Approximating All Most-Preferred Diagnoses using Greedy Algorithms^{*}

Gregory Provan*

* Department of Computer Science, University College Cork, Cork, Ireland (Tel: 353-21-490-1816; e-mail: g.provan@ cs.ucc.ie).

Abstract: For applications in which computing the set of all diagnoses is important, algorithms that compile the system model, such as the ATMS, are well-known. However, the spaceand time-complexity of such approaches can be prohibitively large. We show how we can use a preference function over the space of diagnoses to develop approximation algorithms for computing the most-preferred diagnoses (MPD). We show that the MPD problem can be approximated by greedy algorithms such that the preference weighting of the approximate set of diagnoses is within $(1 - \frac{1}{e})$ of that of the exact set of diagnoses. We demonstrate how the MPD problem can be solved for iterative diagnosis using a fast stochastic diagnosis engine, and by compilation approaches using prime implicant and DNNF compilation languages. We present empirical evidence that the compilation algorithms enable space reductions of several orders-of-magnitude over the full compilation, while losing relatively little query completeness.

1. INTRODUCTION

A variety of applications require computing the set of all diagnoses. All diagnoses can be computed using compilation approaches, such as the DNNF [Darwiche [2001]], OBDD [Bryant [1992]], or prime implicate [de Kleer [1986]] encodings; alternatively, one can use a single-diagnosis algorithm multiple times. However, the space- and/or time-complexity of both approaches can be prohibitive. In particular, the size of the compiled representation can be exponentially larger than that of the original function; for DNNF the size is exponential in the treewidth of the interaction graph [Darwiche [2001]].

In many applications, one is really interested in the set of preferred diagnoses, based on a preference relation, e.g., as defined in terms of the probabilistically most-likely diagnoses, or the cardinality-minimal diagnoses. For such applications, we define the most-preferred diagnosis (MPD) problem as the task of computing a subset of diagnoses, given an observation α , which is most-preferred with respect to a preference function ϕ .

In this article we propose a sound but incomplete approximation technique for solving the MPD problem. We assume that we have a preference function that identifies the most likely diagnoses (or satisfying assignments) of the diagnosis model Φ . Given a preference function ϕ over Φ , we calculate an approximate compilation $\zeta(\Phi)$ by pruning diagnoses whose preference value is below a threshold ϱ .

Our contributions are as follows:

- (1) We prove that we can define a submodular function for the MPD problem, based on the subset, setcardinality and probability-ordering preference functions.
- (2) Given that the MPD problem is submodular, we show that we can approximate the set of most-

preferred diagnoses by a set whose preference weight is guaranteed to be within $(1 - \frac{1}{e})$ of the optimal preference weight.

- (3) We show that we can significantly bound the diagnosis space by searching only over the most-preferred diagnosis sub-space.
- (4) We show that the greedy approach can work over a bounded sub-space for both the iterative and compilation diagnosis approaches.
- (5) We empirically validate this approach for two important compilation targets, prime implicants and DNNF, showing that we can achieve space reductions of several orders-of-magnitude over the full compilation, while losing relatively little query completeness.

2. RELATED WORK

This section reviews prior work in related areas.

Several algorithms have been applied to the task of computing all diagnoses. The primary algorithms are all methods that compile the diagnostic system model prior to diagnostic inference. All such algorithms have exponential worst-case space complexity. The complexity of compilation has been addressed in many papers, including [Cadoli et al. [2002]]. Ferrara et al. [2007] have proven that for temporal logic model checking, preprocessing cannot reduce complexity, given polynomial space bounds (in the size of the input) on the output. In the case of compiling using BDDs, one cannot guarantee that a BDD will not be of size exponential in the number of variables, since the problem of finding the best variable ordering is NP-hard [Bollig and Wegener [1996]]. Further, the approximation problem is also hard, as for any constant c > 1 it is NPhard to compute a variable ordering resulting in an OBDD with a size that is at most c times larger than optimal [Sieling [2002]]. In the case of DNNF, the size of the DNNF generated for problems in propositional logic, Bayesian

^{*} Supported by SFI grants 04/IN3/I524 and 06/SRC/I1091.

networks and constraint satisfaction is exponential in the treewidth of their interaction graph [Darwiche [2001]].¹

We extend the approximate compilation work of [Venturini and Provan [2008]] by showing that we can have guarantees on approximate diagnosis for compilations, and showing that similar guarantees exist not just for compiled diagnosis approaches but also for iterative diagnosis algorithms. Our notion of preference-based compilation is different to the use of cost functions for DNNF minimaldiagnosis extraction [Darwiche [1998]]. The cost-function approach aims to compute the most-preferred diagnosis given a complete DNNF and an observation; in our case we are incrementally compiling all diagnoses which are more preferred than a given threshold ρ . Hence, whereas using probabilistic cost-functions can prune a complete DNNF such that valid diagnoses may be lost [Darwiche [1998]], the threshold-based incremental compilation guarantees that no diagnoses $\delta \in \Delta_{\rho}$ will fail to be included in the approximate compilation, as shown in Theorem 2.

