersets from further consideration. Greiner et al. [15] later corrected shortcomings of the algorithm description in regard to non-minimal contradictions and developed Hitting Set Directed Acyclic Graph (HS-DAG) utilizing a directed acyclic graph instead of a tree.

In the case of abductive diagnoses, we are interested in the conflicts generated by the theorem prover which should only encompass elements of *Hyp*. Hence, in our evaluation we use the publicly available Java diagnosis engine jdiagengine[5] [51] that implements a conflict-driven search via HS-DAG coupled with an incremental assumption-based linear time Horn clause theorem prover (LTUR) [43]. Since the refutations returned by LTUR are not guaranteed to be minimal, the returned conflicts may not correspond to the parsimonious explanations we seek to obtain. We propose two solutions to this issue: (1) Simply recording all contradictions while constructing the entire HS-DAG and afterwards removing all conflict supersets ensures all remaining refutations correspond to the parsimonious diagnoses. We refer to this implementation as *HS-DAG* within our empirical study. (2) Another option is to minimize conflicts right after they have been returned by the theorem prover by employing a minimization procedure such as QuickXplain [24]. In the set-up $HS\text{-}DAG_{QX}$ we minimize the refutations computed by LTUR right away before continuing to construct the tree. For this purpose, we implemented an assumption-based QuickXplain in Java. QuickXplain itself depends on a theorem prover, which in our case is another instance of jdiagengine's assumption-based LTUR.

5.1.4 Conflict-driven search via power set exploration

Conflicts are known as MUSes or unsatisfiable cores within the area of infeasibility analysis. A Minimal Unsatisfiable Subset (MUS) is a set of clauses that cannot be satisfied simultaneously, while every proper subset of an MUS is satisfiable [37]. Liffiton et al. [37] suggests to compute MUSes by traversing the power set of clauses encoded as a Boolean formula. This Boolean formula characterizes the already explored parts of the lattice.

The computation starts with a seed, i.e., a set of clauses from the unexplored region of the power set; in case the seed is unsatisfiable it is reduced to an MUS, i.e., the lattice is descended until a minimal set of unsatisfiable clauses is reached. Afterwards the encoding of the lattice is augmented in such a way that the MUS and all of its supersets are marked as already processed. In case the seed is satisfiable it is expanded until a maximal set of satisfiable clauses, a Maximal Satisfiable Subset (MSS), is found. Again the power set encoding is adapted to update the already investigated portions of the lattice. Once there are no more unexplored regions, all MUSes, i.e., conflicts, have been uncovered. Arif et al. [1] present a version favoring unsatisfiable seeds by computing maximal models of the map, i.e., the maximum number of literals is *true* without violating a clause. This extension renders the ascending of the lattice in case the seed is satisfiable unnecessary.

We have implemented the general approach proposed by Liffiton et al. [37] in Java. To favor unsatisfiable cores early on in the computation, we implemented Arif et al. 's [1] method to find maximal model seeds. In addition, we use the SAT solver SAT4J[6] [9] in order to determine the satisfiability of the Boolean formula representing the map. For our first set-up *XPLorer* we implemented the insertion-based MUS extraction algorithm as suggested by Arif et al. [1] using jdiagengine's incremental assumption-based LTUR. In the second set-up $XPLorer_{QX}$, we take advantage of the fact that each unsatisfiable seed already constitutes a conflict and the MUS extraction merely reduces it toward a minimal contradiction; hence, we can apply the assumption-based QuickXplain Java implementation to minimize the unsatisfiable seed to an MUS, i.e., an abductive explanation.

5.1.5 Abduction under stable model semantics

In ASP, abduction is framed as the search for stable models of a logic program representation of the problem [13]. The resulting stable models, i.e., answer sets, then constitute the abductive explanations. We use an encoding of propositional abduction by Saikko et al. [55][7] in conjunction with the C++ ASP solver clingo 4.5.4[8] [12]. To enumerate all subset minimal answer sets, we adapt the encoding by removing the optimization criteria and call the solver with `--heuristic=Domain`, `--enum-mod=domRec`, and`--dom-mod=5,16`. These parameters ensure the

[4] Note here that consistency-based diagnoses are not equivalent to abductive explanations, as the required relations between solutions and observations are based only on consistency instead of entailment.

[5] www.ist.tugraz.at/modremas/index.html

[6] www.sat4j.org/

[7] www.cs.helsinki.fi/group/coreo/abhs/

[8] www.potassco.org/clingo/

Table 1 Sample statistics [33]

	Artificial Samples				FMEA Samples			
	MIN	MAX	AVG	MED	MIN	MAX	AVG	MED
$\|Hyp\|$	10	504	275.07	320.00	3	90	26.16	20.00
$\|A \setminus Hyp\|$	6	6,466	1,903.23	1,668.00	5	83	26.60	21.00
$\|Th\|$	10	7,186	2,950.10	2,731.00	12	298	70.59	37.00
$\|Obs\|$	1	5	2.86	3.00	1	29	10.79	10.00
$\|\Delta\|$	1	50	2.76	1.00	1	2,288	67.98	6.00

computation of subset minimal answer sets. We denote this approach *ASP* in the evaluation.

5.2 Benchmarks

We exploit two sample sets for our evaluation: (1) an artificially created set of PHCAPs (*Artificial Samples*) and (2) diagnosis problems constructed based on real system failure assessments characterizing component faults and their manifestations (*FMEA Samples*). In Table 1 we report on the characteristics of the generated abduction problems.

Figure 3 shows the principle structure of the artificial examples as an And-or-graph, where $H_i \in Hyp$ and $e_j \in A \setminus Hyp$. Each rule's body encompasses elements of A, while the head may only contain literals from $A \setminus Hyp$. The samples were built based on several parameters: n sets the minimal number of negative literals in the clause, r determines how many clauses are generated with the same set of negative literals, o fixates the maximum overlap between positive literals of different clauses, and k determines the maximum number of observations, which are randomly selected from $A \setminus Hyp$ and for simplicity only comprise positive observations. We obtained 166 PHCAPs invoked with n=*1*, r=o=*15*, and k=*5*. FMEA is a reliability analysis tool that records possible component faults and their effects on the system behavior and function [18]. Thus, these assessments provide information suitable for constructing an abductive diagnosis model [62]. For the evaluation, we use twelve FMEAs encompassing failure knowledge of different technical systems (e.g., electrical circuits, a connector system by Ford, Focal Plane Unit of the Heterodyne Instrument for the Far Infrared built for the Herschel Space Observatory, the Anticoincidence Detector mounted on the Large Area Telescope of the Fermi Gamma-ray Space Telescope, printed circuit boards, the Maritim IT Standard, as well as components, such as rectifier, inverter, transformer, main bearing, and backup components of an industrial wind turbine) and compile them into abductive knowledge bases. The compilation procedure, as described by Wotawa [62], matches each row within the FMEA table to a set of bijunctive Horn clauses. Each clause represents the cause-effect relation between a fault mode and one of its manifestations. These FMEA-based models, thus, only contain implications from a single fault to a single effect. Thus contrast the artificial examples, which can have several levels of depth.

