Developing Large-Scale Bayesian Networks by Composition: Fault Diagnosis of Electrical Power Systems in Aircraft and Spacecraft

Ole J. Mengshoel
CMU, Intelligent Systems Division
NASA Ames Research Center
Moffett Field, CA 94035
Phone: (650) 604-4199
Ole.J.Mengshoel@nasa.gov

Scott Poll
Intelligent Systems Division
NASA Ames Research Center
Moffett Field, CA 94035
Phone: (650) 604-2143
Scott.Poll@nasa.gov

Tolga Kurtoglu
MCT, Intelligent Systems Division
NASA Ames Research Center
Moffett Field, CA 94035
Phone: (650) 604-1738
Tolga.Kurtoglu@nasa.gov

Abstract

In this paper, we investigate the use of Bayesian networks to construct large-scale diagnostic systems. In particular, we consider the development of large-scale Bayesian networks by composition. This compositional approach reflects how (often redundant) subsystems are architected to form systems such as electrical power systems. We develop high-level specifications, Bayesian networks, clique trees, and arithmetic circuits representing 24 different electrical power systems. The largest among these 24 Bayesian networks contains over 1,000 random variables. Another BN represents the real-world electrical power system ADAPT, which is representative of electrical power systems deployed in aerospace vehicles. In addition to demonstrating the scalability of the compositional approach, we briefly report on experimental results from the diagnostic competition DXC, where the ProADAPT team, using techniques discussed here, obtained the highest scores in both Tier 1 (among 9 international competitors) and Tier 2 (among 6 international competitors) of the industrial track. While we consider diagnosis of power systems specifically, we believe this work is relevant to other system health management problems, in particular in dependable systems such as aircraft and spacecraft.

1 Introduction

This paper is concerned with efficient probabilistic reasoning and diagnosis in particular. Our approach is based on developing a Bayesian network [Pearl, 1988] model of a system, and then using it to efficiently compute answers to probabilistic queries. Bayesian networks and their inference engines provide a well-established approach to model-based diagnosis and monitoring [Lerner *et al.*, 2000; Chien *et al.*, 2002; Yongli *et al.*, 2006; Mengshoel *et al.*, 2008].

We focus on NASA-relevant research problems that represent challenges in aircraft and spacecraft health management. We take as our point of departure an electrical power system known as the Advanced Diagnostics and Prognostics Testbed (ADAPT). ADAPT is an electrical power system (EPS) developed at NASA Ames for supporting the development of diagnostic and prognostic models; for evaluating advanced warning systems; and for testing diagnostic tools and algorithms [Poll *et al.*, 2007]. ADAPT is representative of electrical power systems deployed in aerospace vehicles.

Progress in probabilistic model-based diagnosis is stimulated by real-world applications, and EPSs raise several challenges including the following: (1) The challenge of developing models that are capable of accurately diagnosing 100s or 1000s of different faults, many of which may occur at the same time; (2) The challenge of real-time diagnostic computing, especially on on-board avionics systems with limited processor and memory capacity [Musliner *et al.*, 1995; Mengshoel, 2007a]; (3) The challenge of developing BNs (and in particular large-scale BNs) for a wide spectrum of system sizes while obtaining high performance.

To start addressing these challenges, we have developed a probabilistic approach to model-based diagnosis for ADAPT [Mengshoel *et al.*, 2008; 2009; Ricks and Mengshoel, 2009]. Our probabilistic models represent the health state of sensors and other system components explicitly by means of random variables. To address challenge (1) of model development, we have developed a systematic approach to representing electrical power systems as Bayesian networks, supported by an easy-to-use specification language. To address the real-time reasoning challenge (2), we compile BNs into arithmetic circuits or clique trees. The evaluation of arithmetic circuits and clique trees addresses challenge (2) by being predictable and fast. In experiments with an ADAPT BN containing 503 discrete nodes and 579 edges, the time taken to exactly compute the most probable explanation using an arithmetic circuits or a clique tree was in the order of 1-10 milliseconds [Mengshoel *et al.*, 2009].

While this paper investigates all three challenges associated with model-based reasoning identified above, we focus on the challenge (3) and present the following analytical and experimental contributions. We introduce an analytical approach, based on clique tree clustering [Lauritzen and Spiegelhalter, 1988], that aids in developing large-scale BNs by composition. This compositional approach reflects how (often redundant) subsystems are architected to form systems such as EPSs. Experimentally, we consider BNs representing 24 different EPS architectures including ADAPT, formed

Yannick Pencolé, Thierry Vidal, and Roberto Micalizio, Editors.
The IJCAI-09 Workshop on Self-* and Autonomous Systems: reasoning and integration challenges (SAS-09), July 13, 2009, Pasadena, California, USA.

by the integration of a varying number of power storage and power distribution subsystems. These 24 BNs are representative of real-world EPSs, and are thus to be contrasted with the synthetic problem instances often used for large-scale experimentation [Mitchell *et al.*, 1992; Ide *et al.*, 2004; Mengshoel *et al.*, 2006]. Previous work at the intersection of EPSs and diagnosis using BNs typically considers individual EPSs and their corresponding BNs [Chien *et al.*, 2002; Yongli *et al.*, 2006; Mengshoel *et al.*, 2008]; we are not aware of other efforts that consider BNs representing 20-30 realistic and distinctly different EPSs as is done in this paper. Also, existing work on BNs for EPSs, has, with a few exceptions [Mengshoel *et al.*, 2008; 2009; Ricks and Mengshoel, 2009], been in the area of terrestrial EPSs [Chien *et al.*, 2002; Yongli *et al.*, 2006] rather than in the area of EPSs for aerospace vehicles. While we consider EPS health management specifically, the work has application to numerous health management problems, including such problems in aircraft and spacecraft.

The remainder of this paper is structured as follows. Concepts related to Bayesian network are presented first, followed by a discussion of EPSs. We then present our scalability analysis and an EPS case study. We report strong experimental results, both diagnostic performance in the diagnostic challenge competition DXC and scalability performance for 24 different EPSs including ADAPT. Finally, we conclude and outline future research.