3. NOTATION AND PRELIMINARIES

This section introduces our notation. We first define the propositional language we will use, and use this language to specify the two main compilation languages we examine, prime implicants and DNNF. Then we introduce our preferred diagnosis notions.

3.1 Propositional Logic

We assume a standard propositional logic in this paper. We use a finite alphabet of propositional symbols, $\Psi = \{z_1, z_2, ..., z_n\}$, using the usual boolean connectives \land, \lor, \neg , and \Rightarrow for constructing well-formed formulae from Ψ . A literal is a propositional symbol or its negation. A clause $z_1 \lor z_2 \lor \cdots \lor z_k \lor \neg z_{k+1} \lor \cdots \lor \neg z_n$ is a disjunction of literals. A clause is Horn (definite Horn) if $k \leq 1$ (k = 1). A function (or formula) Φ is a conjunction of clauses; in this article we assume that a formula is defined over n symbols, unless stated otherwise. The size of a formula f is |f|.

A (partial) interpretation γ for Ψ , a mapping from (a subset of) Ψ to {true, false}, can be extended to boolean formulae in the usual recursive way. Γ is the set of all interpretations. A *partial solution* is a partial interpretation consistent with f. A *model* of a formula Φ is an interpretation γ that maps Φ to true. \mathcal{B} is the set of all boolean formulae over Ψ . The function $atoms : \mathcal{B} \to 2^{\Psi}$ maps a formula Φ to the set of propositional symbols occurring in Φ .

Prime implicants An implicant I of a formula Φ is a conjunction of literals such that $I \Rightarrow \Phi$. An implicant I is a prime implicant (PI) if, for every conjunct J obtained by removing one or more literals from $I, J \neq \Phi$. In other words, a prime implicant is a minimal implicant of Φ . The disjunction Π of all prime implicants of a formula Φ is equivalent to Φ , i.e. Π preserves the models of Φ . The size of Π is the sum of the size of all prime implicants.

Decomposable Negation Normal Form A formula Φ is in Negation Normal Form (NNF) if its literals are joined using only the operators \vee and \wedge . The Decomposable Negation Normal Form (DNNF) is a subclass of NNF satisfying the decomposability property, i.e., for every conjunction $\alpha = \alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n$ in a DNNF, it holds that $atoms(\alpha_i) \cap atoms(\alpha_j) = \emptyset$ for $i \neq j$. A DNNF can be represented by a rooted directed acyclic graph, where each leaf node is associated with a literal or truth value, and each intermediate node corresponds to either \vee or \wedge .

3.2 Most-Preferred Diagnosis

Central to MBD, a *model* of an artifact is represented as a propositional WFF over some set of variables. Discerning two subsets of these variables as *assumable* and *observable*² variables gives us a diagnostic system.

Definition 1. (Diagnostic System). A diagnostic system \mathcal{D} is defined as the triple $\mathcal{D} = \langle \Phi, \mathcal{C}, \alpha \rangle$, where Φ is a propositional theory over a set of variables $V, \mathcal{C} \subseteq V$, $\alpha \subseteq V, \mathcal{C}$ is the set of assumables, and α is the set of observables.

The traditional query in MBD computes terms of assumable variables which are explanations for the system description and an observation.

Definition 2. (Health Assignment). Given a diagnostic system $\mathcal{D} = \langle \Phi, \mathcal{C}, \alpha \rangle$, an assignment H to all variables in \mathcal{C} is defined as a health assignment.

A health assignment H is a conjunction of propositional literals. In some cases it is convenient to use the set of negative or positive literals in H. These two sets are denoted as $\mathcal{L}^-(H)$ and $\mathcal{L}^+(H)$, respectively. \mathcal{H} is the set of all health assignments.

We now formally define consistency-based diagnosis.

Definition 3. (Diagnosis). Given a diagnostic system $\mathcal{D} = \langle \Phi, \mathcal{C}, \alpha \rangle$, an observation α over some variables in α , and a health assignment δ , δ is a diagnosis iff $\Phi \wedge \alpha \wedge \delta \not\models \perp$.