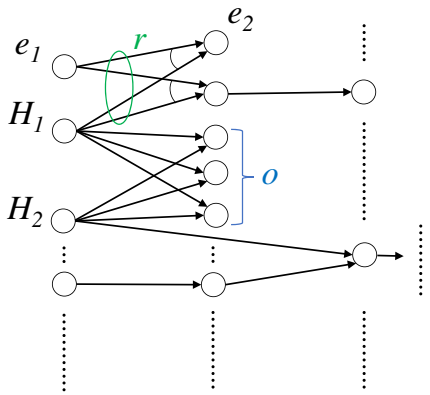


Fig. 3 Artificial example structure [33]

To construct the 213 diagnosis problems, we randomly chose observations from the set of failure effects described in the assessment. In contrast to the artificial examples, the FMEA models only comprise bijunctive definite Horn clauses, i.e., each clause features a single hypothesis in the body and a proposition from $A \setminus Hyp$ as head.

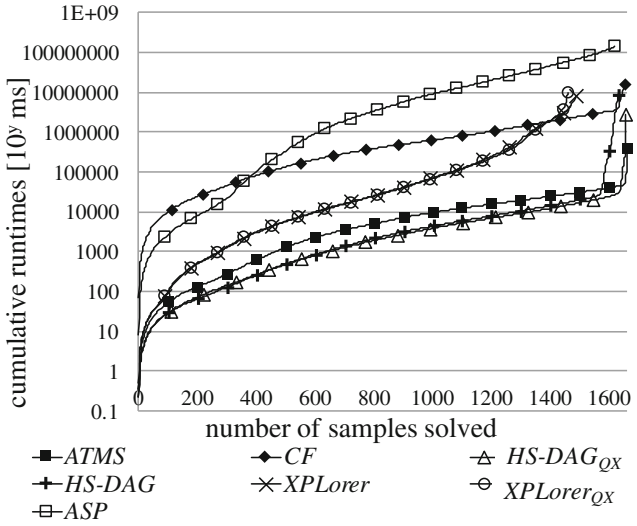
5.3 Empirical performance data

To generate the performance data necessary for constructing a classifier for our meta-approach, we obtained execution time data for all algorithms contained in our portfolio on both benchmarks. Each algorithm was invoked ten times on each PHCAP with the aim to compute all parsimonious diagnoses. In case the execution on a PHCAP was not finished after twenty minutes, the computation was interrupted. Timed out executions are penalized with $\theta = 40$*minutes*. Figure 4a and b depict the number of samples solved for growing cumulative log runtime.

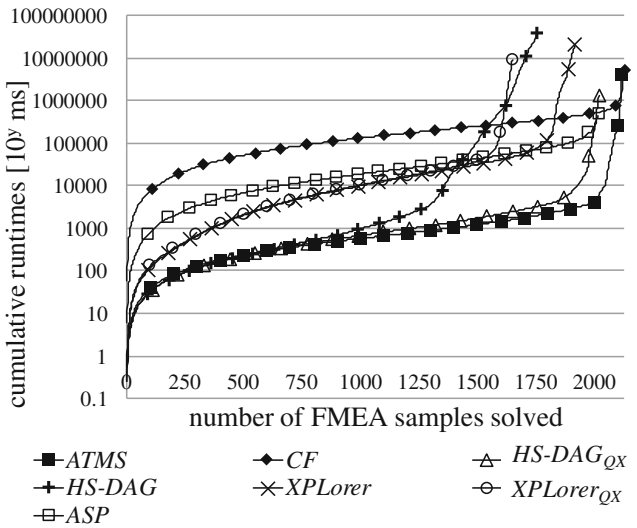
Given the data we can determine that the two most promising techniques based on the benchmarks are *ATMS* and *HS-DAG*$_{QX}$. Both methods solve a reasonable amount of samples within the given time frame and our data shows that *ATMS* on average is the most efficient approach, while *HS-DAG*$_{QX}$ computes diagnoses faster for more instances. Especially on *FMEA Samples*, we can observe that the *ATMS* is the superior approach. Even though the methods using general solvers, i.e., *CF* and *ASP*, are around two orders of magnitude slower than the best Horn reasoner, they show a consistent performance by being able to compute explanations for most samples within the given runtime allowance.

5.4 Meta experiment Set-up

METAB itself is implemented in Java. To create the predictor based on the features, we exploit the Waikato Environment



(a) *Artificial Samples*



(b) *FMEA Samples*

Fig. 4 Numbers of diagnosis samples solved over time [33]

for Knowledge Analysis (WEKA) library [17] which provides a vast variety of machine learning algorithms. For each PHCAP execution we collect the metrics listed in Fig. 2 to gather our attributes for prediction. The feature vector itself holds one more nominal value: the class attribute. It corresponds to the algorithm's name that has solved the diagnosis problem the fastest in the experiments conducted previously [33]. Hence, this last attribute is to be predicted. To evaluate the classification accuracy based on the features, we randomly split each benchmark into a training set comprising 80 % of the data and a test set holding the remaining 20 % of examples. Due to dividing samples arbitrarily, it is not warranted that each PHCAP is represented in equal measures within the test set. Hence, it may also occur that the test set comprises samples particularly suitable for a certain approach.

To determine an appropriate machine learning method, we performed model selection via 10-fold cross-validation on the training data for several classification algorithms available in WEKA. Based on the accuracy results obtained we chose to use the random subspace method [19] in combination with WEKA's decision tree with reduced-error pruning (REP Tree) for *Artificial Samples*. In the random subspace method, subsets of components of the feature vector are selected pseudorandomly and a decision tree is generated based solely on the chosen attributes. The resulting classifier then depends on several decision trees constructed in this manner, i.e., a decision forest. For *FMEA Samples* we decided on WEKA's decision table majority classifier, which also relies on a suitable subset of features to build the decision table.

5.5 Results

Table 2 depicts all classification statistics based on (1) the 10-fold cross-validation on the training data and (2) the evaluation on the test set. Utilizing the random subspace method we reach between 66.39 % and 67.17 % correctly labeled samples. A PHCAP is labeled correctly in case the predicted algorithm was the most efficient on the problem instance in our empirical performance data. For the *FMEA Samples* we receive slightly better results with more diagnosis problems (between 76.76 % and 77.79 %) classified correctly.