2 Preliminaries

The diagnosis task can be approached from different perspectives [Pearl, 1988; Cordier *et al.*, 2004]. We take in this paper a probabilistic perspective, and investigate Bayesian networks. A Bayesian network (BN) structures a multi-variate probability distribution by using a directed acyclic graph (DAG). Our emphasis will be on DAGs in which nodes represent discrete random variables. Specifically, a (discrete) BN node V is a discrete random variable with a mutually exclusive, exhaustive, and finite state space $\Omega_V = \Omega(V) = \{v_1, ..., v_m\}$. We use the notation Π_V for the parents of a node V, Ψ_V for the children of V, and π_V for an instantiation of all parents Π_V of V. The notion of a Bayesian network can now be introduced [Pearl, 1988].

Definition 1 (Bayesian network) *A Bayesian network is a tuple* $(\boldsymbol{V}, \boldsymbol{W}, \boldsymbol{P})$*, where* $(\boldsymbol{V}, \boldsymbol{W})$ *is a DAG with nodes* $\boldsymbol{V} = \{V_1, ...V_n\}$*, directed edges* $\boldsymbol{W} = \{W_1, ..., W_m\}$*, and where* $\boldsymbol{P} = \{\Pr(V_1 \mid \Pi_{V_1}), \ldots, \Pr(V_n \mid \Pi_{V_n})\}$ *is a set of conditional probability tables (CPTs). For each node* $V_i \in \boldsymbol{V}$ *there is one CPT, which defines a conditional probability distribution* $\Pr(V_i \mid \Pi_{V_i})$.

The independence assumptions induced by $(\boldsymbol{V}, \boldsymbol{W})$ in Definition 1 imply the following joint distribution:

$$\Pr(\boldsymbol{v}) = \Pr(V_1 = v_1, \ldots, V_n = v_n) = \prod_{i=1}^{n} \Pr(v_i \mid \pi_{V_i}), \tag{1}$$

where $\Pi_{V_i} \subset \{V_{i+1}, \ldots, V_n\} \subseteq \boldsymbol{V}$, assuming a reverse topological sort of $\boldsymbol{V}$. (This is possible since $(\boldsymbol{V}, \boldsymbol{W})$ is a DAG.)

A BN can be provided *evidence* by setting or clamping evidence variables $\boldsymbol{E} \subset \boldsymbol{V}$ to known states $\boldsymbol{e}$. Taking into account the input on evidence variables, different probabilistic queries can be answered [Pearl, 1988]. These probabilistic queries include marginals, most probable explanation (MPE), and maximum aposteriori probability (MAP). Probabilistic queries can be used for diagnosis, in which case health variables $\boldsymbol{H} \subseteq \boldsymbol{V} - \boldsymbol{E}$ — representing the health of components, sensors, or both [Mengshoel *et al.*, 2008] — are queried.

Two broad classes Bayesian network inference approaches exist: Interpretation and compilation. In interpretation approaches, a Bayesian network is directly used for inference. In compilation approaches, such as the clique tree [Lauritzen and Spiegelhalter, 1988; Shenoy, 1989] and arithmetic circuit [Darwiche, 2003; Chavira and Darwiche, 2007] approaches investigated here, a Bayesian network is off-line compiled into a secondary data structure, and this secondary data structure is then used for on-line inference. In clique tree clustering, on-line inference consists of propagation in a clique tree. In arithmetic circuit evaluation, on-line inference is performed in an arithmetic circuit. In both cases, on-line and off-line computation time depends on a number of structural and numerical factors associated with a BN and is not yet, despite recent progress [Mengshoel *et al.*, 2006; Mengshoel, 2007b], sufficiently understood.

3 Electrical Power Systems and ADAPT

Electrical power systems (EPSs) play a crucial role in aircraft and spacecraft [Button and Chicatelli, 2005; Poll *et al.*, 2007]. The ADAPT EPS testbed has been developed to support the investigation of system health management technologies in a real-world setting. In this paper, we investigate ADAPT's power storage and distribution subsystems. Over a hundred sensors report their measurements to a diagnostic system that monitors the status of the EPS. Typical sensor measurements of system variables include voltages, currents, temperatures, and relay positions. The ADAPT testbed provides a controlled environment to inject failures in a repeatable manner, and this makes it ideal for use in experiments with novel diagnostic techniques and models.

The physical hardware of the ADAPT EPS consists of battery chargers, batteries, relays, circuit breakers, inverters, wires, sensors, and loads. Most of the hardware is contained within equipment racks or cabinets, with the exception of the loads which are placed in the surrounding lab area. Three batteries may be interchangeably connected to two load banks. Each load bank can connect up to 6 alternating current (ac) loads and 2 direct current (dc) loads. The locations of the loads with respect to the load bank connection points are fixed for the purposes of any given experiment. Different configurations or modes of the EPS are commanded by opening and closing different combinations of relays between the batteries and the loads. As a consequence, ADAPT's system behavior is hybrid, consisting of discrete mode changes and continuous behavior within the modes.

Yannick Pencolé, Thierry Vidal, and Roberto Micalizio, Editors.
The IJCAI-09 Workshop on Self-* and Autonomous Systems: reasoning and integration challenges (SAS-09), July 13, 2009, Pasadena, California, USA.

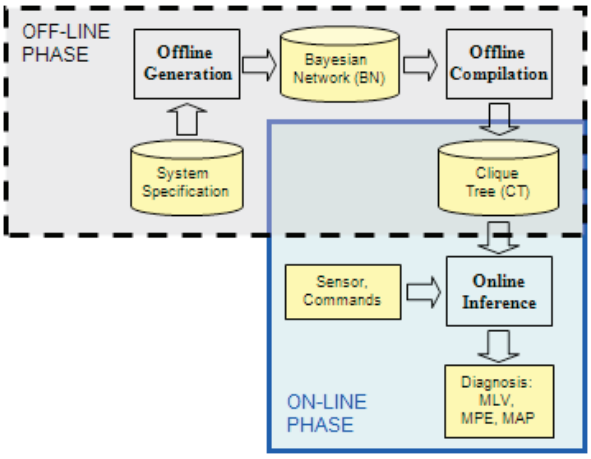

Figure 1: The off-line and on-line phases of our approach. Off-line, Bayesian networks are auto-generated from system specifications, and clique trees or arithmetic circuits are compiled from Bayesian networks. On-line, clique trees or arithmetic circuits are used for diagnosis.