In the MBD literature, a range of types of "preferred" diagnosis has been proposed. This turns the MBD problem into an optimization problem. We now consider the case where we use a preference function ϕ to guide the choice of *most-preferred* diagnoses.

Definition 4. (Preference Function). A preference function ϕ defined over the diagnoses Δ of Φ defines a partial ordering \succ over Δ . We say that diagnosis $\delta_1 \in \Delta$ is preferred to diagnosis $\delta_2 \in \Delta$, written $\delta_1 \succ \delta_2$, if $\phi(\delta_1) > \phi(\delta_2)$ for comparison function >.

In the following definition we consider the standard notions of minimal, minimal-cardinality (MC) and mostprobable (MP) diagnoses. Table 1 shows the symbols we use to define these preferred diagnoses, using the notions of Definition 1.

Given the ordering induced by ϕ , we must convert this ordering into a weighting function f such that we can

 $^{^1\,}$ It turns out that many real-world problems, e.g., the ISCAS benchmark circuits [Brglez et al. [1989]], do not have treewidths bounded by some relatively small integer, in order to ensure compact DNNF compilations.

 $^{^2\,}$ In the MBD literature the assumable variables are also referred to as "component", "failure-mode", or "health" variables. Observable variables are also called "measurable", or "control" variables.

	Symbol	$\phi(\delta)$	Comparison
Minimal	δ^{\subseteq}	$\mathcal{L}^{-}(\delta^{\subseteq})$	\subseteq
MC	δ^{\leq}	$ \mathcal{L}^{-}(\delta^{\leq}) $	\leq
MP	δ^p	$Pr(\delta^p)$	\geq
TT 1 1 1	DC	1 1	

Table 1. Preferred diagnosis summary

apply appropriate computational machinery to find mostpreferred sets of diagnoses. Using f, we can express our MPD problem as follows:

Problem 1. (Most-Preferred Diagnosis Problem). Given a weight function f induced by preference-ordering ϕ , and observation α , compute a most-preferred subset of diagnoses, which is given by $\Delta^*(\alpha) = \arg \max_{\delta \in \Delta} f(\delta)$.

We are also interested in computing a threshold-bounded set of diagnoses, i.e., the subset of diagnoses whose preference exceeds a bound ρ . We can define this set as follows:

Problem 2. (ρ -Preferred Diagnosis Problem). Given a preference function ϕ over the set of all diagnoses, and an observation α , compute a ρ -preferred diagnosis set $\Delta_{\rho}(\alpha) = \{\delta(\alpha) | \phi(\delta(\alpha)) > \rho\}.$

4. SUBMODULARITY OF MPD

We now show how the submodularity of our MPD problem means that our greedy approach is as accurate as any polytime approximation algorithm. Submodular functions are a key concept in combinatorial optimization Nemhauser et al. [1978]. Intuitively, sub-modularity highlights the notion that, as we select more diagnoses based on preferenceranking, we have diminishing returns for the cumulative preference weight.

In performing inference on a submodular function, we assume that we have a diagnosis evaluation oracle Θ , such that, when given as input a set $H \subseteq \mathcal{H}$, Θ outputs f(H). As a consequence, our submodular function can be solved using polynomially many queries to the oracle Θ .

Definition 5. (Submodularity). A function f(S) defined on subsets $S \subseteq U$ of a universal set U is said to be submodular if for any two sets $A \subset B \subseteq U$ and an element $x \notin B$, we have

$$f(A \cup \{x\}) - f(A) \ge f(B \cup \{x\}) - f(B).$$
(1)

In the following, we show that diagnostic operations based on the preference orderings of subset-inclusion, setcardinality and probability-ordering are submodular. We first define an appropriate weighting function f for each preference ordering, and then show that f is submodular.

Subset-Inclusion: The first lemma that we must prove is that diagnostic operations based on subset-inclusion are submodular. To show this, we must define an appropriate submodular function f(A). Note that subset-inclusion induces a poset over U. We can associate with the ordering on this poset a set of integers, such that for the ordering from $\bot = \{\emptyset\}$, up to \top , we have the corresponding integers 1 up to n; we define this corresponding set of integers as Λ . As a consequence, we can define a function $f(A) : A \to \Lambda$, which has the property that for any $A \subseteq B \subseteq U, f(A) \leq f(B)$.

