Memory-Limited Model-Based Diagnosis (Extended Abstract)*

Patrick Rodler
University of Klagenfurt, Austria
patrick.rodler@aau.at

Abstract

Model-based diagnosis is a principled and broadly applicable AI-based approach to tackle debugging problems in a wide range of areas including software, knowledge bases, circuits, cars, and robots. Whenever the sound and complete computation of fault explanations in a given preference order (e.g., wrt. cardinality or probability) is required, all existing diagnosis algorithms suffer from an exponential space complexity. This can prevent their application on memory-restricted devices and for memory-intensive problem cases. As a remedy, we propose RBF-HS, a diagnostic search based on Korf's seminal RBFS algorithm which can enumerate an arbitrary fixed number of fault explanations in best-first order within linear space bounds, without sacrificing other desirable properties. Evaluations on real-world diagnosis cases show that RBF-HS, when used to compute minimum-cardinality fault explanations, in most cases saves substantial space while requiring only reasonably more or even less time than Reiter's HS-Tree, one of the most influential diagnostic algorithms with the same properties.

1 Introduction and Preliminaries

1.1 Model-Based Diagnosis

Model-Based Diagnosis [Reiter, 1987] is a well-founded, principled and broadly applicable approach to detecting, finding and fixing faults in numerous types of systems, such as software [Hunt, 1998], recommender systems [Felfernig *et al.*, 2007], spreadsheets [Jannach and Schmitz, 2016], ontologies [Shchekotykhin *et al.*, 2012], knowledge bases [Rodler, 2015], hardware [Friedrich *et al.*, 1999], circuits [de Kleer and Williams, 1987], robots [Zaman *et al.*, 2013], scheduling problems [Rodler *et al.*, 2021], cars [Sachenbacher *et al.*, 1998], and aircrafts [Gorinevsky *et al.*, 2002].

Technically, model-based diagnosis assumes a system (e.g., software, circuit, knowledge base, physical device) consisting of a set of *components* $\textsc{comps} = \{c_1, \dots, c_n\}$ (e.g., lines of code, gates, axioms, physical constituents) which is formally described in some monotonic logical language. Beside any relevant general knowledge about the system, this *system description* $\textsc{sd}$ includes a specification of the normal behavior (logical sentence $\textsc{beh}(c_i)$) of all components $c_i \in \textsc{comps}$ of the form $\textsc{ok}(c_i) \to \textsc{beh}(c_i)$. As a result, when assuming all components to be fault-free, i.e., $\textsc{ok}(\textsc{comps}) := \{\textsc{ok}(c_1), \dots, \textsc{ok}(c_n)\}$, conclusions about the normal system behavior can be drawn by means of theorem provers. When the real system behavior, ascertained through *system observations* and/or *system measurements* (stated as logical sentences $\textsc{obs}$ and $\textsc{meas}$), is inconsistent with the system behavior predicted by $\textsc{sd}$, the normality-assumption for some components has to be retracted. We call $\langle \textsc{sd}, \textsc{comps}, \textsc{obs}, \textsc{meas} \rangle$ a *diagnosis problem instance (DPI)*.

1.2 Diagnoses

Given a DPI, one of the most central goals in model-based diagnosis is the localization of the actually faulty system components, e.g., the code lines that have to be modified in order for the software to produce the right output, or the parts of a car that have to be repaired or replaced in order for the car to start. One step to this end is the enumeration of sets of potentially faulty components, called diagnoses. Formally, a *(minimal / minimum-cardinality) diagnosis* is an (irreducible / minimal-cardinality) set of components $\mathcal{D} \subseteq \textsc{comps}$ such that $\textsc{sd} \cup \textsc{obs} \cup \textsc{meas} \cup \textsc{ok}(\textsc{comps} \setminus \mathcal{D}) \cup \textsc{nok}(\mathcal{D})$ is consistent where $\textsc{nok}(X) := \{\neg\textsc{ok}(c_i) \mid c_i \in X\}$. So, a diagnosis is a set of components whose abnormality would explain the observed incorrect system behavior. To deal with the potentially numerous diagnoses for a given system (cf., e.g., [Shchekotykhin *et al.*, 2012]), it can often be pivotal to

1. ascertain diagnoses in order from most to least plausible, e.g., minimal cardinality or maximal probability first [de Kleer, 1991; Reiter, 1987] (*best-first diagnosis computation*), and/or
2. acquire additional knowledge about the system, e.g., in terms of measurements, to eliminate spurious diagnoses [de Kleer and Williams, 1987] (*sequential diagnosis*).