4 Architecture Overview

The architecture of our approach, which is also discussed elsewhere [Mengshoel *et al.*, 2008; 2009; Ricks and Mengshoel, 2009], is given in Figure 1. A system specification, which is created by a user according to a simple high-level specification language, is input to an off-line generation process, which auto-generates a BN. This BN is then compiled into a clique tree [Lauritzen and Spiegelhalter, 1988; Shenoy, 1989] or an arithmetic circuit [Darwiche, 2003; Chavira and Darwiche, 2007]. A high-level specification is, in our case, a sequence of statements, and our language's syntax is presented in Table 1. Generally speaking, an EPS specification captures, in an easy-to-read manner, the flow of power from the sources (batteries) to the sinks (loads) as determined by the structure of the EPS. Each line in the specification represents a part, which currently can be either be a *source* (battery), a *basic* part, a *sensor*, or a *sink* (load) — see [Mengshoel *et al.*, 2008; 2009]. In Table 1, `<name>` is an identifier and `<p>` is a probability. In a specification, a part's name, type (e.g., `source`, `load`, `breaker`, `relay`, `sensorCurrent`, `sensorVoltage`), the probability of failure, and a set of upstream parts (closer to some battery) are all defined. An example specification is provided in Section 6. In aerospace, as well as in other industries with dependability requirements, failure probabilities are obtained as part of often mandatory processes known as Failure Mode and Effects Analysis (FMEA) or Failure Mode, Effects, and Criticality Analysis (FMECA). Other sources of component failure probabilities include standards such as IEEE 493, "Recommended Practice for the Design of Reliable Industrial and Commercial Power Systems," also known as the Gold Book.

This auto-generation architecture and the high-level specification language are similar to but also different from approaches such as Probabilistic Relational Models, Bayesian Logic Programming, Stochastic Logic Programs, and Object-Oriented BNs [Getoor and Taskar, 2007]. It is similar in its goal of making large-scale probabilistic model development [Neil *et al.*, 2000] easier and its emphasis on higher-level structures compared to the propositional nature of BNs. However, there is a difference in that our specification language emphasizes ease-of-use and is more of a domain-specific language, while the alternative languages identified above are more general and expressive.

```
<eps> ::= <component>+
<component> ::= (<source> | <basic> | <sensor> | <sink>) ";"
<source> ::= <name> ":" "source" ":" <p> ":"
<basic> ::= <name> ":" <btype> ":" <p> ":" <name>+
<sensor> ::= <name> ":" <stype> ":" <p> ":" <name>
<sink> ::= <name> ":" "sink" ":" <p> ":" <name>+
<btype> ::= "load" | "wire" | "inverter" | "breaker" | "relay"
<stype> ::= "sensorCurrent" | "sensorVoltage" | "sensorTouch"
```

Table 1: The syntax of the specification language for electrical power systems.

5 Composition and Scalability Analysis

We have developed a multi-variate Bayesian network model of the ADAPT EPS, containing over 500 random variables including over 100 health variables, where the health variables include components and sensors [Mengshoel *et al.*, 2008; 2009]. This BN supports the diagnosis of multiple sensor and/or component faults. We now consider the scalability over a range of BNs representing different EPSs, including the ADAPT BN as described above as one data point.

Scalability, in terms of space requirement and computation time for clique tree evaluation, is determined by clique tree size [Lauritzen and Spiegelhalter, 1988].

Definition 2 (Clique tree size) *Let Γ be the set of cliques in a clique tree compiled from a BN β. The (total) clique tree size is defined as*

$$s_{CT}(\Gamma) = \sum_{\gamma \in \Gamma} \prod_{X \in \gamma} |\Omega_X| \,. \tag{2}$$

In (2), we first multiply the cardinalities of the nodes in a clique γ, and then sum over all the cliques Γ in order to obtain total clique tree size. A number of interacting factors determine the number of cliques and the size of each clique in (2); we now discuss a few of them.

The Subsystem (or Composition) Factor: Suppose that we consider an EPS as a system that might be part of a larger system-of-systems (SoS) such as an aircraft. As we vary the size of the SoS, the size of its systems typically also need to vary. For example, as we vary the aircraft under consideration from a small UAV to a large commercial aircraft, the characteristics of the EPS also change. Since a diagnostic BN needs to vary accordingly, we now consider the impact on clique tree size. We partition a BN's nodes into subsystems $\Upsilon = \{1, ..., \upsilon\}$, and identify subsystem types $\Theta = \{1, ..., \theta\}$, with $\theta \leq \upsilon$. In EPSs, typical subsystem types are: power generation, power storage, and power distribution. ADAPT has, for example, 3 power storage and 2 power distribution subsystems. Hence, $\Upsilon = \{1, 2, 3, 4, 5\}$ and $\Theta = \{1, 2\}$ for ADAPT.

Yannick Pencolé, Thierry Vidal, and Roberto Micalizio, Editors.
The IJCAI-09 Workshop on Self-* and Autonomous Systems: reasoning and integration challenges (SAS-09), July 13, 2009, Pasadena, California, USA.

We now introduce a map f from nodes into subsystems: $f : \mathbf{V} \rightarrow \Upsilon$, and also a map g from subsystems into subsystem types: $g : \Upsilon \rightarrow \Theta$. Now, we can define different subsets of cliques from Γ, specifically $\Gamma_i = \{\gamma \in \Gamma \mid \text{for all } X \in \gamma, f(X) = i\}$, and obtain the following:

$$s_{\text{CT}}(\Gamma_i) = \sum_{\gamma \in \Gamma_i} \prod_{X \in \gamma} |\Omega_X| \,. \tag{3}$$

In words, (3) provides the sizes of all cliques in a subsystem.