Ideally, we could improve upon the accuracy results. A common strategy to enhance classification precision is to adapt the feature vector. In this respect, we explore WEKA's attribute selection in order to determine whether we could remove certain features while achieving the same or better prediction quality. Since the number and composition of the reduced feature set depends highly on the performed selection process, we conducted some informal evaluations to decide on the leading method. In the end, we chose to rank attributes in the artificial case due to their information gain in consideration of the class feature and in case of the *FMEA Samples* based on a subset of features that are statistically relevant to the class attribute (*Relief Feature Selection*). Using these methods we could diminish the set of features to fifteen and twenty-one in case of the artificial and FMEA examples, respectively.[9] As reported in Table 3 the results improved from 67.17 % to 69.58 % and 76.76 % to 78.41 % on the test sets. Table 4 lists the selected attributes according to their rank. While the suggested feature sets diverge, there are certain attributes

[9] Of course the attribute to predict, i.e., the most efficient algorithm, remains in the feature vector.

Table 2 Classification Statistics

		Artificial Samples	FMEA Samples
	Classifier	Random Subspace Method with REP Tree	Decision Table
Cross Validation [10-fold]	|Data Set|	1660	2130
	Total Training Time [in ms]	114	604
	Total Test Time [in ms]	29	58
	Correctly Classified Instances	1102 (66.39 %)	1657 (77.79 %)
	Incorrectly Classified Instances	558 (33.61 %)	473 (22.21 %)
	Mean Absolute Error	0.17	0.16
Train and Test [80 %,20 %]	|Train|	1328	1704
	|Test|	332	426
	Total Training Time [in ms]	114	604
	Total Test Time [in ms]	29	58
	Correctly Classified Instances	223 (67.17 %)	327 (76.76 %)
	Incorrectly Classified Instances	109 (32.83 %)	99 (23.24 %)
	Mean Absolute Error	0.18	0.16

which seemingly are important in both benchmarks, namely the metrics which characterize the basic composition of the theory, e.g., *overlap* and *indegree*.

The confusion matrices in Table 5a and b give a deeper insight into the number of correctly and wrongly labeled instances of the test set based on the attribute selected classifiers. Each column of the matrix reports on the number of instances labeled a certain algorithm, while the rows represent the number of PHCAPs the algorithm was superior to the other approaches. For example, Table 5a shows that *HS-DAG* was predicted as the most efficient method 191 of 332 times, while it actually performed the best on 169 samples. Hence, on 22 samples the classifier incorrectly selected *HS-DAG*. In 33 cases, the instance was mislabeled as $HS\text{-}DAG_{QX}$ though *HS-DAG* would have been the best suited candidate. For 35 examples the situation was the other way around. This is due to both approaches experiencing a similar runtime behavior for most samples (as is also apparent from Fig. 4).

Table 6 reports on the binary classification measures for each approach on the test set. The best results in each column are emphasized. Precision denotes the proportion of cases correctly labeled an approach to all the samples the algorithm was selected as the abduction method, whereas Recall is the ratio of samples correctly predicted to instances where the approach is actually the fastest. The F_1-Score is a combined metric based on the weighted average of Precision and Recall [58]. Considering for example the

Table 3 Attribute selected classification statistics

		Artificial Samples	FMEA Samples
	Classifier	Random Subspace Method with REP Tree	Decision Table
	Attribute Selection Evaluator	Information Gain	Relief Feature Selection
Cross Validation [10-fold]	|Data Set|	1660	2130
	Total Training Time [in ms]	114	604
	Total Test Time [in ms]	29	58
	Correctly Classified Instances	1093 (65.84 %)	1655 (77.70 %)
	Incorrectly Classified Instances	567 (34.16 %)	475 (22.30 %)
	Mean Absolute Error	0.17	0.16
Train and Test [80 %,20 %]	|Train|	1328	1704
	|Test|	332	426
	Total Training Time [in ms]	53	650
	Total Test Time [in ms]	26	97
	Correctly Classified Instances	231 (69.58 %)	334 (78.41 %)
	Incorrectly Classified Instances	101 (30.42 %)	92 (21.59 %)
	Mean Absolute Error	0.17	0.16

Table 4 Selected attributes

	Artificial Samples	FMEA Samples
1	*Covering*: standard deviation	*Number of observations*
2	*Overlap with labels*: standard deviation	*Overlap current observation with label*: maximum
3	*Overlap with labels*: average	*Overlap current observation with label*: standard deviation
4	*Indegree current observation nodes*: average	*Overlap current observation*: maximum
5	*Outdegree of hypothesis nodes*: standard deviation	*Number of independent diagnosis subproblems including current observations*
6	*Indegree current observation nodes*: standard deviation	*Indegree current observation nodes*: maximum
7	*Overlap current observation*: maximum	*Indegree current observation nodes*: standard deviation
8	*Overlap current observation with label*: average	*Overlap current observation*: standard deviation
9	*Overlap current observation*: average	*Shortest path*: standard deviation
10	*Indegree of effect nodes*: average	*Indegree of effect nodes*: maximum
11	*Outdegree of hypothesis nodes*: average	*Number of hypotheses*
12	*Indegree current observation nodes*: maximum	*Shortest path*: maximum
13	*Indegree of effect nodes*: standard deviation	*Indegree of effect nodes*: standard deviation
14	*Overlap current observation*: standard deviation	*Kolmogorov complexity*
15	*Covering*: average	*Number of causal relations*
16		*Number of effects*
17		*Overlap*: standard deviation
18		*Overlap with labels*: standard deviation
19		*Number of independent diagnosis subproblems*
20		*Overlap current observation*: average
21		*Indegree current observation nodes*: average

Table 5 Confusion matrices. The rows represent the actual number of instances labeled as the class, while the columns show the number of predicted instances

		Predicted							
		ATMS	CF	HS-DAG	HS-DAG$_{QX}$	XPLorer	XPLorer$_{QX}$	ASP	Total
(a) Artificial Samples									
Actual	ATMS	**12**	0	16	8	0	0	0	36
	CF	0	**0**	0	0	0	0	0	0
	HS-DAG	0	0	**136**	33	0	0	0	169
	HS-DAG$_{QX}$	1	0	35	**83**	0	0	0	119
	XPLorer	0	0	3	3	**0**	0	0	6
	XPLorer$_{QX}$	0	0	1	1	0	**0**	0	2
	ASP	0	0	0	0	0	0	**0**	0
	Total	13	0	191	128	0	0	0	332
(b) FMEA Samples									
Actual	ATMS	**216**	1	10	17	0	0	0	244
	CF	1	**2**	0	0	1	0	0	4
	HS-DAG	24	0	**32**	14	0	0	0	70
	HS-DAG$_{QX}$	16	0	8	**84**	0	0	0	108
	XPLorer	0	0	0	0	**0**	0	0	0
	XPLorer$_{QX}$	0	0	0	0	0	**0**	0	0
	ASP	0	0	0	0	0	0	**0**	0
	Total	257	3	50	115	1	0	0	426

