Linear-Space Best-First Diagnosis Search*

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

Abstract

Various model-based diagnosis scenarios require the computation of the most preferred fault explanations. Existing algorithms that are sound (i.e., output *only* actual fault explanations) and complete (i.e., can return *all* explanations), however, require exponential space to achieve this task. As a remedy, to enable successful diagnosis on memory-restricted devices and for memory-intensive problem cases, we propose RBF-HS, a diagnostic search based on Korf's seminal RBFS algorithm. RBF-HS can enumerate an arbitrary fixed number of fault explanations in best-first order within linear space bounds, without sacrificing the desirable soundness or completeness properties. Evaluations using real-world diagnosis cases show that RBF-HS, when used to compute minimum-cardinality fault explanations, in most cases saves substantial space (up to 98 %) while requiring only reasonably more or even less time than Reiter's HS-Tree, one of the most widely used diagnostic algorithms with the same properties.

Model-Based Diagnosis (Reiter 1987) assumes a system (e.g., software, circuit, knowledge base, physical device) consisting of a set of *components* $\text{COMPS} = \{c_1, \dots, c_n\}$ (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* SD includes a specification of the normal behavior (logical sentence $\text{BEH}(c_i)$) of all components $c_i \in \text{COMPS}$ of the form $\text{OK}(c_i) \rightarrow \text{BEH}(c_i)$. As a result, when assuming all components to be fault-free, i.e., $\text{OK}(\text{COMPS}) := \{\text{OK}(c_1), \dots, \text{OK}(c_n)\}$, 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 OBS and MEAS), is inconsistent with the system behavior predicted by SD, the normality-assumption for some of the components has to be retracted. We call $\langle \text{SD}, \text{COMPS}, \text{OBS}, \text{MEAS} \rangle$ a *diagnosis problem instance (DPI)*. A *(minimal / minimum-cardinality) diagnosis* is an (irreducible / minimal-cardinality) set of components $\mathcal{D} \subseteq \text{COMPS}$ such that $\text{SD} \cup \text{OBS} \cup \text{MEAS} \cup \text{OK}(\text{COMPS} \setminus \mathcal{D}) \cup \text{NOK}(\mathcal{D})$ is consistent where $\text{NOK}(X) := \{\neg\text{OK}(c_i) \mid c_i \in X\}$. So, a diagnosis is a set of components whose abnormality would explain the observed incorrect system behavior.

Diagnosis Computation is often accomplished with the aid of conflicts. A *(minimal) conflict* is an (irreducible) set of components $\mathcal{C} \subseteq \text{COMPS}$ 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).

A prominent instance of a diagnostic search is HS-Tree (Reiter 1987). It uses a FIFO-queue (breadth-first search), and defines node labeling and assignment as follows:

1. If n is a non-minimal diagnosis (superset of some already found minimal diagnosis) 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 2020c).
3. Else, n is labeled with ✓ (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 prob-

*This work was supported by the Austrian Science Fund (FWF), contract P-32445-N38. See (Rodler 2020a,b) for extended versions including all details, further evaluations and proofs.

ability) can be obtained by sorting the queue using suitable cost functions (uniform-cost HS-Tree (Rodler 2015)).

Properties of Diagnostic Algorithms. Literature covers a wide variety of diagnosis computation algorithms with very different properties. This heterogeneity is motivated by different problem cases, domains and requirements. Desirable and sometimes 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 particular logic and theorem prover used (*generality*). Unfortunately, however, all existing diagnosis algorithms featuring these four properties require a worst-case exponential amount of memory. This can prevent their successful use on memory-intensive problem cases (Shchekotykhin et al. 2014) or on memory-limited (e.g., IoT) devices.

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 above. 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.

Evaluation. We conducted extensive experiments on a benchmark of 12 real-world DPIs from the knowledge-based systems domain. 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 and Elichanova 2020). For these reasons, HS-Tree, which is a state-of-the-art method featuring these properties, is the commonly used method in this application area.

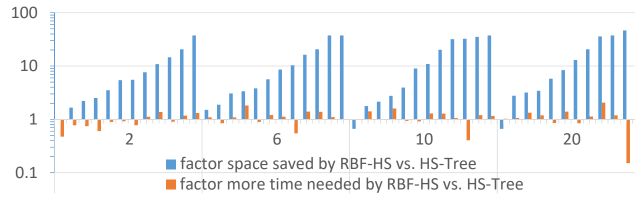

Figure 1: Experiment Results: *(y-axis)* memory reduction and time overhead factors of RBF-HS vs. HS-Tree. *(x-axis)* 12 studied DPIs (sorted from low to high space savings) for each number of computed diagnoses $k \in \{2, 6, 10, 20\}$.

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. 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.

Fig. 1 shows the results for the sequential diagnosis tests (we got almost identical results for the single-shot tests). The insights are: *(i)* 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). *(ii)* 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. *(iii)* In 38 % of the cases, RBF-HS in fact exhibited *both* a lower runtime *and* a lower space consumption than HS-Tree. We even observed 85 % runtime along with 98 % memory savings in one case.

Conclusions. 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 widely used diagnosis algorithm with the same properties. Hence, this work once again shows that ideas and concepts from the heuristic search domain can be fruitfully exploited beyond the boundaries of classic search applications.

References

Baader, F.; Calvanese, D.; McGuinness, D. L.; Nardi, D.; and Patel-Schneider, P. F., eds. 2007. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press.

de Kleer, J.; and Williams, B. C. 1987. Diagnosing multiple faults. *Artificial Intelligence* 32(1): 97–130.

Junker, U. 2004. QuickXplain: Preferred Explanations and Relaxations for Over-Constrained Problems. In *AAAI Conference on Artificial Intelligence (AAAI)*.

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

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

Rodler, P. 2015. *Interactive Debugging of Knowledge Bases*. Ph.D. thesis, University of Klagenfurt. arXiv:1605.05950.

Rodler, P. 2020a. Memory-Limited Model-Based Diagnosis. arXiv:2010.04282.

Rodler, P. 2020b. Sound, Complete, Linear-Space, Best-First Diagnosis Search. In *Int'l Workshop on Principles of Diagnosis (DX)*. arXiv:2009.12190.

Rodler, P. 2020c. Understanding the QuickXPlain Algorithm: Simple Explanation and Formal Proof. arXiv:2001.01835.

Rodler, P.; and Elichanova, F. 2020. Do We Really Sample Right In Model-Based Diagnosis? In *Int'l Workshop on Principles of Diagnosis (DX)*. arXiv:2009.12178.

Shchekotykhin, K.; Friedrich, G.; Rodler, P.; and Fleiss, P. 2014. Sequential diagnosis of high cardinality faults in knowledge-bases by direct diagnosis generation. In *European Conference on Artificial Intelligence (ECAI)*, 813–818.

Sirin, E.; Parsia, B.; Grau, B. C.; Kalyanpur, A.; and Katz, Y. 2007. Pellet: A practical OWL-DL reasoner. *Web Semantics: Science, Services and Agents on the World Wide Web* 5(2): 51–53.