We define a set of interaction cliques Γ_0 as $\Gamma_0 = \Gamma - \cup_{i=1}^{v} \Gamma_i$. The set Γ_0 represents the interaction between different subsystems. We obtain the following alternative expression for total clique tree size:

$$s_{\text{CT}}(\Gamma) = \sum_{i=0}^{v} s_{\text{CT}}(\Gamma_i). \tag{4}$$

Now, instead of considering the subsystems individually as in (4), we make the assumption that each of them is identical (given its type). Formally, we let $i \in \Upsilon$ and assume $s_{\text{CT}}(\Gamma_i) = s_{\text{CT}}(\Gamma_{g(i)})$ as well as $c_0 = 1$ and obtain the following result:

$$s_{\text{CT}}(\Gamma) = \sum_{i=0}^{\theta} c_i \times s_{\text{CT}}(\Gamma_i), \tag{5}$$

where c_i represents the number of times a subsystem of type $i \in \Theta$ is found in a system. The significance of (5) is that it enables us to analyze the impact (on clique tree size) of different systems, with different size and redundancy requirements, by taking a compositional approach. Specifically, if we know or can reliably estimate $s_{\text{CT}}(\Gamma_i)$, we just need to count the number of times c_i a subsystem type i occurs, and then do this for all subsystem types in a given system. This aligns well with design methodologies that use redundancy and product-line approaches to support the development of EPSs for vehicles with different power requirements.

An important but non-trivial question to consider is the value of $s_{\text{CT}}(\Gamma_0)$ in (5) as subsystems are composed in different ways to form a system. Based on (5), we can identify a few special cases and simplifications; further information is provided by our experiments. One simplification, which we call perfect compositionality, puts $c_0 = 0$ in (5) to ignore interactions and adds together the size of each subsystem. Clearly, this creates a lower bound that scales linearly with the number of subsystems c_i for a given Γ_i.

The State Space (or Discretization) Factor: In EPSs, continuous signals are often converted to discrete digital numbers by means of analog-to-digital (A/D) converters. A key parameter in A/D conversion is the number of bits in discretized signal, and how to map these discretized into BN node states. Fundamentally, there is a desire to maximize the fidelity of the BN to the underlying EPS, but at the same time the computation time cannot get too large, because then a diagnosis will not be computed in time. The cardinality of a node has a multiplicative effect in all the cliques in which it is an element, see (2), and hence one needs to carefully trade off the potential improvement in diagnostic accuracy (due to increased discretization) with the cost of increased computation time. Further, this factor may need to be taken into account multiple times if $c_i > 1$ in (5).

Part Name	Type of Part	Failure Probability	Upstream Part
Battery1	battery	0.0005	
Wire1	wire	0.0000	Battery1
Voltage1	sensorVoltage	0.0005	Wire1
Current1	sensorCurrent	0.0005	Wire1
Breaker1	breaker	0.0005	Wire1
Status1	sensorTouch	0.0005	Breaker1
Wire2	wire	0.0000	Breaker1
Relay1	relay	0.0005	Wire2
Feedback1	sensorTouch	0.0005	Relay1
Load1	load	0.0005	Relay1
Temp1	sensorCurrent	0.0005	Load1

Table 2: High-level specification of a small electrical power system (EPS). The EPS consists of two subsystems, namely a battery subsystem (lines from Battery1 to Status1) and a load bank subsystem (lines from Wire2 to Temp1).

The Interaction (or Ambiguity) Factor: Increased interaction or ambiguity in a BN has a detrimental effect on scalability. Consider bipartite BNs as an example [Mengshoel *et al.*, 2006; Mengshoel, 2007b]. An example of low ambiguity is when each leaf node has one parent node. An example of high ambiguity is when each leaf node has five parent nodes. Everything else being equal, the higher the ambiguity, the faster cycles are induced in the moral graph, as a function of the ratio of leaf nodes to root nodes, thereby more quickly inducing cliques with many BN nodes in the clique tree. This factor is perhaps less of a concern in engineered systems including EPSs, since they are typically less ambiguous and often close to tree structured (see experimental results below). However, there may be some ambiguity in the interaction between subsystems, thus impacting the term $s_{\text{CT}}(\Gamma_0)$ in (5).

6 Electrical Power System Case Study

The high-level specification for a small EPS is shown in Table 2. We hypothesize that it is much easier for users, including people well-versed in probabilistic models, to provide information in the format illustrated in Table 2 compared to what is illustrated in Figure 2. On the other hand, the high-level specification language is restricted to represent a certain class of BNs and not BNs in general.

Each line in a high-level specification represents one part of an EPS, and also contains information about its type, failure probability, and location within the overall system. For example, the line *Breaker1 breaker 0.0005 Wire1* in Table 2 communicates that *Breaker1* is a circuit *breaker*; has failure probability *0.0005*; and is downstream of *Wire1*. Broadly speaking, this specification is for an EPS with a single battery, *Battery1*, powering a single load *Load1*, and containing a few sensors and components. Specifically, *Battery1* has a wire *Wire1* downstream of it. *Wire1* has three parts connected to it, namely a voltage sensor *Voltage1*, a current sensor *Current1*, and a circuit breaker *Breaker1*. *Breaker1* has a feedback sensor *Status1* attached to it. *Status1* reports whether the breaker