Note, the novel technique suggested in this work is applicable for both tasks 1 and 2 (see Sections 2 and 3).

1.3 Diagnosis Computation

Diagnoses are often computed with the aid of conflicts. A *(minimal) conflict* is an (irreducible) set of components $\mathcal{C} \subseteq$

*See [Rodler, 2022c] for the full paper.

COMPS such that assuming all of them fault-free, i.e., $\text{OK}(\mathcal{C})$, is inconsistent with the current knowledge about the system, i.e., $\text{SD} \cup \text{OBS} \cup \text{MEAS} \cup \text{OK}(\mathcal{C}) \models \bot$. Diagnoses and conflicts are related in terms of a *hitting set property*: A (minimal) diagnosis is a (minimal) hitting set of all minimal conflicts. (X is a *hitting set* of a collection of sets $\mathbf{S}$ iff $X \subseteq \bigcup_{S \in \mathbf{S}} S$ and $X \cap S \neq \emptyset$ for all $S \in \mathbf{S}$.) For complexity and efficiency reasons, diagnosis computation is usually focused on minimal diagnoses only.

Given a DPI $\langle \text{SD}, \text{COMPS}, \text{OBS}, \text{MEAS} \rangle$, a *generic* (hitting-set-based) diagnosis search algorithm works as follows:

- Start with a queue including only the root node $\emptyset$.
- While the queue is non-empty and not enough minimal diagnoses have been found, poll the first node n from the queue and process it. That is, *compute a label L for* n, and, *based on L, assign* n (or potentially its successors) to an appropriate node class (e.g., solutions, non-solutions).

Different *specific* diagnosis computation algorithms are obtained by (re)defining *(i)* the sorting of the queue and *(ii)* the node processing procedure (node labeling and assignment).

1.4 A Prominent Example: HS-Tree

An instance of an influential and important diagnostic search is HS-Tree [Reiter, 1987]. It uses a FIFO-queue (breadth-first search) and an initially empty list $\mathbf{D}$ to store the found minimal diagnoses, and defines node labeling and assignment as follows:

1. If n is a non-minimal diagnosis (superset of some already found minimal diagnosis in $\mathbf{D}$) or a duplicate (set-equal to some other node in the queue), then it is labeled with $\times$ (leaf node; irrelevant node; discard n).
2. Else, if there is a minimal conflict $\mathcal{C}$ such that $\mathsf{n} \cap \mathcal{C} = \emptyset$, then n is labeled by $\mathcal{C}$ (internal node). This results in $|\mathcal{C}|$ successor nodes of n that are added to the queue, each constructed as $\mathsf{n} \cup \{c_i\}$, for all $c_i \in \mathcal{C}$. Note that the computation of each conflict requires $O(|\text{COMPS}|)$ theorem prover calls, and can be accomplished, e.g., by the QuickXplain algorithm [Junker, 2004; Rodler, 2022a].
3. Else, n is labeled with $\checkmark$ (leaf node; minimal diagnosis; add n to the list of solutions $\mathbf{D}$).

After the tree is completed (queue is empty), $\mathbf{D}$ includes exactly all minimal diagnoses for the given DPI, sorted by cardinality. Other sortings of $\mathbf{D}$ (e.g., based on diagnosis probability) can be obtained by sorting the queue using suitable cost functions (uniform-cost HS-Tree [Rodler, 2015]).

1.5 Existing Diagnostic Algorithms

Literature covers a wide variety of diagnosis computation techniques, e.g., [Abreu and van Gemund, 2009; Darwiche, 2001; de Kleer, 2011; Feldman *et al.*, 2008; Felfernig *et al.*, 2011; Lin and Jiang, 2003; Metodi *et al.*, 2014; Rodler and Herold, 2018; Rodler, 2020a; Shchekotykhin *et al.*, 2014; Torasso and Torta, 2006; Zhao and Ouyang, 2015]; see [Rodler, 2023; Rodler, 2022b] for surveys, and, e.g., [Nica *et al.*, 2013; Rodler, 2022e] for empirical comparisons. Overall, these methods are highly heterogeneous in terms of their properties [Rodler, 2022b], motivated by different problem cases, domains and requirements for which they were devised.

