Diagnosing Faults in Electrical Power Systems of Spacecraft and Aircraft

Ole J. Mengshoel
USRA/RIACS
NASA Ames Research Center
Moffett Field, CA 94035
Phone: (650) 604-4199
Ole.J.Mengshoel@nasa.gov

Adnan Darwiche
Computer Science Department
University of California
Los Angeles, CA 90095
Phone: (310) 206-5201
darwiche@cs.ucla.edu

Keith Cascio
Computer Science Department
University of California
Los Angeles, CA 90095
Phone: (310) 825-7564
keith@cs.ucla.edu

Mark Chavira*
Computer Science Department
University of California
Los Angeles, CA 90095
Phone: (310) 825-7564
chavira@cs.ucla.edu

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

Serdar Uckun
Intelligent Systems Division
NASA Ames Research Center
Moffett Field, CA 94035
Phone: (650) 604-4996
Serdar.Uckun@nasa.gov

Abstract

Electrical power systems play a critical role in spacecraft and aircraft. This paper discusses our development of a diagnostic capability for an electrical power system testbed, ADAPT, using probabilistic techniques. In the context of ADAPT, we present two challenges, regarding modelling and real-time performance, often encountered in real-world diagnostic applications. To meet the modelling challenge, we discuss our novel high-level specification language which supports auto-generation of Bayesian networks. To meet the real-time challenge, we compile Bayesian networks into arithmetic circuits. Arithmetic circuits typically have small footprints and are optimized for the real-time avionics systems found in spacecraft and aircraft. Using our approach, we present how Bayesian networks with over 400 nodes are auto-generated and then compiled into arithmetic circuits. Using real-world data from ADAPT as well as simulated data, we obtain average inference times smaller than one millisecond when computing diagnostic queries using arithmetic circuits that model our real-world electrical power system.

Introduction

Electrical power systems (EPS) are critical for the proper operation of aircraft and spacecraft (Button & Chicatelli 2005; Poll *et al.* 2007). EPS loads in an aerospace vehicle may include crucial subsystems such as avionics, propulsion, life support, and thermal management systems. Apart from their crucial role in spacecraft and aircraft, electrical power systems also play central roles in other parts of society, thus proper management of their health is important.

*Mark Chavira is currently at Google.

Copyright © 2008, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

There are several challenges associated with EPS fault diagnosis. In this paper we discuss two of these challenges, the modelling challenge and the real-time reasoning challenge. While we discuss these challenges in the context of diagnostic reasoning for a real-world EPS, we believe they are of more general interest.

The *modelling challenge* concerns how to model the combination of deterministic and uncertain behavior seen in EPSs. For example, there is uncertainty due to the component and sensor failures and because of sensor noise. Another part of the challenge is to model the EPS in sufficient detail to ensure high diagnostic accuracy. At the same time, the diagnostic model developed for a particular EPS should be easy to construct, extend, and update.

The *real-time reasoning challenge* is associated with the embedding of AI components, including diagnostic reasoners, into hard real-time systems (Musliner *et al.* 1995). For NASA, decision support for manned missions and autonomous action for unmanned missions are both of great interest. The avionics of both manned and unmanned vehicles often utilize a hard real-time operating system (RTOS). An embedded diagnostic engine, which is part of a vehicle's avionics, should therefore be designed within the RTOS framework. For example, an RTOS task needs to declare a worst-case execution time (WCET). Unfortunately, BN inference problems are inherently computationally hard (Cooper 1990; Shimony 1994; Park & Darwiche 2004). In addition, many inference algorithms are associated with high expectations and/or variances in their execution times, and their WCETs are unknown. The real-time reasoning challenge is to develop real-time diagnostic systems, despite the computational hardness of diagnosis problems.

In this paper we present our novel probabilistic approach to EPS fault diagnosis. We discuss the construction of diagnostic BNs for EPSs, using the Advanced Diagnostics and Prognostics Testbed (ADAPT) as a case study (Poll *et al.* 2007). These BNs explicitly represent the health of sensors and components. We emphasize the systematic structuring of these Bayesian networks, based on EPS structure and component types, and discuss the automatic BN generation based on a novel, high-level system specification language. While there is a variety of technology choices for EPS diagnosis, real-time operation and resource constraints on aircraft and spacecraft limit the usefulness of many advanced technologies. As a result, mission- or safety-critical diagnosis applications are often performed using simple lookup tables or production rules. The approach described in this paper combines the expressive power and mathematical rigor of probabilistic methods with the predictability of non-model-based approaches.