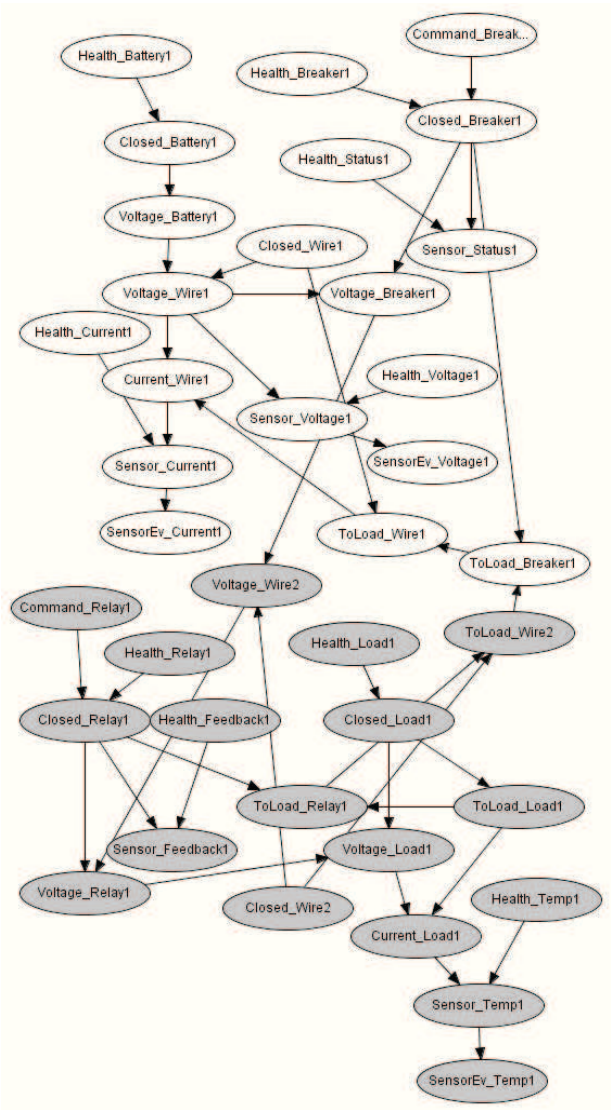

Figure 2: The BN auto-generated from a high-level specification (see Table 2) of a small electrical power system. The BN represents two subsystems, namely a battery subsystem (white nodes) and a load bank subsystem (dark grey nodes). Formally, we have $\Upsilon = \{1, 2\}$ and $\Theta = \{1, 2\}$, with the map f as indicated by the coloring and the map g simply $g(i) = i$ for $i \in \{1, 2\}$. Roughly speaking, the BN reflects both the "push" of power from the battery to the load as well as the "pull" of current by the load. For example, *Voltage1_Battery1* is — subject to *Health_Battery1* (whether *Battery1* is operational or not) and *Closed_Wire1* (whether *Wire1* is open or closed) — pushed downstream to *Voltage1_Wire1*, and so forth.

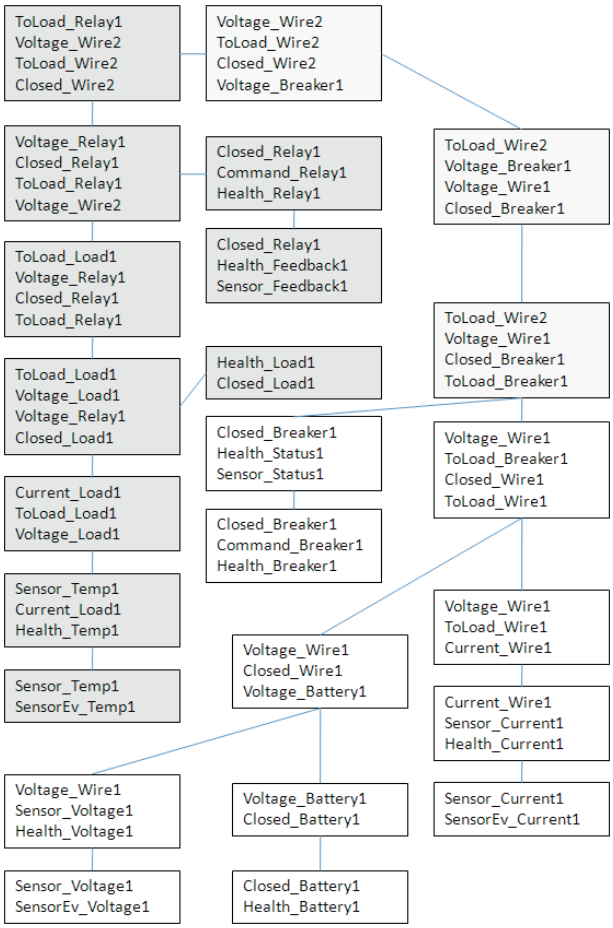

Figure 3: The clique tree compiled from a BN (see Figure 2) representing a small electrical power systems. These cliques can be partitioned into 11 nodes that represent the battery subsystem (white nodes), 10 nodes that represent the load bank subsystem (dark grey nodes), and 3 nodes that represent both subsystems (light grey nodes).

is open or closed. *Wire2*, which is the first part that we consider to be part of the load bank subsystem, is downstream of *Breaker1* and has feedback sensor *Feedback1* as well as *Relay1* attached to it. *Relay1* controls power flow into *Load1*, which has a sensor *Temp1* attached to it.

Nodes in the auto-generated BN can be partitioned into Γ_0, Γ_1, and Γ_2, as indicated in Figure 2. Figure 3 shows a clique tree resulting from the compilation of this BN. Cliques in Γ_1 represent the battery subsystem, those in Γ_2 the load bank subsystem, while cliques in Γ_0 represent the interaction between the two subsystems. Clique tree size is $s_{\text{CT}}(\Gamma) = 264$, with $s_{\text{CT}}(\Gamma_0) = 48$, $s_{\text{CT}}(\Gamma_1) = 98$, and $s_{\text{CT}}(\Gamma_2) = 118$.

7 Experiments

To complement our analysis earlier in this article as well as related experimental results for ADAPT [Mengshoel *et al.*, 2008; 2009; Ricks and Mengshoel, 2009], we now report on

Yannick Pencolé, Thierry Vidal, and Roberto Micalizio, Editors.
The IJCAI-09 Workshop on Self-* and Autonomous Systems: reasoning and integration challenges (SAS-09), July 13, 2009, Pasadena, California, USA.

diagnosis and scalability experiments.

7.1 Diagnosis Experiments

The diagnosis experiments we summarize here were conducted as part of the diagnostic challenge competition DXC, hosted by the 20th International Workshop on the Principles of Diagnosis (see `http://www.dx-competition.org/` for details). The ADAPT EPS was used to generate fault and nominal scenarios for the industrial track of DXC. Fault scenarios contained single or multiple abrupt faults injected simultaneously or sequentially. The fault types were additive parametric (abrupt changes in parameter values) and discrete (unexpected changes in system state). The faults were permanent; once injected they persisted until the end of the scenario. Faults were inserted with equal probabilities, and included both component and sensor faults.