Desirable and often also necessary properties of diagnostic algorithms are that only and all minimal diagnoses are found (*soundness* and *completeness*), that diagnoses are enumerated in order as per some preference criterion, e.g., maximal probability or minimal cardinality (*best-first property*), and that the algorithm is applicable to any DPI regardless of the problem domain, used logic and chosen theorem prover (*generality*). Unfortunately, however, all existing diagnosis algorithms featuring these four properties require a worst-case exponential amount of memory. This can prevent their successful adoption for memory-intensive problem cases [Shchekotykhin *et al.*, 2014] or for memory-limited (e.g., IoT) devices.

2 Contribution

2.1 New Approach: RBF-HS

As a remedy to this issue, we propose a diagnostic search called Recursive Best-First Hitting Set Search (RBF-HS) based on Korf's seminal RBFS algorithm [Korf, 1993]. RBFS is a path-finding search that implements a scheme that can be synopsized as

- *(complete and best-first):* always expand current globally best node while storing current globally second-best node,
- *(undo and forget to keep space linear):* backtrack and explore second-best node if none of the child nodes of best node is better than second-best,
- *(remember utility of forgotten subtrees to keep the search progressing):* before deleting a subtree in the course of backtracking, store cost of subtree's best node,
- *(restore utility at regeneration to avoid redundancy):* when re-exploring a subtree, use this stored cost value to update node costs in the subtree.

To devise RBF-HS, we first analyzed which general aspects make diagnosis searches different from path-finding searches. In this regard, we identified, e.g., the necessity of defining a suitable node labeling and assignment strategy, that solutions are sets and not paths, that multiple solutions are generally sought, that different conditions on the cost functions have to apply, and that certain provisions are necessary to guarantee diagnostic soundness and completeness. We then modified RBFS accordingly to account for all these differences.

So, roughly, RBF-HS integrates the search strategy of RBFS with the general principles of hitting-set-based diagnosis searches discussed in Section 1. As a result, RBF-HS is sound, complete, best-first, general, *and linear-space*; a combination of features no existing diagnostic technique offers. More specifically, RBF-HS allows to generate an arbitrary fixed number of minimal diagnoses in best-first order within linear space bounds, can be used out of the box for diagnosis problems expressed in any monotonic knowledge representation language, and can operate with any theorem prover.

2.2 Application Scope

Notably, RBF-HS is applicable to optimal (minimal) hitting-set computation (e.g., [Gainer-Dewar and Vera-Licona, 2017]) in general. Due to the relevance of hitting-set computation to many other important fields, the application scope of RBF-HS spans far beyond the domain of model-based diagnosis. In particular, it may find fruitful application for, e.g., (Max)SAT [Davies and Bacchus, 2011], constraint satisfaction [Bailey and Stuckey, 2005] and optimization problems [Saikko, 2019], set-theoretic duality [Slaney, 2014], explainable AI [Ignatiev, 2020], social and life sciences [Amburg *et al.*, 2021], as well as for solving a wide range of other NP-complete problems [Moreno-Centeno and Karp, 2013].

3 Evaluation

3.1 Domain

We conducted comprehensive experiments on 12 real-world DPIs from a collection of widely used benchmark problems in the knowledge-based systems domain (cf., e.g., [del Vescovo *et al.*, 2010; Kalyanpur, 2006; Qi and Hunter, 2007; Rodler, 2022d; Shchekotykhin *et al.*, 2012; Stuckenschmidt, 2008]). In this field, *soundness* and *completeness* are required to guarantee the localization of the *actually* faulty knowledge in often critical (e.g., medical) applications; *generality* is pivotal to deal with a myriad of different logics and theorem provers that are used to optimally trade off expressivity against reasoning complexity [Baader *et al.*, 2007]; and *best-first computation* is desired to monitor the most relevant fault explanations in order to terminate the debugging early if the actual fault is recognized, and, moreover, can boost the overall diagnostic efficiency [Rodler, 2022e]. For these reasons, HS-Tree, described in Section 1, which is a state-of-the-art method featuring these properties, is the commonly used method in this application area.