The bold value indicates the number of correctly predicted instances

Table 6 Classifier performance measures

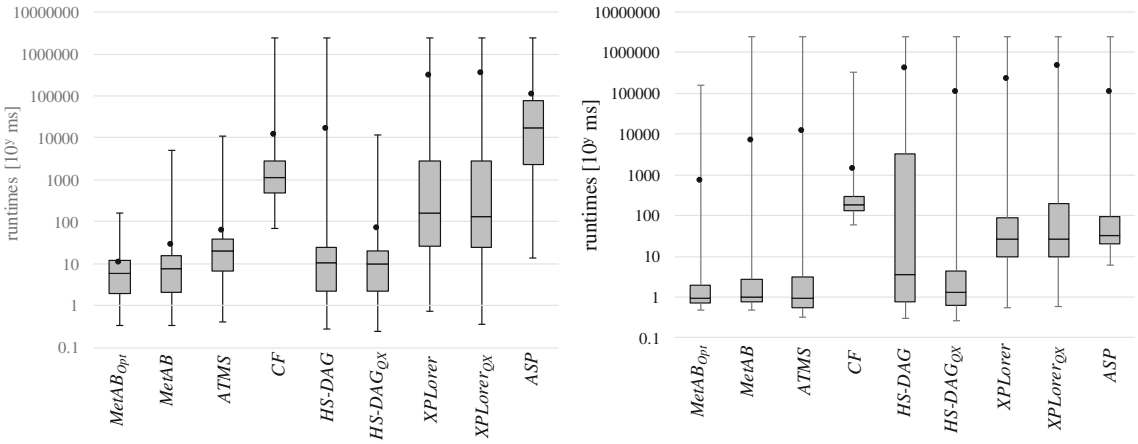
	Precision	Recall	F_1-Score	Accuracy	Specificity	AUROC
(a) Artificial Samples						
ATMS	**0.92**	0.33	0.49	0.92	**1.00**	0.85
CF	0.00	0.00	0.00	**1.00**	**1.00**	–
HS-DAG	0.71	**0.80**	**0.76**	0.77	0.66	0.81
HS-DAG$_{QX}$	0.65	0.70	0.67	0.76	0.79	0.82
XPLorer	0.00	0.00	0.00	0.98	**1.00**	**0.98**
XPLorer$_{QX}$	0.00	0.00	0.00	0.99	**1.00**	0.55
ASP	0.00	0.00	0.00	**1.00**	**1.00**	–
(b) FMEA Samples						
ATMS	**0.84**	**0.89**	**0.86**	0.83	0.77	0.91
CF	0.60	0.50	0.57	0.99	**1.00**	**0.99**
HS-DAG	0.64	0.46	0.53	0.87	0.95	0.88
HS-DAG$_{QX}$	0.73	0.78	0.75	0.87	0.90	0.92
XPLorer	0.00	0.00	0.00	**1.00**	**1.00**	–
XPLorer$_{QX}$	0.00	0.00	0.00	**1.00**	**1.00**	–
ASP	0.00	0.00	0.00	**1.00**	**1.00**	–

The bold numbers indicate the best value

ATMS in *Artificial Samples*, we can see that the Precision value is close to the best value of 1, since it was only once selected incorrectly. However, the Recall value is rather disappointing due to various samples where *ATMS* were in fact the fastest method, but was not identified as such by the classifier. This trade-off is also apparent in the F_1-Score, which hence is only 0.49. In contrast in the *FMEA Samples*, Precision and Recall values are good and hence also the F_1-Score for *ATMS* is promising. Other common measures in multi-class classification are Accuracy, i.e., the overall effectiveness of a classifier, Specificity, i.e., how well negative labels are classified, and the area under the receiver operating characteristic curve (AUROC), i.e., the ability of the predictor to avoid false labeling [58, 61].

From the contingency tables is apparent that *HS-DAG*, *HS-DAG$_{QX}$* and *ATMS* are the best performing approaches with some variations between the sample sets. A premature analysis of these results would suggest that applying for instance *HS-DAG* to every artificial instance would yield the optimal runtime for most problems. However, based on the entire set of problems, i.e. test and training, *HS-DAG* is not the most efficient approach as its computation time is notably larger on several instances than other algorithms. Thus, we propose to use our meta-approach MetAB which we subsequently compare to always using a single diagnosis method.

MetAB's overall runtime 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



(a) *Artificial Samples* (b) *FMEA Samples*

Fig. 5 Statistical distribution for the runtimes [10^y ms] on the test set

Table 7 Runtime results of the meta-approach in comparison to *ATMS* and *HS-DAG*$_{QX}$

	Artificial Samples			FMEA Samples		
	MetAB	ATMS	HS-DAG$_{QX}$	MetAB	ATMS	HS-DAG$_{QX}$
MIN	0.32	0.34	**0.22**	0.48	0.33	**0.25**
MAX	**4,940.16**	170,838.66	2,400,000	2,400,000.23	**2,400,000**	**2,400,000**
AVG	**28.16**	226.28	10,269.27	**6,963.16**	14,239.04	124,552.12
MED	**7.3**8	19.91	8.13	1.04	**0.91**	1.39
SD	**271.09**	4,529.92	148,023.42	**117,443.57**	174,655.60	531,157.15