The industrial track consisted of two tiers, Tier 1 and Tier 2. The Tier 1 experiments were easier than the Tier 2 experiments, for several reasons. First, only a subset of ADAPT was used, namely one battery and one load — a fan — on one load bank. Second, all relevant relays were kept in their closed positions for Tier 1, thus minimizing the number of modes and the effect of transients (which may cause false positives). BNs, here denoted DXCT1 and DXCT2 were developed for Tier 1 and Tier 2 respectively, and compiled to ACs that were used for on-line diagnosis in the ProADAPT system [Ricks and Mengshoel, 2009].

In Table 3, we highlight the DXC results for the top three competitors in each tier. As reflected in the table, eight metrics were used. The metrics capture both detection (finding out that some part failed) and isolation (finding out which part failed, and how) performance. Within each tier, and for each metric, each diagnostic system was measured, scored, and ranked relative to the other systems. The maximum score was 100. Diagnostic systems were ranked from 1 to m, where $m = 9$ for Tier 1 and $m = 6$ for Tier 2.

In Table 3, we note that ProADAPT, using ACs compiled from the DXCT1 and DXCT1 BNs, has the best score and rank in both tiers. For the 62 Tier 1 scenarios, which were either nominal or contained one fault, ProADAPT's FP and FN rates are very low, and detection accuracy is high. For the 120 Tier 2 scenarios, which were nominal or contained single, double, or triple faults, ProADAPT again had the highest score. Compared to its competitors, ProADAPT has a low false positives rate and few classification errors; as a consequence the score for mean time to detect suffers somewhat.

7.2 Scalability Experiments

The goal of the second set of experiments was to study BNs representing different EPSs with varying number of subsystems of different types. Different EPS models were created using the high-level specification language. One goal was to study the sizes of the generated BNs, clique trees, and arithmetic circuits. Clique tree and arithmetic circuit sizes determine computation time, which is one important design parameter when developing diagnostic systems for EPSs. We developed 24 different EPS architectures using the high-level specification language, giving 24 auto-generated BNs, which were compiled into clique trees and arithmetic circuits. In Table 4, the notation EPS(x,y) is used to represent an EPS with x battery subsystems and y load bank subsystems (see [Poll *et al.*, 2007] for details on these subsystems).

We now turn to the experiments results for the 24 EPS models including ADAPT.[1] Table 4 and Figure 4 summarize the experimental results; key observations are:

- In Table 4, $\min(m/n) = 1.13$, while $\max(m/n) = 1.17$. This shows that our auto-generated BNs are fortunately quite sparse, given that $n = m + 1$ for trees.
- There is an approximately 5-time increase in BN size from EPS(1,1) to EPS(6,4), an 8.5-time increase in arithmetic circuit size, and a little over 12-time increase in clique tree size. We believe that these are quite promising scalability results, given the inherent hardness of BN computation. Further, if we consider EPS(5,4) instead of the outlier EPS(6,4), we have 4.4 times as many BN nodes compared to EPS(1,1) and only an 8-time increase in clique tree size.
- The clique tree regression results in Figure 4 exhibit better fit for the exponential model ($y = 1112.7e^{0.0027x}$ with $R^2 = 0.9266$) than for the linear model ($y = 18.948x$ - 4185.3 with $R^2 = 0.7647$), pointing to the importance of the potentially nonlinear term $s_{\text{CT}}(\Gamma_0)$ in (5). However, and in particular if the outlier EPS(6,4) is excluded, both models are quite reasonable. The arithmetic circuit regression results are similar, with an exponential model $y = 1320.8e^{0.0023x}$ (with $R^2 = 0.9535$) and a linear model $y = 12.819x$ - 1743 (with $R^2 = 0.8634$).
- The ratio $s_{\text{AC}}/s_{\text{CT}}$, shown in Table 4, generally reflects a smaller growth of the arithmetic circuits relative to the cliques trees as a function of n, thus scalability is generally better for arithmetic circuits here.

8 Conclusion and Future Work

Due to their high level of predictability and fast execution times, Bayesian network compilation approaches are well-suited to automated diagnosis in the setting of on-board resource-bounded reasoning and real-time systems of interest to NASA [Mengshoel *et al.*, 2008]. This paper improves the understanding of the scaling behavior of clique trees and arithmetic circuits in the context of composing large-scale BNs. A designer of model-based diagnostic systems, using Bayesian networks, can use our results to determine the impact of varying EPS architectures, consisting of repeated subsystems, on the computation time of diagnostic queries.

This work has been performed in the context of NASA's ADAPT electrical power system testbed. ADAPT is representative of EPSs deployed on aerospace vehicles. In this paper we have investigated how the BN-based approach to probabilistic diagnosis for ADAPT scales to other electrical power systems composed in a similar manner from power storage and power distribution subsystems.

[1] The DXCT2 BN is similar to EPS(3,2), while the DXCT1 BN is similar to EPS(1,1), except that DXCT1 had just one load — a fan.

Yannick Pencolé, Thierry Vidal, and Roberto Micalizio, Editors.
The IJCAI-09 Workshop on Self-* and Autonomous Systems: reasoning and integration challenges (SAS-09), July 13, 2009, Pasadena, California, USA.