Lemma 1. The function $f_{\subseteq}(H)$ induced by subset-inclusion \subseteq is submodular.

Proof. We must prove that $f_{\subseteq}(A \cup \{x\}) - f_{\subseteq}(A) \ge f_{\subseteq}(B \cup \{x\}) - f_{\subseteq}(B)$. Taking the left-hand-side, we have $f_{\subseteq}(A \cup \{x\}) - f_{\subseteq}(A) = (f_{\subseteq}(A) + 1) - f_{\subseteq}(A) = 1$. Taking the right-hand-side, we have $f_{\subseteq}(B \cup \{x\}) - f_{\subseteq}(B) = (f_{\subseteq}(B) + 1) - f_{\subseteq}(B) = 1$. Hence we obtain the required Equation 1.

Set Cardinality: We must define an appropriate submodular weight function $f_{\leq}(A)$. If there are m health variables, we can define the weight of a health assignment as $m - |\mathcal{L}^{-}(H)|$. Then the cumulative weight of a set of health assignments is given by $f_{\leq}(\mathcal{H}) = \sum_{H_i \in \mathcal{H}} (m - |\mathcal{L}^{-}(H_i)|)$.

Lemma 2. The function $f_{\leq}(\mathcal{H})$ induced by a max-cardinality ordering is submodular.

Proof. By the definition of submodularity, we must show that $f_{\leq}(A \cup \{x\}) - f_{\leq}(A) \geq f_{\leq}(B \cup \{x\}) - f_{\leq}(B)$. Taking the left-hand-side, we have $f_{\leq}(A \cup \{x\}) - f_{\leq}(A) = f_{\leq}(A) + f_{\leq}(\{x\}) - f_{\leq}(A) = f_{\leq}(\{x\})$. Taking the right-hand-side, we have $f_{\leq}(B \cup \{x\}) - f_{\leq}(B) = f_{\leq}(B) + f_{\leq}(\{x\}) - f_{\leq}(B) = f_{\leq}(\{x\})$. Hence we obtain the required equation 1.

Set Probability-Ordering: For this preference function, we define $f_p(H)$ in terms of the cumulative distribution function (cdf) for a set \mathcal{H} of health assignments. The probability of a health assignment is given by $Pr(H) = \prod_{V_i \in H} Pr(V_i)$; the cdf for a set \mathcal{H} of health assignments is given by $f_p(\mathcal{H}) = \sum_{H \in \mathcal{H}} Pr(H)$.

Lemma 3. The function $f_p(\mathcal{H})$ induced by a max-probability ordering is submodular.

Proof. Assume that we have health assignments $A \in 2^H$ and $B \in 2^H$. Let $Pr(A) = \pi_A$, and $Pr(B) = \pi_B$. By the definition of submodularity, we must show that $f(A \cup \{x\}) - f(A) \ge f(B \cup \{x\}) - f(B)$. Taking the left-hand-side, we have $f_p(A \cup \{x\}) - f_p(A) = \pi_A + \pi_x - \pi_A = \pi_A$. Taking the right-hand-side, we have $f_p(B \cup \{x\}) - f_p(B) = \pi_B + \pi_x - \pi_B = \pi_B$. Hence we obtain the required equation 1.

Next, we employ some well-studied results about the optimality of greedy algorithms on submodular functions. First, it has been shown that a greedy algorithm provides a constant-factor approximation for submodular problems.

Lemma 4. (Nemhauser et al. [1978]). For any normalized, monotonic submodular function f, the set Σ^* obtained by a greedy algorithm achieves at least a constant fraction $1 - \frac{1}{e}$ of the objective value obtained by the optimal solution:

$$f(\Sigma^*) \ge \left(1 - \frac{1}{e}\right) \max_{|S| \le k} f(S),$$

where k is a bound on the number of diagnoses we can address.

Proof. Cf. Nemhauser et al. [1978].

Moreover, Feige [1998] has shown that the $(1 - \frac{1}{e})$ approximation obtained for such problems using a greedy algorithm is optimal unless P = NP. Together, these results indicate the following:

Theorem 1. A greedy algorithm provides a constant-factor approximation to the MPD problem (Problem 1), such that no poly-time algorithm can provide a better approximation unless P = NP.

Proof. Follows from Lemmas 1 through 4, and the [Feige [1998]] result.

5. ITERATIVE MPD COMPUTATION

This section describes how we compute the MPD set using multiple iterations of a single-diagnosis algorithm Θ .