The bold numbers indicate the best value

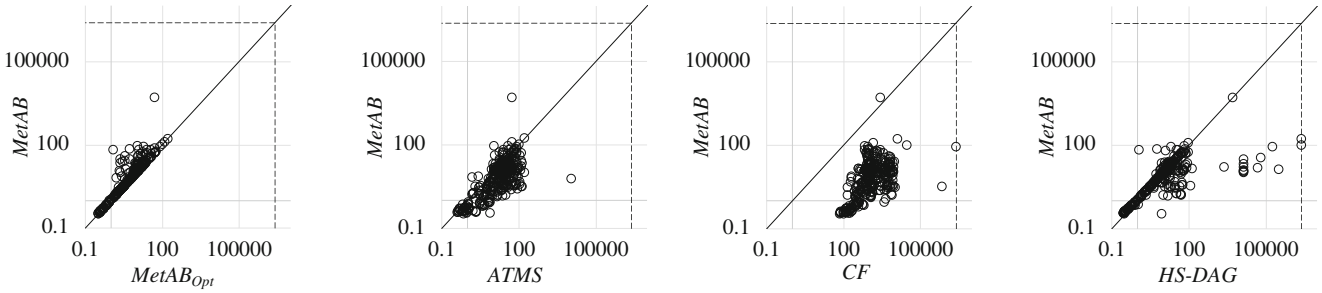
predict the "best" algorithm, and (3) the diagnosis time of the suggested abduction procedure. In regard to the feasibility of the meta-approach, we like to refer back to a particular characteristic of model-based diagnosis, namely the availability of the system description offline. As the model has to be present before the computation of the diagnoses, it allows us to extract most of the metrics utilized in the algorithm selection offline. On average we have observed an offline feature computation time of around 52,868.39 ms for the artificial instances and 165.85 ms for the FMEA benchmark.[10] The online computation of the features, which are inherent to the specific instance of the PHCAP, is negligible (on average $<$ 0.1 ms). Based on the total classification time on the test set, we approximate that the prediction for a single instance on average below 0.5 ms. The third factor, the diagnosis time, is much dependent on the predictive capabilities of the classifier. A known drawback of this type of algorithm selection approach, where a single method is chosen and executed on the instance, is that in case the prediction is incorrect the meta-approach might be rather inefficient [34].

Figure 5a and b depict the distributions of the log runtimes of the abductive reasoning algorithms on the test sets. In addition to the seven abduction methods, we have included on the one hand our meta approach *MetAB* and on the other hand an optimal algorithm selection approach *MetAB*$_{Opt}$. For *MetAB*$_{Opt}$ we assume a classifier that labels each diagnosis problem correctly and thus computes the explanations with the approach requiring the minimal runtime. *MetAB*$_{Opt}$'s computation time thus consists of the collection of the online metrics, the classification, and the diagnosis using the fastest approach. For the artificial examples, *MetAB* and *MetAB*$_{Opt}$ show the best average and median runtime results, with an average percentage difference between them of 15.30 %. However, In case of the FMEA examples, *ATMS* provides the best median results on the test set followed by *MetAB*$_{Opt}$ and *MetAB*. This is unsurprising given the dominance of *ATMS* on *FMEA Samples*. On average, yet, *ATMS* is slower than *MetAB*$_{Opt}$ and *MetAB*. The average percentage difference between *MetAB*$_{Opt}$ and *MetAB* is 10.93 %.

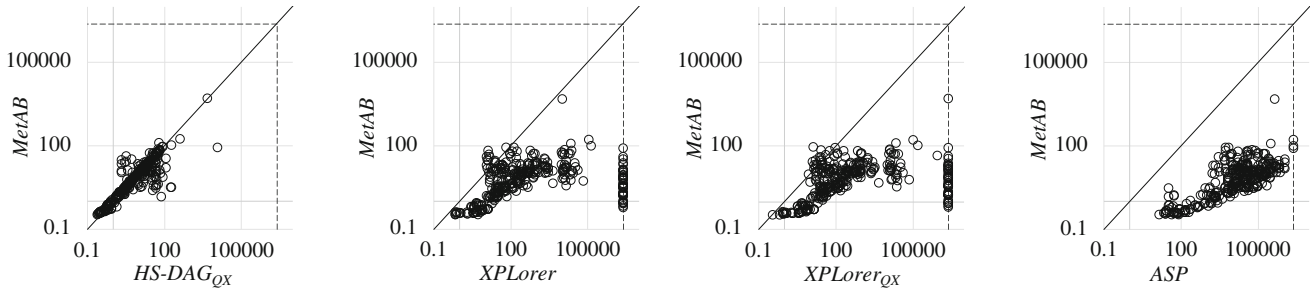
Table 7 lists the runtime results of *MetAB* in comparison to *ATMS* and *HS-DAG*$_{QX}$. From the table we can observe that for the FMEA examples the classifier at least once chose an algorithm which cannot compute the diagnoses within the given time frame, hence, the maximum runtime corresponds to the penality of $\theta = 40$ *minutes* plus the classification time.

For a more in-depth comparison of METAB to the other diagnosis methods, we provide various runtime scatterplots in Figs. 6 and 7. The x and y values characterize the penalized log runtimes of the corresponding algorithm pair, while each data point represents one sample run. For instance, in Fig. 6b, we compare *MetAB* to *ATMS* on the artificial samples. Points above the diagonal represent executions where *ATMS* was more efficient than *MetAB*, while points below the line indicate samples where *MetAB* was superior. The dashed lines mark the penalized runtime, i.e., every execution exceeding the runtime limit is located on the dashed lines. As apparent from Figs. 6a and 7a, *MetAB* performs not as good as the ideal approach *MetAB*$_{Opt}$. Nevertheless, given the classification accuracy most data points are on the diagonal or close to it confirming that the selected algorithm is either the fastest or among the best solvers for the PHCAP. For the FMEA examples, we can see that on one instance, *MetAB* choses an algorithm exceeding the time limit hence featuring a penalized runtime. From the remaining plots, we can conclude that while on some PHCAPs the meta-approach cannot compete with always choosing a single algorithm, the bulk of data points is usually located below the diagonal suggesting the benefit of *MetAB* even when an ideal prediction is not possible. In these cases, where the Figs. 6a and 7a show that while on the artificial samples *MetAB* shows convincing runtime results, the same cannot be stated for the FMEA examples. As the models only feature bijunctive Horn clauses the computation of the online features requires around the same time as the diagnosis itself. Hence, *MetAB*'s prediction only causes computational overhead. Further the scatter plots indicate that while consequence finding using SOLAR

[10] The computation effort for the offline metrics can be quite extensive given the structure of the model, due to constructing and traversing the graph.