We have experimentally evaluated our approach on a number of ADAPT fault scenarios. In order to enable RTOS embedding, the ADAPT BN was compiled (off-line) into an arithmetic circuit, which was then evaluated on-line (Darwiche 2003; Chavira & Darwiche 2007). A unique point compared to previous work (Chien, Chen, & Lin 2002; Yongli, Limin, & Jinling 2006) is how a complex diagnostic search space is reduced to an arithmetic circuit and a small-footprint arithmetic circuit evaluator. Compiling an ADAPT BN, which contains over 400 nodes representing over 100 EPS components, to an arithmetic circuit, and evaluating it using the ACE arithmetic circuit evaluator, turns out to give accurate diagnostic results as well inference times that are less than one millisecond for all our fault scenarios. This is a successful demonstration of our approach on a real-world problem of great importance to NASA (Button & Chicatelli 2005; Poll *et al.* 2007).

The rest of this paper is structured as follows. First, we discuss challenges associated with the diagnosis of electrical power systems. Second, we present our approach to diagnosis of electrical power systems by means of auto-generated Bayesian networks and arithmetic circuits. Finally, we present empirical results for an electrical power system test bed.

Diagnosis of Electrical Power Systems

In this section we discuss the crucial role of electrical power systems in aerospace.

The Role of Electrical Power Systems in Aerospace

EPS loads in an aerospace vehicle include the following crucial subsystems: avionics, propulsion, life support, and thermal management systems. Loss of electrical power to any of these subsystems could lead to serious consequences for personnel or the vehicle.

There are, from the point of view of vehicle health management, several technical challenges associated with electrical power systems. First, electrical power systems often have a large number of distinct modes due to mode-inducing components such as relays, circuit breakers, and loads. If an EPS has m such components, and we conservatively assume 2 discrete states for each, there are potentially 2^m modes in the EPS. Second, while much EPS behavior is deterministic, there is both sensor noise and system state uncertainty in EPSs. Sensor noise is due to the imperfections of sensing, while system state uncertainty is due to failures of EPS components and sensors. These two technical challenges are our main concern in this paper. Our use of Bayesian networks and arithmetic circuits, rather than other approaches to technical diagnosis, is motivated by the need to construct EPS diagnostic models that capture both deterministic and uncertain behavior when many modes are present.

ADAPT: An Electrical Power System Testbed

The Advanced Diagnostic and Prognostic Testbed (ADAPT) is an electrical power system testbed developed at the NASA Ames Research Center. ADAPT provides: (i) a standard testbed for evaluating diagnostic algorithms and software; (ii) a capability for controlled insertion of faults, giving repeatable failure scenarios; and (iii) a mechanism for maturing and transitioning diagnostic technologies onto manned and unmanned vehicles (Poll *et al.* 2007). The EPS functions of ADAPT are as follows (see also `http://ti.arc.nasa.gov/adapt/`). For power generation, ADAPT currently uses utility power. For power storage, ADAPT contains 3 sets of 24 VDC 100 Amp-hr sealed lead acid batteries. Power distribution is aided by electromechanical relays and two load banks with AC and DC outputs; there are also several circuit breakers. ADAPT loads include pumps, fans, and light bulbs. There are sensors of several types, specifically for measuring voltage, current, relay position, temperature, light, and liquid flow. Control and monitoring of ADAPT takes place through programmable automation controllers. With the sensors included, ADAPT contains a few hundred components and is representative of EPSs used in aerospace.

Meeting The Modelling Challenge

Bayesian networks (BNs) are used to represent multivariate probability distributions for the purpose of reasoning and learning under uncertainty (Pearl 1988). In BNs, random variables are represented as nodes in directed acyclic graphs. Each node has a conditional probability table (CPT). BNs can contain both discrete and continuous random variables; the EPS BNs discussed in this paper contain discrete variables only. While a joint probability table's size is exponential in the number of discrete random variables, the BN provides a mechanism to compactly represent the joint probability table.