If we assume that we can rank-order all potential diagnoses in terms of their weight, $f(\delta)$, then we can adopt a greedy approach that selects the maximum-weight diagnosis at each step, and checks its consistency using Θ . By Theorem 1, we know that such a greedy algorithm will provide a constant-factor approximation of the weighted diagnosis set generated by an optimal algorithm. The set of approximate diagnoses will be sound if the diagnostic oracle is sound.

Since we will only need a polynomial number of calls to Θ [Feige [1998]], the complexity of this approach is bounded by the complexity of each oracle call. With our propositional language, this diagnostic-oracle task is NPhard, so we employ an incomplete stochastic diagnosis oracle, SAFARI [Feldman et al. [2008]].

We have also developed a second algorithm that allows a user to tune the MPD inference according to a weight threshold ρ , i.e., we solve Problem 2. In this case, we assume that we compute an ordering of the health-assignment subsets using ϕ . Then we check the most-likely health assignments using the diagnostic oracle Θ , as shown in Algorithm 1.

Algorithm 1 Pseudocode for iterative ρ -MPD.			
1: fu	nction MPD _{ϱ} ($\mathcal{D}, \alpha, \varrho, \mathcal{H}$) returns a set Δ_{ϱ}		
	inputs: \mathcal{D} , MPD system; α , observation;		
	ϱ , weight threshold; \mathcal{H} , health set		
	local variables: Δ_{ϱ} , set of terms, initially \emptyset		
2:	$H \leftarrow \arg \max\{f(H) H \in \mathcal{H}\}$		

3:	repeat
4:	If $\Phi \cup \alpha \cup H \not\models \bot$ then $\Delta_{\rho} \leftarrow \Delta_{\rho} \cup H$
5:	$\mathcal{H} \leftarrow \mathcal{H} \setminus H$
6:	until $f(H) < \varrho$
7:	$\mathbf{return}\ \Delta_{\varrho}$
8:	end function

There are several optimizations that can be employed to reduce the computational overhead of this iterative approach. The most important concerns bounding the space of possible diagnoses, which can be $O(2^n)$ for binaryvalued health variables, and larger for multi-valued variables. In real-world domains, the failure probabilities of components are typically very small—on the order of 10^{-2} or smaller—so health assignments with many faulty components have a very low probability of occurring, and hence contribute marginally to the cdf $f(\mathcal{H})$. With analogous arguments possible for the other preference orderings, one can strictly bound the space of possible diagnoses to \mathcal{H}_{ϱ} through an appropriate threshold ϱ , with minimal effect on the ratio $\frac{f(\mathcal{H}_{\varrho})}{f(\mathcal{H})}$. This limits the potential number of calls to the diagnosis oracle, which is the primary computational bottleneck. Second, using a fast but incomplete oracle creates significant speedups over a complete oracle. Using an incomplete oracle like SAFARI weakens the $(1 - \frac{1}{e})$ -optimality guarantees, while a complete algorithm or a SAT solver maintains the guarantees with computational penalties.

6. COMPILED MPD COMPUTATION

The second method for computing MPD, compilation, generates a compiled representation $\zeta(\Phi)$ from which all possible diagnoses can be extracted. We introduce a new method of thinking about compilation, one in which, rather than completely compiling a boolean function Φ , we are compiling only those solutions to Φ (representing sets of diagnoses) induced by a preference relation ϕ and threshold ϱ . We show that this preference-based compilation function, f_{PSE} , is submodular, and hence a greedy approximation algorithm for f_{PSE} will approximate the complete set of diagnoses to within $(1 - \frac{1}{e})$ of optimal, and will be sound. We define a compilation function that captures the compilation approaches defined using PIs [de Kleer [1986]], DNNF [Darwiche [2001]], and OBDDs [Bryant [1992]], as follows.

Definition 6. (Compilation). Given a problem P, an instance $\langle \Phi, \alpha \rangle$ of P and a query function Q_P , we define a compilation function of Φ , $\zeta(\Phi)$, such that there exists a query function $Q'_P : \zeta(\mathcal{B}) \times \Psi^* \to \{\text{yes, no}\}$ and $\forall \langle \Phi, \alpha \rangle \in \mathcal{B} \times \Psi^*$ it holds that $Q'_P(\zeta(\Phi), \sigma) = Q_P(\Phi, \alpha)$.³

It is clear from Definition 6 that a compilation $\zeta(\Phi)$ preserves the models of Φ . We now describe several properties of a compilation given preference function ϕ . It is important that the compilation preserves the preference order over the models of Φ .