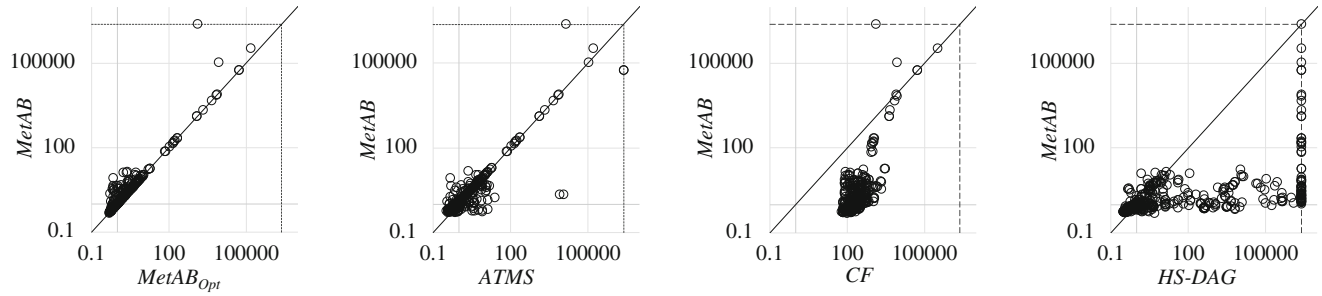
(a) *MetAB* and $MetAB_{Opt}$ (b) *MetAB* and *ATMS* (c) *MetAB* and *CF* (d) *MetAB* and *HS-DAG*



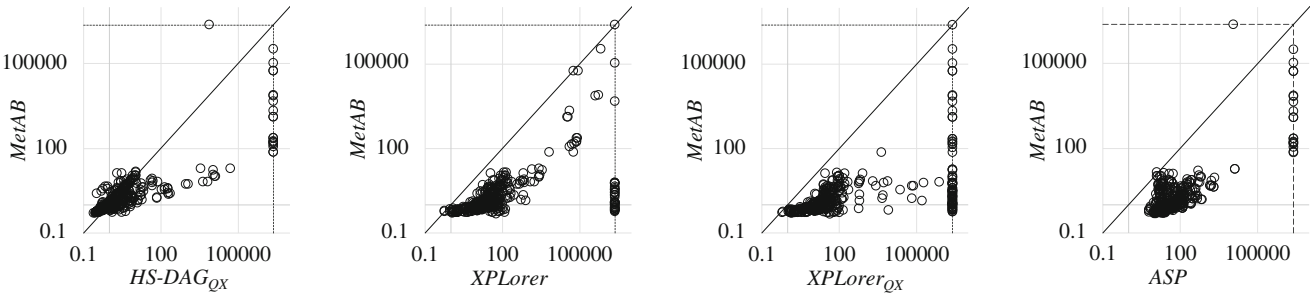
(e) *MetAB* and $HS\text{-}DAG_{QX}$ (f) *MetAB* and *XPLorer* (g) *MetAB* and $XPLorer_{QX}$ (h) *MetAB* and *ASP*

Fig. 6 Scatter plots of runtime [10^y ms] comparisons on the test set of *Artificial Samples*

does not provide efficient runtimes, it is a dependable approach computing diagnoses for all PHCAPs constructed on top of the FMEAs. Both general reasoners, SOLAR and clingo, are not specialized Horn abduction solvers, but designed for first-order clausal theories and normal as well as disjunctive logic programs, respectively. Therefore, *CF* and *ASP* are around two orders of magnitude slower than *MetAB*. However, as both approaches perform consistently



(a) *MetAB* and $MetAB_{Opt}$ (b) *MetAB* and *ATMS* (c) *MetAB* and *CF* (d) *MetAB* and *HS-DAG*



(e) *MetAB* and $HS\text{-}DAG_{QX}$ (f) *MetAB* and *XPLorer* (g) *MetAB* and $XPLorer_{QX}$ (h) *MetAB* and *ASP*

Fig. 7 Scatter plots of runtime [10^y ms] comparisons on the test set of *FMEA Samples*

and compute explanations for most samples within the given runtime allowance, we conclude that off-the-shelf tools can very well function as suitable abduction engines despite not being competitive in regard to runtime.

To evaluate the efficiency of our meta-approach in comparison to always choosing a single approach, we determine whether the median difference between *MetAB* and the best performing approaches on the corresponding sample set is significant. Since dealing with non-normal runtime distributions and under the assumption that the observations are independent and identically distributed, we apply the one-tailed Wilcoxon Signed-Rank Test [60]. Suppose paired runtime data $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ from *MetAB* and the compared abduction approach, respectively. We propose hypothesis $H_0 : m_X \leq m_Y$, stating a median difference of zero between pairs of observations. For *FMEA Samples*, we compared *MetAB* and *ATMS* and formulate our alternative hypothesis as $H_1 : m_X > m_Y$. Given the one-tailed test for $\alpha = 0.05$, we reject H_0, i.e., stating a significant difference between *MetAB* and *ATMS*. This indicates that on simple examples such as the one's generated based on FMEAs, choosing the ATMS as abduction procedure is advantageous on the test data. In regard to the artificial examples, we compared the meta-approach to *HS-DAG* and *HS-DAG*$_{QX}$ with $H_1 : m_X < m_Y$. In both cases, we accept the null hypothesis since the Wilcoxon signed-rank test determined an insignificant improvement of *MetAB* in comparison to the hitting set approaches.

6 Conclusion

Algorithm selection has been proposed as a way to deal with the issue that given different problem instances there is no universally superior approach computing solutions efficient for all samples, but the performance of a certain method highly depends on the underlying characteristics of the problem instance. Abductive diagnosis represents a novel application area of the portfolio approach. Utilizing a machine learning classifier and structural attributes of the Horn theories, we have developed a meta-approach to abductive model-based diagnosis that aims at predicting the most suitable method for a diagnosis problem on a case-by-case basis. Algorithm selection can be particularly useful in the context of model-based diagnosis, since the structural features of the underlying system description can be computed mostly offline and only the dynamic portion depending on the observations have to be derived during computation.

We evaluated the algorithm selection method within an empirical set-up based on seven abductive reasoning methods and two benchmarks. Subsequently, we compared abduction via the meta-approach to always choosing a single abductive reasoning mechanism. The accuracy of our algorithm selection technique is satisfactory and *MetAB* is in fact on average more efficient than choosing a single abductive reasoning approach on both sample sets. Hence, based on our benchmarks we can advocate for the benefit of using a portfolio method for abduction. While on average the fastest, our meta-approach cannot outperform all other abduction techniques on median on the simple diagnosis problems stemming from FMEAs. These models are characterized by bijunctive Horn clauses, where the meta-approach's set-up effort might not justify the runtime improvements. Possibly on larger models, we could observe a better performance of *MetAB* in comparison to *ATMS*, which is known to have difficultly propagating label values given larger more interconnected theories.

Thus, adapting and extending the benchmarks used for evaluation might provide improved runtime results for our portfolio approach. Even though we have applied a simple attribute selection, a deeper analysis of negligible or combinable features would definitely improve the technique since choosing the incorrect method can be an expensive mistake within our set-up. In addition, removing some of the attributes would decrease the computational effort especially in regard to the offline features, which may take a long time to derive depending on the problem structure.