The main points of our BN-based EPS modelling approach are as follows: (i) We use three different models during development and deployment. (ii) We explicitly represent EPS health using random variables, thus supporting different diagnostic queries of interest. (iii) Finally, we take a component-oriented and causal approach, where the BN structure reflects the components and causal structure of an EPS. We now discuss these three main points in turn.

(i) Figure 1 shows how the diagnostic system developer

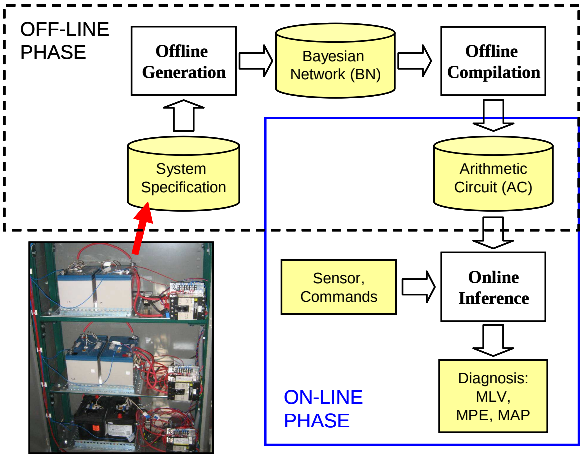

Figure 1: Our approach uses three distinct models that play different roles in the development process: a system specification, a Bayesian network, and an arithmetic circuit.

is supported by a technique and tool pipeline for auto-generation of an arithmetic circuit from a high-level system specification. Bayesian networks serve as intermediate models. Our current ADAPT BN consists of well over 400 nodes, and a more detailed BN or a BN for a larger EPS could easily contain 1000 BN nodes or more. Unfortunately, developing such large BNs by hand, especially in the face of complex BN and EPS topologies, is non-trivial. To meet this modelling challenge, the Offline Generation process depicted in Figure 1 supports the automatic generation of an EPS Bayesian network from a high-level EPS system specification. This architecture clearly illustrates how our work is different from previous work on EPS fault diagnosis using BNs (Chien, Chen, & Lin 2002; Yongli, Limin, & Jinling 2006). As an example, the line `\Relay1 : relay : 0.0005 : Wire2"` in the system specification expresses that we have a relay, `Relay1`, with failure probability `0.0005` connected to `Wire2`. The line `\Feedback1 : sensorTouch : 0.0005 : Relay1"` shows that a feedback sensor, `Feedback1`, is attached to `Relay1`. These two lines result in five BN nodes being auto-generated as shown in the upper right corner of Figure 2. We will explore these five nodes in more detail shortly.

Our specification language is quite general and supports an interesting range of EPSs beyond ADAPT. The algorithm that auto-generates a Bayesian network from a system specification works as follows. Given small BNs representing different components, as presented above for `Relay1` and `Feedback1`, an overall BN is composed according to the EPS topology as it is reflected in the system specification. A key benefit of the specification language is that it is tailored to EPSs and is much more succinct than a Bayesian network (which again is much more succinct than an arithmetic circuit). In addition to making the BN and AC technologies available to a much broader user community, this approach accommodates rapid changes in EPS topology as well as in

```
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 ;
```

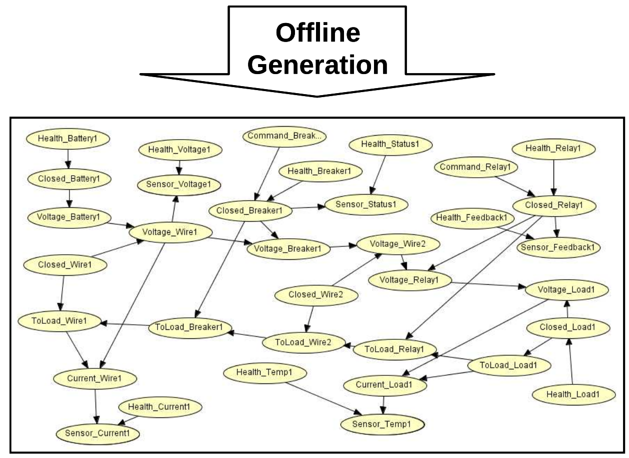

Figure 2: Auto-generation of a Bayesian network representing an electrical power system (bottom) from a high-level system specification (top).

individual EPS components.