3.2 Experiments

In our experiments, we thus compared RBF-HS against HS-Tree. We considered two algorithm application scenarios: single-shot and sequential diagnosis. In the single-shot tests, each algorithm had to compute the k best diagnoses for each DPI. In the sequential diagnosis [de Kleer and Williams, 1987] tests, per DPI, each algorithm had to compute k best diagnoses multiple times in an iterative diagnosis session, each time for a different version (including one more measurement) of the given DPI. Each session was executed until all but one minimal diagnosis were ruled out by the measurements; these were selected based on the well-known information gain heuristic (cf., e.g., [de Kleer and Williams, 1987; Rodler, 2017; Rodler and Schmid, 2018; Siddiqi and Huang, 2011]). We used $k \in \{2, 6, 10, 20\}$, defined the "best" diagnoses to be the ones of minimal cardinality, and adopted Pellet [Sirin *et al.*, 2007] as a theorem prover.

3.3 Results

Fig. 1 shows the results for the sequential diagnosis tests (we got almost identical results for the single-shot tests). The main insights are:

- Whenever a DPI was non-trivial to solve, RBF-HS traded space favorably for time compared to HS-Tree (blue bars higher than orange ones).
- Space savings (blue bars) of RBF-HS were significant, amounting to an avg. / max. of 93 % / 98 % of the memory consumed by HS-Tree. Time overheads (orange bars) of RBF-HS, in contrast, remained reasonable in all cases.

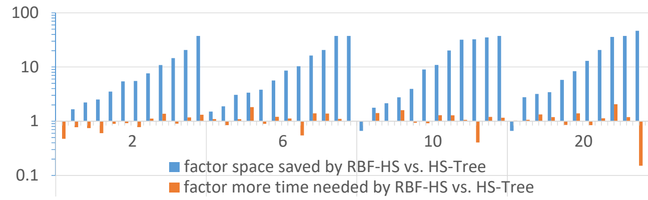

Figure 1: Experiment Results: *(y-axis)* memory reduction and time overhead factors of RBF-HS vs. HS-Tree. *(x-axis)* 12 studied DPIs (sorted from low to high space savings) for each number of computed diagnoses $k \in \{2, 6, 10, 20\}$. Note: The full paper [Rodler, 2022c] provides much more comprehensive evaluation results.

- In 38 % of the cases, RBF-HS exhibited *both* a lower runtime *and* a lower space consumption than HS-Tree (upward blue bars and downward orange bars). We even observed 85 % runtime along with 98 % memory savings in one case.

Additional findings (not shown in Fig. 1) are:

- RBF-HS scales well to large numbers of computed diagnoses and to problems involving high-cardinality minimal diagnoses [Shchekotykhin *et al.*, 2014].
- RBF-HS can deal with highly expressive knowledge representation languages, which makes it well suited, e.g., for ontology quality assurance and debugging [Kalyanpur, 2006; Meilicke, 2011; Rodler, 2015; Shchekotykhin *et al.*, 2012], important research fields in the context of the Semantic Web [Berners-Lee *et al.*, 2001].
- In several cases, HS-Tree ran out of memory (32 GB) while RBF-HS could successfully solve the diagnosis problem under negligible memory requirements.

4 Conclusion

We have proposed a novel diagnostic search based on Korf's seminal RBFS algorithm which gives theoretical guarantees (soundness, completeness, best-first property, generality, linear space complexity) no other diagnostic method does. In experiments on real-world cases, our approach proved to be significantly more efficient wrt. memory consumption and almost on par wrt. runtime, compared to a highly influential and widely used diagnosis algorithm with the same properties. Importantly, RBF-HS is applicable to any hitting-set problem, and thus has the potential to positively impact a number of important domains beyond diagnosis, such as (Max)SAT, (Max)CSP, explainable AI, life sciences, or NP-complete problem solving.