There may be situations where certain instances are inherently difficult and thus require an extensive amount of time on any technique (including the meta-approach). In these situations, switching between computation methods may provide better results than a simple algorithm selection. This would be particularly interesting if approaches favor certain kinds of explanations and would produce those efficiently early on in the calculation. Then, by switching between the abduction methods and adding constraints that ensure previously derived diagnoses are not considered again, we may be able to improve the computation even further. However, it would remain to determine to what extend the selection of the next approach and the switching would increase the computation time to decide if this approach is beneficial.

Funding Information Open access funding provided by Graz University.

References

1. Arif MF, Mencía C, Marques-Silva J (2015) Efficient MUS enumeration of Horn formulae with applications to axiom pinpointing. In: International conference on theory and applications of satisfiability testing. Springer, pp 324–342
2. Bischl B, Kerschke P, Kotthoff L, Lindauer M, Malitsky Y, Fréchette A, Hoos H, Hutter F, Leyton-Brown K, Tierney K et al (2016) ASLib: a benchmark library for algorithm selection. Artif Intell 237:41–58
3. Bylander T, Allemang D, Tanner MC, Josephson JR (1991) The computational complexity of abduction. Artif Intell 49(1-3):25–60
4. Console L, Dressier O (1999) Model-based diagnosis in the real world: lessons learned and challenges remaining. In: Proceedings of the 16th international joint conference on artificial intelligence - volume 2, IJCAI'99, pp 1393–1400
5. Console L, Torasso P (1991) A spectrum of logical definitions of model-based diagnosis. Comput Intell 7(3):133–141
6. De Kleer J (1986) An assumption-based model-based diagnosis. Artif Intell 28(2):127–162
7. De Kleer J (1986) Problem solving with the AModel-based diagnosis. Artif Intell 28(2):197–224
8. De Kleer J, Williams BC (1987) Diagnosing multiple faults. Artif Intell 32(1):97–130
9. Eén N, Sörensson N (2003) An extensible SAT-solver. In: International conferemce on theory and applications of SAT. Springer, pp 502–518
10. Eiter T, Gottlob G (1995) The complexity of logic-based abduction. J ACM (JACM) 42(1):3–42
11. Friedrich G, Gottlob G, Nejdl W (1990) Hypothesis classification, abductive diagnosis and therapy. Expert Systems in Engineering Principles and Applications:69–78
12. Gebser M, Kaminski R, Kaufmann B, Schaub T (2014) Clingo = ASP + control: preliminary report. In: Leuschel M, Schrijvers T (eds) Technical Communications of the 30th International Conference on Logic Programming, pp 1–13
13. Gelfond M, Lifschitz V (1988) The stable model semantics for logic programming. In: Logic programming, proceedings of the fifth international conference and symposium, vol 88, pp 1070–1080
14. Gordon AS (2016) Commonsense interpretation of triangle behavior. In: Proceedings of the thirtieth AAAI conference on artificial intelligence. AAAI Press, pp 3719–3725
15. Greiner R, Smith BA, Wilkerson RW (1989) A correction to the algorithm in Reiter's theory of diagnosis. Artif Intell 41(1):79–88
16. Guo H, Hsu WH (2007) A machine learning approach to algorithm selection for NP-hard optimization problems: a case study on the MPE problem. Ann Oper Res 156(1):61–82
17. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH (2009) The WEKA data mining software: an update. ACM SIGKDD Explorations Newsletter 11(1):10–18
18. Hawkins PG, Woollons DJ (1998) Failure modes and effects analysis of complex engineering systems using functional models. Artif Intell Eng 12(4):375–397
19. Ho TK (1998) The random subspace method for constructing decision forests. IEEE Trans Pattern Anal Mach Intell 20(8):832–844
20. Hutter F, Hamadi Y, Hoos H, Leyton-Brown K (2006) Performance prediction and automated tuning of randomized and parametric algorithms. Principles Pract Constraint Programming-CP 2006:213–228
21. Hutter F, Xu L, Hoos HH, Leyton-Brown K (2014) Algorithm runtime prediction: methods & evaluation. Artif Intell 206:79–111
22. Ignatiev A, Morgado A, Marques-silva J (2016) Propositional abduction with implicit hitting sets. In: Proceedings of the 22nd european conference on artificial intelligence, pp 1327–1335
23. Inoue K (1992) Linear resolution for consequence finding. Artif Intell 56(2):301–353
24. Junker U (2004) QUICKXPLAIN: preferred explanations and relaxations for over-constrained problems. In: Proceedings of the 19th national conference on artificial intelligence. AAAI Press, pp 167–172
25. Kakas AC, Kowalski RA, Toni F (1992) Abductive logic programming. J Log Comput 2(6):719–770
26. Kakas AC, Van Nuffelen B, Denecker M (2001) A-system: problem solving through abduction. In: Proceedings of the seventeenth international joint conference on artificial intelligence, pp 591–596
27. Kohlas J, Anrig B, Haenni R, Monney PA (1998) Model-based diagnostics and probabilistic assumption-based reasoning. Artif Intell 104(1-2):71–106
28. Koitz R, Wotawa F (2015) Finding explanations: an empirical evaluation of abductive diagnosis algorithms. In: Proceedings of the 2015 international conference on defeasible and ampliative reasoning-volume 1423. CEUR-WS.org, pp 36–42
29. Koitz R, Wotawa F (2016) Exploiting structural metrics in FMEA-based abductive diagnosis. In: Proceedings of the twenty-seventh international workshop on principles of diagnosis (DX-2016), pp 1–8
30. Koitz R, Wotawa F (2016) Improving abductive diagnosis through structural features: a meta-approach. In: Proceedings of the international workshop on defeasible and ampliative reasoning (DARe-16), pp 1–9
31. Koitz R, Wotawa F (2016) Integration of failure assessments into the diagnostic process. In: Proceedings of the annual conference of the PHM society (PHM-2016), pp 117–128
32. Koitz R, Wotawa F (2016) On structural properties to improve FMEA-based abductive diagnosis. In: Workshop on knowledge-based techniques for problem solving and reasoning, pp 1–7
33. Koitz R, Wotawa F (2017) Faster Horn diagnosis - a performance comparison of abductive reasoning algorithms. Submitted to Artificial Intelligence
34. Kotthoff L (2014) Algorithm selection for combinatorial search problems: a survey. AI Mag 35(3):48–60
35. Levesque HJ (1989) A knowledge-level account of abduction. In: Proceedings of the 11th international joint conference on artificial intelligence-volume 2. Morgan Kaufmann Publishers Inc, pp 1061–1067
36. Leyton-Brown K, Nudelman E, Andrew G, McFadden J, Shoham Y (2003) A portfolio approach to algorithm select. In: Proceedings of the 18th international joint conference on artificial intelligence. Morgan Kaufmann Publishers Inc, pp 1542–1543
37. Liffiton MH, Sakallah KA (2008) Algorithms for computing minimal unsatisfiable subsets of constraints. J Autom Reason 40(1):1–33
38. Lucas PJ (2001) Bayesian model-based diagnosis. Int J Approx Reason 27(2):99–119
39. Malitsky Y, O'Sullivan B, Previti A, Marques-Silva J (2014) A portfolio approach to enumerating minimal correction subsets for satisfiability problems. In: International conference on AI and OR techniques in constriant programming for combinatorial optimization problems. Springer, pp 368–376
40. Marquis P (2000) Consequence finding algorithms. In: Handbook of defeasible reasoning and uncertainty management systems. Springer, pp 41–145
41. McIlraith SA (1994) Generating tests using abduction. In: Proceedings of the fourth international conference on principles