Metric	ADAPT DXC Tier 1			ADAPT DXC Tier 2		
	ProADAPT	**RODON**	**HyDE-S**	**ProADAPT**	**Stanford**	**RODON**
False positives (FP) rate	*0.0333*	0.0645	0.2000	*0.0732*	0.3256	0.5417
False negatives (FN) rate	*0.0313*	0.0968	0.0741	*0.1392*	0.0519	0.0972
Detection accuracy	*0.9677*	0.9194	0.8548	*0.8833*	0.8500	0.7250
Classification errors	*2.0*	10.0	26.0	*76.0*	110.5	84.1
Mean time to detect T_d (ms)	*1,392*	218	130	*5981*	3946	3490
Mean time to isolate T_i (ms)	*4,084*	7,205	653	*12,486*	14,103	36,331
Mean CPU time T_c (ms)	*1,601*	11,766	513	*3,416*	963	8,0261
Mean peak memory usage (kb)	*1,680*	26,679	5,795	*6,539*	5,912	29,878
Score	*72.80*	59.85	59.50	*83.20*	81.50	70.50
Rank	*1*	2	3	*1*	2	3

Table 3: The performance of the ProADAPT and other diagnostic systems, for the two different ADAPT configurations Tier 1 and Tier 2 used in DXC. Our ProADAPT system used arithmetic circuits compiled from the DXCT1 and DXCT2 BNs.

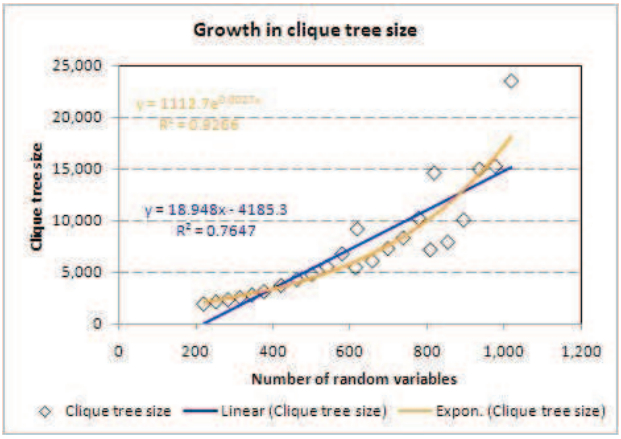

Figure 4: This figure shows how clique tree size s_{CT} varies as a function of the number of BN nodes n. Clique tree size determines computation time, while the number of random variables varies from EPS to EPS. Each data point, of which there are 24, represents an EPS.

This work enables the transition of diagnostic and health management technologies to NASA's mission systems. In particular, it appears that Bayesian networks, techniques, and algorithms for diagnosis can be applied to distinguish between sensor failures and component failures, a problem of great interest to NASA. Future work will aim to help NASA in developing model-based diagnostic and sensor validation approaches that take into account the limited resources available on varying mission hardware. In addition, the compositional approach taken here has the potential to help bridge the gap between hardware (such as the EPS) and software (such as the EPS diagnostics) design. Accordingly, software performance criteria (such as diagnostic computation time) can be incorporated into the design considerations along with hardware design. In the EPS context, future work will aim to help the architectural design of the EPS systems by providing a method to concurrently analyze the impact of varying EPS architectures on the computational performance of diagnostic systems designed to operate on them. This can provide a much needed formal approach for architectural design of safety critical systems — such as an EPS — which often employ redundant system architectures based primarily on expert opinion to mitigate potential effects of sensor and component failures.

References

[Button and Chicatelli, 2005] R. M. Button and A. Chicatelli. Electrical power system health management. In *Proceedings of the 1st International Forum on Integrated System Health Engineering and Management in Aerospace*, Napa, CA, 2005.

[Chavira and Darwiche, 2007] M. Chavira and A. Darwiche. Compiling Bayesian networks using variable elimination. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI-07)*, pages 2443–2449, Hyderabad, India, 2007.

[Chien *et al.*, 2002] C.-F. Chien, S.-L. Chen, and Y.-S. Lin. Using Bayesian network for fault location on distribution feeder. *IEEE Transactions on Power Delivery*, 17:785–793, 2002.

[Cordier *et al.*, 2004] M.-O. Cordier, P. Dague, F. Levy, J. Montmain, M. Staroswiecki, and L. Trave-Massuyes. Conflicts versus analytical redundancy relations: a comparative analysis of the model based diagnosis approach from the artificial intelligence and automatic control perspectives. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 34(5):2163–2177, 2004.

[Darwiche, 2003] A. Darwiche. A differential approach to inference in Bayesian networks. *Journal of the ACM*, 50(3):280–305, 2003.

[Getoor and Taskar, 2007] L. Getoor and B. Taskar, editors. *Introduction to Statistical Relational Learning*. The MIT Press, August 2007.

[Ide *et al.*, 2004] J. S. Ide, F. G. Cozman, and F. T. Ramos. Generating random Bayesian networks with constraints on induced width. In *Proceedings of the 16th European Conference on Artificial Intelligence*, pages 323–327, 2004.

[Lauritzen and Spiegelhalter, 1988] S. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems (with discussion). *Journal of the Royal Statistical Society series B*, 50(2):157–224, 1988.

[Lerner *et al.*, 2000] U. Lerner, R. Parr, D. Koller, and G. Biswas. Bayesian fault detection and diagnosis in dynamic systems. In *Proceedings of the Seventeenth national Conference on Artificial Intelligence (AAAI-00)*, pages 531–537, 2000.

Yannick Pencolé, Thierry Vidal, and Roberto Micalizio, Editors.
The IJCAI-09 Workshop on Self-* and Autonomous Systems: reasoning and integration challenges (SAS-09), July 13, 2009, Pasadena, California, USA.