5 Remarks and Additional Resources

Restricted Scope: For brevity, this extended abstract outlines the contributions of the associated full paper [Rodler, 2022c] only partially. For instance, the full paper presents a second diagnostic algorithm, called HBF-HS (Hybrid Best-First Hitting Set Search). In a nutshell, it constitutes a generalization of RBS-HS which combines the latter with HS-Tree in order to optimize the diagnosis computation time (of RBF-HS) while still preserving problem solvability by preventing a memory overflow. Besides all complexity analyses and correctness proofs, the full paper moreover contains a much more comprehensive evaluation, including results for HBF-HS, experiments using a second measurement selection heuristic as well as an alternative diagnosis preference criterion (maximal probability), scalability tests, and a discussion of the performance of the suggested algorithms on particularly hard diagnostic problem cases.

Prior Versions: Earlier versions of the full paper [Rodler, 2022c] associated with this extended abstract were published at the *Int'l Workshop on Principles of Diagnosis 2020* [Rodler, 2020b] and at the *Int'l Symposium on Combinatorial Search 2021* [Rodler, 2021].

Presentations: A short video presentation on this work can be found at https://slideslive.com/38964052, and animated slides can be accessed via http://isbi.aau.at/ontodebug/presentation_rbfhs_dx2020_FINAL.pptx.

Debugging Tool: It is ongoing work to integrate the novel RBF-HS and HBF-HS algorithms into the ontology debugging tool *OntoDebug* [Schekotihin *et al.*, 2018a; Schekotihin *et al.*, 2018b], an official and free plug-in for *Protégé* [Musen, 2015], the currently most popular open-source integrated development environment for ontologies. Please check http://isbi.aau.at/ontodebug/ for further information.

Acknowledgments

This work was funded by the Austrian Science Fund (FWF), contract P-32445-N38. I thank Richard Korf for his seminal work [Korf, 1993] which inspired me to do this research.

References

[Abreu and van Gemund, 2009] Rui Abreu and Arjan J.C. van Gemund. A Low-Cost Approximate Minimal Hitting Set Algorithm and its Application to Model-Based Diagnosis. In *SARA*, 2009.

[Amburg *et al.*, 2021] Ilya Amburg, Jon Kleinberg, and Austin R. Benson. Planted hitting set recovery in hypergraphs. *Journal of Physics: Complexity*, 2(3):035004, 2021.

[Baader *et al.*, 2007] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2007.

[Bailey and Stuckey, 2005] James Bailey and Peter J. Stuckey. Discovery of minimal unsatisfiable subsets of constraints using hitting set dualization. In *PADL*, 2005.

[Berners-Lee *et al.*, 2001] Tim Berners-Lee, James Hendler, and Ora Lassila. The Semantic Web. *Scientific American*, 284(5):34–43, 2001.

[Darwiche, 2001] Adnan Darwiche. Decomposable negation normal form. *Journal of the ACM*, 48(4):608–647, 2001.

[Davies and Bacchus, 2011] Jessica Davies and Fahiem Bacchus. Solving MAXSAT by solving a sequence of simpler SAT instances. In *CP*, 2011.

[de Kleer and Williams, 1987] Johan de Kleer and Brian C. Williams. Diagnosing multiple faults. *Artificial Intelligence*, 32(1):97–130, 1987.

[de Kleer, 1991] Johan de Kleer. Focusing on Probable Diagnoses. In *AAAI*, 1991.

[de Kleer, 2011] Johan de Kleer. Hitting set algorithms for model-based diagnosis. In *DX*, 2011.

[del Vescovo *et al.*, 2010] Chiara del Vescovo, Bijan Parsia, Ulrike Sattler, and Thomas Schneider. The modular structure of an ontology: an empirical study. In *WoMO*, 2010.

[Feldman *et al.*, 2008] Alexander Feldman, Gregory Provan, and Arjan J.C. van Gemund. Computing Minimal Diagnoses by Greedy Stochastic Search. In *AAAI*, 2008.

[Felfernig *et al.*, 2007] Alexander Felfernig, Gerhard Friedrich, and Erich Teppan. Automated Debugging and Repair of Utility Constraints in Recommender Knowledge Bases. In *DX*, 2007.