Definition 7. (Preference Preservation). A compilation $\zeta(\Phi)$ preserves a preference function ϕ if, given $\langle \Phi, \sigma \rangle$, for any pair of diagnoses δ_1, δ_2 such that $\delta_1, \delta_2 \in \Delta_{\Phi}$ and $\delta_1, \delta_2 \in \Delta_{\zeta(\Phi)}, \delta_1 \succ \delta_2$ is valid in $\zeta(\Phi)$ iff $\delta_1 \succ \delta_2$ is valid in Φ .

Since we are interested in compiling only the mostpreferred diagnoses, we define *preferred approximate compilations* as compilations including only a subset of mostpreferred diagnoses:

Definition 8. (Preferred Approximate Compilation). Given a compilation $\zeta(\Phi)$ with space of diagnoses $\Delta_{\zeta(\Phi)}$, $\zeta(\Phi)$ denotes a preferred approximate compilation of Φ if: (1) $\Delta_{\zeta(\Phi)} \subseteq \Delta_{\Phi}$; and (2) $\forall \delta_1 \in \Delta_{\zeta(\Phi)}, \forall \delta_2 \in \Delta_{\Phi} \setminus \Delta_{\zeta(\Phi)}$, it holds that $\delta_1 \succ \delta_2$.

We aim to compile all diagnoses with valuations at least as preferred as ρ , which we denote as a ρ -sound compilation. Definition 9. (ρ -sound compilation). Given a preference function ϕ over Φ and a threshold ρ , the preferred approximate compilation $\zeta(\Phi)$ denotes a ρ -sound compilation of Φ if: (1) $\zeta(\Phi)$ preserves the preference function ϕ ; and (2) $\zeta(\Phi)$ contains every diagnosis $\delta \in \Delta_{\rho}$, i.e. $\Delta_{\zeta(\Phi)} \supseteq \Delta_{\rho}$.

We have designed our algorithms [Venturini and Provan [2008]] to incrementally generate partial diagnoses of increasing size only if the partial diagnosis at each step has

³ Note that this definition differs from the original definition of [Cadoli et al. [2002]], where a compilation is assumed to be of size polynomial in $|\Phi|$ and answering the query Q'_P is assumed to require a time polynomial in $|\alpha| + |\zeta(\Phi)|$.

preference weight greater than ρ . We call this approach partial-solution extension (PSE), which we can prove holds for the set ϕ^* of preference functions for \subseteq , \leq and probability.

Theorem 2. Given a preference function $\phi \in \phi^*$ over Φ and a threshold ϱ , partial-solution extension (PSE) is guaranteed to generate a ϱ -sound compilation of Φ .

Proof: First, it is trivial to show that, given two partial solutions δ_1 and δ_2 which agree on truth-assignments λ_1 and λ_2 (except that δ_2 makes one more assignment than δ_1), $\lambda_1 \succ \lambda_2$.⁴ Using this, we can now show that we never exclude a solution $\delta \in \Delta_{\varrho}$. Assume that partial solution λ_1 is less preferred than ϱ ; by pruning λ_1 , there can be no more-preferred extension λ' to λ_1 , since for any extension λ'' of $\lambda_1, \lambda_1 \succ \lambda''$. Hence, we never exclude any solutions which are extensions of λ_1 . Using an inductive argument, for any partial solution we never omit any solution $\delta \in \Delta_{\varrho}$ through PSE. \Box

We now show that the PSE function, f_{PSE} , is submodular for the probabilistic preference order; analogous arguments can be made for the subset and set-cardinality orders. Our PSE function operates over partial-diagnosis set functions, i.e., subsets of health variables. Hence, for the probabilistic preference order, $f_{PSE} : 2^{\mathcal{H}} \rightarrow [0, 1]$. f_{PSE} extends a partial diagnosis δ to create a more complete health assignment H, and uses an oracle to check if H is consistent and has $Pr(H) > \varrho$. As before, we assume that components fail independently, so the probability of a health assignment His $Pr(H) = \prod_{h_i \in H} Pr(h_i)$.

Lemma 5. The function $f_{PSE}(\mathcal{H})$ induced by a maxprobability ordering is submodular.