Name of EPS	**Batte-ries**	**Load Banks**	**Spec. Lines**	n **- BN Nodes**	m **- BN Edges**	**Ratio** m/n	s_{CT} **- CT Size**	s_{AC} **- AC Size**	**Ratio** $s_{\text{AC}}/s_{\text{CT}}$
EPS(6,4)	6	4	292	1,018	1,194	1.17	23,428	15,977	0.68
EPS(5,4)	5	4	280	977	1,143	1.17	15,202	11,359	0.75
EPS(4,4)	4	4	268	936	1,092	1.17	14,912	9,316	0.62
EPS(3,4)	3	4	256	895	1,041	1.16	10,014	8,591	0.86
EPS(2,4)	2	4	244	854	990	1.16	7,868	7,391	0.94
EPS(1,4)	1	4	232	809	935	1.16	7,130	6,867	0.96
EPS(6,3)	6	3	237	819	954	1.16	14,550	9,454	0.65
EPS(5,3)	5	3	225	779	906	1.16	10,164	8,490	0.84
EPS(4,3)	4	3	213	739	858	1.16	8,274	7,440	0.90
EPS(3,3)	3	3	201	699	810	1.16	7,248	6,471	0.89
EPS(2,3)	2	3	189	659	762	1.16	6,046	5,774	0.96
EPS(1,3)	1	3	177	616	711	1.15	5,404	5,248	0.97
EPS(6,2)	6	2	182	620	714	1.15	9,128	6,547	0.72
EPS(5,2)	5	2	170	581	669	1.15	6,726	5,853	0.87
EPS(4,2)	4	2	158	542	624	1.15	5,476	5,140	0.94
EPS(3,2)	*3*	*2*	*146*	*503*	*579*	*1.15*	*4,738*	*4,557*	0.96
EPS(2,2)	2	2	134	464	534	1.15	4,224	4,110	0.97
EPS(1,2)	1	2	122	423	487	1.15	3,678	3,592	0.98
EPS(6,1)	6	1	115	379	426	1.12	3,082	3,077	1.00
EPS(5,1)	5	1	105	348	392	1.13	2,768	2,864	1.03
EPS(4,1)	4	1	95	317	358	1.13	2,518	2,616	1.04
EPS(3,1)	3	1	85	286	324	1.13	2,300	2,390	1.04
EPS(2,1)	2	1	75	255	290	1.14	2,098	2,129	1.01
EPS(1,1)	1	1	65	223	255	1.14	1,896	1,878	0.99

Table 4: The effect of varying the number of two types of EPS subsystems, namely a battery subsystem and a load bank subsystem, is considered. In EPS(x, y), the number of battery subsystems, x, varied from 1 to 6. The number of load bank subsystems, y, is varied from 1 to 4. The table also shows the number of BN nodes n, the number of BN edges m, the ratio of BN edges to BN nodes m/n, the clique tree size s_{CT}, the arithmetic circuit size s_{AC} (measured in number of AC nodes), and the ratio $s_{\text{CT}}/s_{\text{CT}}$. The ADAPT BN corresponds to the highlighted EPS(3, 2) model. The clique tree and arithmetic circuit sizes determine the computation time for a wide range of probabilistic queries.

[Mengshoel *et al.*, 2006] O. J. Mengshoel, D. C. Wilkins, and D. Roth. Controlled generation of hard and easy Bayesian networks: Impact on maximal clique tree in tree clustering. *Artificial Intelligence*, 170(16-17):1137–1174, 2006.

[Mengshoel *et al.*, 2008] O. J. Mengshoel, A. Darwiche, K. Cascio, M. Chavira, S. Poll, and S. Uckun. Diagnosing faults in electrical power systems of spacecraft and aircraft. In *Proceedings of the Twentieth Innovative Applications of Artificial Intelligence Conference (IAAI-08)*, pages 1699–1705, Chicago, IL, 2008.

[Mengshoel *et al.*, 2009] O. J. Mengshoel, M. Chavira, K. Cascio, S. Poll, A. Darwiche, and S. Uckun. Probabilistic model-based diagnosis: An electrical power system case study. *IEEE Trans. on Systems, Man, and Cybernetics*, 2009. Accepted for publication.

[Mengshoel, 2007a] O. J. Mengshoel. Designing resource-bounded reasoners using Bayesian networks: System health monitoring and diagnosis. In *Proceedings of the 18th International Workshop on Principles of Diagnosis (DX-07)*, pages 330–337, Nashville, TN, 2007.

[Mengshoel, 2007b] O. J. Mengshoel. Macroscopic models of clique tree growth for Bayesian networks. In *Proceedings of the Twenty-Second National Conference on Artificial Intelligence (AAAI-07)*, pages 1256–1262, Vancouver, British Columbia, 2007.

[Mitchell *et al.*, 1992] D. Mitchell, B. Selman, and H. J. Levesque. Hard and easy distributions of SAT problems. In *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI-92)*, pages 459–465, San Jose, CA, 1992.

[Musliner *et al.*, 1995] D. Musliner, J. Hendler, A. K. Agrawala, E. Durfee, J. K. Strosnider, and C. J. Paul. The challenges of real-time AI. *IEEE Computer*, 28:58–66, January 1995.

[Neil *et al.*, 2000] M. Neil, N. Fenton, and L. Nielson. Building large-scale bayesian networks. *Knowledge Engineering Review*, 15(3):257–284, 2000.

[Pearl, 1988] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, CA, 1988.

[Poll *et al.*, 2007] S. Poll, A. Patterson-Hine, J. Camisa, D. Garcia, D. Hall, C. Lee, O. J. Mengshoel, C. Neukom, D. Nishikawa, J. Ossenfort, A. Sweet, S. Yentus, I. Roychoudhury, M. Daigle, G. Biswas, and X. Koutsoukos. Advanced diagnostics and prognostics testbed. In *Proceedings of the 18th International Workshop on Principles of Diagnosis (DX-07)*, pages 178–185, Nashville, TN, 2007.

[Ricks and Mengshoel, 2009] B. W. Ricks and O. J. Mengshoel. The diagnostic challenge competition: Probabilistic techniques for fault diagnosis in electrical power systems. In *Proceedings of the 20th International Workshop on Principles of Diagnosis (DX-09)*, Stockholm, Sweden, 2009.

[Shenoy, 1989] P. P. Shenoy. A valuation-based language for expert systems. *International Journal of Approximate Reasoning*, 5(3):383–411, 1989.

[Yongli *et al.*, 2006] Z. Yongli, H. Limin, and L. Jinling. Bayesian network-based approach for power system fault diagnosis. *IEEE Transactions on Power Delivery*, 21:634–639, 2006.

Yannick Pencolé, Thierry Vidal, and Roberto Micalizio, Editors.
The IJCAI-09 Workshop on Self-* and Autonomous Systems: reasoning and integration challenges (SAS-09), July 13, 2009, Pasadena, California, USA.