[Felfernig *et al.*, 2011] Alexander Felfernig, Monika Schubert, and Christoph Zehentner. An efficient diagnosis algorithm for inconsistent constraint sets. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 26(1):53–62, 2011.

[Friedrich *et al.*, 1999] Gerhard Friedrich, Markus Stumptner, and Franz Wotawa. Model-based diagnosis of hardware designs. *Artificial Intelligence*, 111(1-2):3–39, 1999.

[Gainer-Dewar and Vera-Licona, 2017] Andrew Gainer-Dewar and Paola Vera-Licona. The minimal hitting set generation problem: algorithms and computation. *SIAM Journal on Discrete Mathematics*, 31(1):63–100, 2017.

[Gorinevsky *et al.*, 2002] Dimitry Gorinevsky, Kevin Dittmar, Dinkar Mylaraswamy, and Emmanuel Nwadiogbu. Model-based diagnostics for an aircraft auxiliary power unit. In *CCA*, 2002.

[Hunt, 1998] John Hunt. Model-based software diagnosis. *Applied Artificial Intelligence*, 12(4):289–308, 1998.

[Ignatiev, 2020] Alexey Ignatiev. Towards Trustable Explainable AI. In *IJCAI*, 2020.

[Jannach and Schmitz, 2016] Dietmar Jannach and Thomas Schmitz. Model-based diagnosis of spreadsheet programs: a constraint-based debugging approach. *Automated Software Engineering*, 23:105–144, 2016.

[Junker, 2004] Ulrich Junker. QuickXplain: Preferred Explanations and Relaxations for Over-Constrained Problems. In *AAAI*, 2004.

[Kalyanpur, 2006] Aditya Kalyanpur. *Debugging and Repair of OWL Ontologies*. PhD Thesis, Univ. of Maryland, College Park, 2006.

[Korf, 1993] Richard E. Korf. Linear-space best-first search. *Artificial Intelligence*, 62(1):41–78, 1993.

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

[Meilicke, 2011] Christian Meilicke. *Alignment Incoherence in Ontology Matching*. PhD Thesis, Univ. of Mannheim, 2011.

[Metodi *et al.*, 2014] Amit Metodi, Roni Stern, Meir Kalech, and Michael Codish. A novel SAT-based approach to model based diagnosis. *Journal of Artificial Intelligence Research*, 51:377–411, 2014.

[Moreno-Centeno and Karp, 2013] Erick Moreno-Centeno and Richard M. Karp. The implicit hitting set approach to solve combinatorial optimization problems with an application to multigenome alignment. *Operations Research*, 61(2):453–468, 2013.

[Musen, 2015] Mark A Musen. The Protégé project: a look back and a look forward. *AI matters*, 1(4):4–12, 2015.

[Nica *et al.*, 2013] Iulia Nica, Ingo Pill, Thomas Quaritsch, and Franz Wotawa. The route to success: A performance comparison of diagnosis algorithms. In *IJCAI*, 2013.

[Qi and Hunter, 2007] Guilin Qi and Anthony Hunter. Measuring incoherence in description logic-based ontologies. In *ISWC*, 2007.

[Reiter, 1987] Raymond Reiter. A Theory of Diagnosis from First Principles. *Artificial Intelligence*, 32(1):57–95, 1987.

[Rodler and Herold, 2018] Patrick Rodler and Manuel Herold. StaticHS: A Variant of Reiter's Hitting Set Tree for Efficient Sequential Diagnosis. In *SoCS*, 2018.

[Rodler and Schmid, 2018] Patrick Rodler and Wolfgang Schmid. On the Impact and Proper Use of Heuristics in Test-Driven Ontology Debugging. In *RuleML+RR*, 2018.

[Rodler *et al.*, 2021] Patrick Rodler, Erich Teppan, and Dietmar Jannach. Randomized Problem-Relaxation Solving for Over-Constrained Schedules. In *KR*, 2021.

[Rodler, 2015] Patrick Rodler. *Interactive Debugging of Knowledge Bases*. PhD Thesis, Univ. of Klagenfurt, 2015.

[Rodler, 2017] Patrick Rodler. On Active Learning Strategies for Sequential Diagnosis. In *DX*, 2017.