Proof. Assume that we have health assignments A and B, with $A \subset B$, $Pr(A) = \pi_A$, and $Pr(B) = \pi_B$. By the definition of submodularity, we must show that $f(A \cup \{x\}) - f(A) \ge f(B \cup \{x\}) - f(B)$. Taking the lefthand-side, we have $f_p(A \cup \{x\}) - f_p(A) = \pi_A * \pi_x - \pi_A = \pi_A(\pi_x - 1)$. Taking the right-hand-side, we have $f_p(B \cup \{x\}) - f_p(B) = \pi_B * \pi_x - \pi_B = \pi_B(\pi_x - 1)$. Since $A \subset B$ and $1 \ge Pr(\cdot) \ge 0$, $\pi_A \le \pi_B$. In addition, since $1 \ge Pr(\pi_x) \ge 0$, $\pi_x - 1 \le 0$. Hence we must have $\pi_A(\pi_x - 1) \ge \pi_B(\pi_x - 1)$, and we obtain the required equation 1.

This submodularity means that a greedy algorithm based on PSE is guaranteed to compile an approximate representation for which the cumulative preference weight is within a constant factor of that of the complete compilation, using an argument analogous to that of Section 4.

We extend this greedy approach to allow a user to specify a preference threshold ρ to control the space of diagnoses in the approximate compilation. We now briefly describe the PI and DNNF ρ -compilation approaches [Venturini and Provan [2008]].

Prime Implicants The compilation algorithm for PIs was implemented using the PRIME algorithm [Shiny and Pujari [2002]]. The original procedure generates full compilations in a *divide-et-impera* fashion, by (1) dividing a formula Φ into two sub-formulae, (2) recursively calculating

the PIs of each sub-formula, and (3) merging the results in order to obtain the PIs of Φ that are more preferred than the threshold ρ . In other words, we discard partial diagnoses that are less preferred than ρ , in accordance with the PSE approach.

DNNF The compilation algorithm for DNNF is based on the algorithm proposed in [Darwiche [2001]]. The original procedure calculates full DNNF compilations of a formula Φ by executing the following steps: (1) a specific subset β of variables of Φ is created; (2) for each instantiation of β -variables, Φ is divided into two sub-formulae; (3) the DNNF compilation of each sub-formula is calculated recursively; (4) the DNNF compilation of Φ is calculated by merging the DNNF representations of the sub-formulae. To calculate approximate compilations, we modified the second step of the original algorithm so that instantiations less preferred than the threshold ρ are not considered. Thus, we compile only those DNNF representations that encode partial diagnoses more preferred than ρ .

7. EMPIRICAL RESULTS

This section summarises results which empirically validate this approach for two important compilation targets, prime implicants and DNNF. We have developed PSE algorithms for PI and DNNF compilations [Venturini and Provan [2008]]. We provide empirical evidence that incremental ρ -compilation achieves space reductions of several orders-of-magnitude over the full compilation, while losing relatively little query completeness. The experiments were run with formulae representing a suite of digital circuits which were generated randomly by a circuit generator program [Provan and Wang [2007]], such that the circuits have properties similar to those of the ISCAS circuit benchmarks [Brglez et al. [1989]]. Each circuit is represented by a formula Φ defined over a set of boolean variables $V = H \cup K$, where a variable $h_i \in H$ denotes the health of gate i, and variables in K denote input/output signals. To each variable $h_i \in H, i = 1, ..., m$ we assign a probability valuation by randomly sampling the value for $Pr(h_i = 1) = p$ from a uniform distribution over ranges R = [i, 1], with $0 \le i < 1$. We studied a probability range R = [0.99, 1]; however, our experimental data indicate that we obtain similar results using different ranges.

For our approach to work, two key properties are required to achieve computational feasibility: (1) the subspace of preferred health assignments H_{ϱ} over which we search must be much smaller that 2^{H} , i.e., $H_{\varrho} \ll 2^{H}$.; and (2) we must have an efficient oracle for the iterative algorithm. The determination of whether $H_{\varrho} \ll 2^{H}$ is dependent on the preference ordering. We have empirically examined the effect of the probabilistic ordering on H_{ϱ} and on the number of PIs. For both cases, we compared the full set of health assignments/PIs with subsets for orderings with tighter ranges on allowable probabilities. Figure 1 shows that the number of PIs decreases as the probability range gets tighter. The number of preferred PIs decreases as $i \to 1$ for all threshold values $\varrho > 0$. Since real-world diagnostic problems typically have very small ranges for fault likelihoods, this indicates that we will have $H_{\varrho} \ll 2^{H}$ and $\Delta_{\rho} \ll \Delta$.

⁴ We call λ_2 an extension of λ_1 .



Fig. 1. Number of prime implicants of a formula Φ encoding a 16-gate circuit; each graph refers to a different probability range.