of knowledge representation and reasoning. Morgan Kaufmann Publishers Inc, pp 449–460
42. McIlraith SA (1998) Logic-based abductive inference. Knowledge Systems Laboratory Technical Report KSL-98-19
43. Minoux M (1988) LTUR: a Simplified linear-time unit resolution algorithm for Horn formulae and computer implementation. Inf Process Lett 29(1):1–12
44. Morak M, Musliu N, Pichler R, Rümmele S, Woltran S (2012) Evaluating tree-decomposition based algorithms for answer set programming. Learn Intell Optim:130–144
45. Mowshowitz A, Dehmer M (2012) Entropy and the complexity of graphs revisited. Entropy 14(3):559–570
46. Musliu N, Schwengerer M (2013) Algorithm selection for the graph coloring problem. In: International conference on learning and intelligent optimization. Springer, pp 389–403
47. Nabeshima H, Iwanuma K, Inoue K, Ray O (2010) SOLAR: an automated deduction system for consequence finding. AI Commun 23(2-3):183–203
48. Nordh G, Zanuttini B (2008) What makes propositional abduction tractable. Artif Intell 172(10):1245–1284
49. Ovchinnikova E, Montazeri N, Alexandrov T, Hobbs JR, McCord MC, Mulkar-Mehta R (2014) Abductive reasoning with a large knowledge base for discourse processing. In: Computing meaning. Springer, pp 107–127
50. Pearl J (1989) Probabilistic reasoning in intelligent systems - networks of plausible inference. Morgan Kaufmann series in representation and reasoning Morgan Kaufmann
51. Peischl B, Wotawa F (2003) Computing diagnosis efficiently: a fast theorem prover for propositional Horn theories. In: Proceedings of the 14th international workshop on principles of diagnosis, pp 175–180
52. Peng Y, Reggia JA (1990) Abductive inference models for diagnostic problem-solving. Springer, Berlin
53. Reiter R (1987) A theory of diagnosis from first principles. Artif Intell 32(1):57–95
54. Rice JR (1976) The algorithm selection problem. Advan Comput 15:65–118
55. Saikko P, Wallner JP, Järvisalo M (2016) Implicit hitting set algorithms for reasoning beyond NP. In: Proceedings of the fifteenth international conference on principles of knowledge representation and reasoning. AAAI Press, pp 104–113
56. Schüller P (2016) Modeling variations of first-order Horn abduction in answer set programming. Fundamenta Informaticae 149(1-2):159–207
57. Smith-Miles KA (2009) Cross-disciplinary perspectives on meta-learning for algorithm selection. ACM Computing Surveys (CSUR) 41(1):6
58. Sokolova M, Lapalme G (2009) A systematic analysis of performance measures for classification tasks. Inf Process Manag 45(4):427–437
59. Wei W, Li CM, Zhang H (2008) Switching among non-weighting, clause weighting, and variable weighting in local search for SAT. In: Proceedings of the 14th international conference on principles and practice of constraint programming. Springer, Berlin, pp 313–326
60. Wilcoxon F (1945) Individual comparisons by ranking methods. Biom Bull 1(6):80–83
61. Witten IH, Frank E, Hall MA (2011) Data mining: practical machine learning tools and techniques, 3rd edn. Morgan Kaufmann Publishers Inc., San Francisco
62. Wotawa F (2014) Failure mode and effect analysis for abductive diagnosis. In: Proceedings of the international workshop on defeasible and ampliative reasoning, pp 1–13
63. Xu L, Hoos HH, Leyton-Brown K (2010) Hydra: automatically configuring algorithms for portfolio-based selection. In: Proceedings of the twenty-fourth AAAI conference on artificial intelligence. AAAI Press, pp 210–216
64. Xu L, Hutter F, Hoos HH, Leyton-Brown K (2008) SATZilla: portfolio-based algorithm selection for SAT. J Artif Intell Res 32:565–606

Roxane Koitz-Hristov is a Ph.D. candidate at the Institute for Software Technology at Graz University of Technology. She obtained her Master's degrees in Software Development and Business Management as well as Computer Science from Graz University of Technology in 2013 and 2016 respectively. In her research, she mainly focuses on improvements in regard to abductive model-based diagnosis. In particular she has worked on the Austrian Funding Agency project AMOR dealing with bridging the gap between the theory and application of model-based diagnosis.

Franz Wotawa received a M.Sc. in Computer Science (1994) and a PhD in 1996 both from the Vienna University of Technology. He is currently professor of software engineering at the Graz University of Technology. Since the founding of the Institute for Software Technology in 2003 to the year 2009 Franz Wotawa had been the head of the institute. His research interests include model-based and qualitative reasoning, theorem proving, mobile robots, verification and validation, and software testing and debugging. Beside theoretical foundations he has always been interested in closing the gap between research and practice. For this purposes he founded Softnet Austria in 2006, which is a nonprofit organization carrying out applied research projects together with companies. Starting from October 2017, Franz Wotawa is the head of the Christian Doppler Laboratory for Quality Assurance Methodologies for Cyber-Physical Systems. During his career Franz Wotawa has written more than 330 papers for journals, books, conferences, and workshops. He supervised 84 master and 34 PhD students. For his work on diagnosis he received the Lifetime Achievement Award of the Intl. Diagnosis Community in 2016. Franz Wotawa has been member of a various number of program committees and organized several workshops and special issues of journals. He is a member of the Academia Europaea, the IEEE Computer Society, ACM, the Austrian Computer Society (OCG), and the Austrian Society for Artificial Intelligence and a Senior Member of the AAAI.