[Rodler, 2020a] Patrick Rodler. Reuse, Reduce and Recycle: Optimizing Reiter's HS-Tree for Sequential Diagnosis. In *ECAI*, 2020.

[Rodler, 2020b] Patrick Rodler. Sound, Complete, Linear-Space, Best-First Diagnosis Search. In *DX*, 2020.

[Rodler, 2021] Patrick Rodler. Linear-Space Best-First Diagnosis Search. In *SoCS*, 2021.

[Rodler, 2022a] Patrick Rodler. A formal proof and simple explanation of the QuickXplain algorithm. *Artificial Intelligence Review*, 55(8):6185–6206, 2022.

[Rodler, 2022b] Patrick Rodler. How should I compute my candidates? A taxonomy and classification of diagnosis computation algorithms. In *DX*, 2022.

[Rodler, 2022c] Patrick Rodler. Memory-limited model-based diagnosis. *Artificial Intelligence*, 305:103681, 2022.

[Rodler, 2022d] Patrick Rodler. One step at a time: An efficient approach to query-based ontology debugging. *Knowledge-Based Systems*, 251:108987, 2022.

[Rodler, 2022e] Patrick Rodler. Random vs. Best-First: Impact of Sampling Strategies on Decision Making in Model-Based Diagnosis. In *AAAI*, 2022.

[Rodler, 2023] Patrick Rodler. DynamicHS: Streamlining Reiter's hitting-set tree for sequential diagnosis. *Information Sciences*, 627:251–279, 2023.

[Sachenbacher *et al.*, 1998] Martin Sachenbacher, Andress Malik, and Peter Struss. From electrics to emissions: experiences in applying model-based diagnosis to real problems in real cars. In *DX*, 1998.

[Saikko, 2019] Paul Saikko. *Implicit hitting set algorithms for constraint optimization*. PhD thesis, Univ. of Helsinki, 2019.

[Schekotihin *et al.*, 2018a] Konstantin Schekotihin, Patrick Rodler, and Wolfgang Schmid. Ontodebug: Interactive ontology debugging plug-in for Protégé. In *FoIKS*, 2018.

[Schekotihin *et al.*, 2018b] Konstantin Schekotihin, Patrick Rodler, Wolfgang Schmid, Matthew Horridge, and Tania Tudorache. Test-Driven Ontology Development in Protégé. In *ICBO*, 2018.

[Shchekotykhin *et al.*, 2012] Kostyantyn Shchekotykhin, Gerhard Friedrich, Philipp Fleiss, and Patrick Rodler. Interactive Ontology Debugging: Two Query Strategies for Efficient Fault Localization. *Web Semantics: Science, Services and Agents on the World Wide Web*, 12-13:88–103, 2012.

[Shchekotykhin *et al.*, 2014] Kostyantyn Shchekotykhin, Gerhard Friedrich, Patrick Rodler, and Philipp Fleiss. Sequential diagnosis of high cardinality faults in knowledge-bases by direct diagnosis generation. In *ECAI*, 2014.

[Siddiqi and Huang, 2011] Sajjad Siddiqi and Jinbo Huang. Sequential diagnosis by abstraction. *Journal of Artificial Intelligence Research*, 41:329–365, 2011.

[Sirin *et al.*, 2007] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. Pellet: A practical OWL-DL reasoner. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(2):51–53, 2007.

[Slaney, 2014] John Slaney. Set-theoretic duality: A fundamental feature of combinatorial optimisation. In *ECAI*, 2014.

[Stuckenschmidt, 2008] John Slaney. Debugging OWL Ontologies-A Reality Check. In *EON*, 2008.

[Torasso and Torta, 2006] Pietro Torasso and Gianluca Torta. Model-based diagnosis through OBDD compilation: A complexity analysis. In *Reasoning, Action and Interaction in AI Theories and Systems*, 2006.

[Zaman *et al.*, 2013] Safdar Zaman, Gerald Steinbauer, Johannes Maurer, Peter Lepej, and Suzana Uran. An integrated model-based diagnosis and repair architecture for ROS-based robot systems. In *ICRA*, 2013.

[Zhao and Ouyang, 2015] Xiangfu Zhao and Dantong Ouyang. Deriving all minimal hitting sets based on join relation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(7):1063–1076, 2015.