Fig. 2. Plots of $f(\Delta_{\varrho})$, or CDF, of diagnosis distributions for PI compilations. Given a full compilation ζ_f and an approximate compilation ζ_a , Memory Ratio is defined as $mr = |\zeta_a|/|\zeta_f|$ and Coverage Ratio is defined as $cr = (\sum_i Pr(\delta_i))/(\sum_j Pr(\delta_j))$, with $\delta_i \in \zeta_a$ and $\delta_j \in \zeta_f$.

Further, note that we can control the number of diagnoses (and thus the size and coverage of the approximate compilation) by selecting appropriate (i, ρ) combinations.

Figure 2 plots $f(\Delta_{\varrho})$, the Cumulative Distribution Functions (CDF), of diagnosis distributions for circuits with 10 to 18 gates, encoded in formulae with 58 to 108 clauses. These graphs refer to PI compilations; we obtain similar results using DNNF. This plot has the typical shape of a submodular function. Here we see that, using approximate compilations, we obtain significant diagnosis coverage yet require just a fraction of the memory of the full compilation. In particular, we obtain up to 5 orders-of-magnitude space savings, while maintaining > 90% query-coverage; moreover, for all circuits very high coverage ratio (> 99%) is possible with 3-4 orders-of-magnitude space savings. The space savings increase with circuit (or formula) size, meaning that this approach scales well with the size of Φ .

8. SUMMARY AND CONCLUSIONS

We have proposed an approach for approximately computing all diagnoses, using either iterative or compiled diagnosis algorithms. This approach avoids the time-/spacecomplexity drawbacks of existing algorithms, at the expense of solution incompleteness. However, by showing that preference-based diagnoses are computed by a submodular function, we show that a simple greedy algorithm can obtain a set of diagnoses within $(1 - \frac{1}{c})$ of optimal.

We summarised some empirical results for the compilation approach for PI and DNNF target languages. All the algorithms are ρ -sound, i.e. they compute approximate compilations that include all diagnoses more preferred than the valuation threshold ρ . These results demonstrate the space efficiency of approximate compilations, showing that we can achieve orders-of-magnitude space savings while covering the majority of diagnoses.

REFERENCES

- B. Bollig and I. Wegener. Improving the variable ordering of OBDDs is NP-complete. *IEEE Transactions on Computers*, 45(9):993–1002, 1996.
- F. Brglez, D. Bryan, and K. Kozminski. Combinational profiles of sequential benchmark circuits. In *IEEE International Symposium on Circuits and Systems*, volume 3, pages 1929–1934, May 1989.
- R.E. Bryant. Symbolic boolean manipulation with ordered binary-decision diagrams. ACM Computing Surveys volume 24, pages 293–318. 1992.
- M. Cadoli, F.M. Donini, P. Liberatore, and M. Schaerf. Preprocessing of Intractable Problems. *Information and Computation*, 176(2):89–120, 2002.
- A. Darwiche. Model-Based Diagnosis using Structured System Descriptions. Journal of Artificial Intelligence Research, 8:165–222, 1998.
- A. Darwiche. Decomposable negation normal form. Journal of the ACM (JACM), 48(4):608–647, 2001.
- Johan de Kleer. An assumption-based TMS. Artif. Intell., 28(2):127–162, 1986.
- U. Feige. A threshold of ln n for approximating set cover. Journal of the ACM (JACM), 45(4):634–652, 1998.
- Alexander Feldman, Gregory Provan, and Arjan van Gemund. Computing minimal diagnoses by greedy stochastic search. In Proc. AAAI, July 2008.
- A. Ferrara, P. Liberatore, and M. Schaerf. Model Checking and Preprocessing. In Proc. AI*IA 2007: Artif. Intelligence and Human-Oriented Computing. Springer, 2007.
- GL Nemhauser, LA Wolsey, and ML Fisher. An analysis of approximations for maximizing submodular set functions (I). *Math. Programming*, 14(1):265–294, 1978.
- G. Provan and J. Wang. Evaluating the adequacy of automated benchmark model generators for model-based diagnostic inference. In *Proceedings of IJCAI-07*, Hyderabad, India, 2007.
- A.K. Shiny and A.K. Pujari. Computation of Prime Implicants using Matrix and Paths. *Journal of Logic* and Computation, 8(2):135–145, 2002.
- Detlef Sieling. The nonapproximability of OBDD minimization. Inf. Comput., 172(2):103–138, 2002.
- A. Venturini and G. Provan. Incremental algorithms for approximate compilation. In *Proc. AAAI*, 2008.