(ii) We now discuss some of the different BN node types that we have used to model an EPS. Let $\boldsymbol{X}$ denote all BN nodes. The EPS health nodes are $\boldsymbol{H}_E = \boldsymbol{H}_C \cup \boldsymbol{H}_S$, where $\boldsymbol{H}_E \subseteq \boldsymbol{X}$ and $\boldsymbol{H}_C \cap \boldsymbol{H}_S = \varnothing$. Here, $\boldsymbol{H}_C$ are the component health nodes and represent the health of an EPS excluding its sensors. $\boldsymbol{H}_S$ are the sensor health nodes, and represent the health of the EPS sensors, both their failure and nominal (healthy) modes. By introducing $\boldsymbol{H}_C$ and $\boldsymbol{H}_S$, we represent the health of EPS components and sensors explicitly in the BN. The BN also contains other types of nodes. Specifically, we have input or evidence nodes $\boldsymbol{E}$, with $\boldsymbol{E} = \boldsymbol{E}_C \cup \boldsymbol{E}_S$, where $\boldsymbol{E} \subseteq \boldsymbol{X}$ and $\boldsymbol{E}_C \cap \boldsymbol{E}_S = \varnothing$. Here, $\boldsymbol{E}_C$ are command nodes representing commands from a user to the EPS. $\boldsymbol{E}_S$ are the sensor nodes, which are used to input sensor readings — for example voltage, current, and temperature — from the EPS. We also have status nodes $\boldsymbol{S}$, with $\boldsymbol{S} \subseteq \boldsymbol{X}$, which are nodes that reflect the EPS structure but do not fit into any of the categories above. Finally, we have $\boldsymbol{X} = \boldsymbol{H}_E \cup \boldsymbol{E} \cup \boldsymbol{S}$, with $\boldsymbol{H}_E \cap \boldsymbol{E} = \varnothing$, $\boldsymbol{H}_E \cap \boldsymbol{S} = \varnothing$, and $\boldsymbol{E} \cap \boldsymbol{S} = \varnothing$. The ADAPT BN currently contains over 400 nodes, and models most of ADAPT from the batteries downstream. Since it is impossible to present this BN in its totality here, Figure 4 shows a representative BN's conditional probability tables (CPTs) along with a corresponding arithmetic circuit.

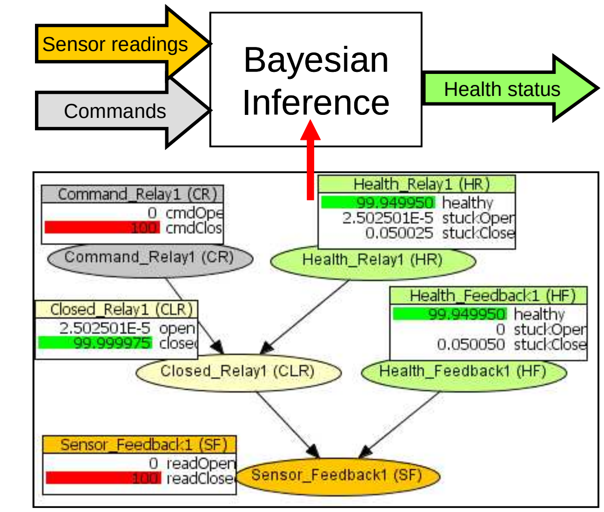

Figure 3: Our Bayesian diagnostic process has as *input* sensor readings for sensor nodes and observed commands for command nodes, and as *output* query nodes that provide the health status of sensors and EPS components.

(iii) Figure 4 and Figure 3 provide a small example of our component-oriented and causal approach to EPS modelling. Here, $\boldsymbol{H}_C = \{$*Health_Relay1*$\}$, $\boldsymbol{H}_S = \{$*Health_Feedback1*$\}$, $\boldsymbol{E}_C = \{$*Command_Relay1*$\}$, $\boldsymbol{E}_S = \{$*Sensor_Feedback1*$\}$, and $\boldsymbol{S} = \{$*Closed_Relay1*$\}$. This BN with five nodes represents an EPS relay with a feedback capability. Causally, the BN represents how the status of a relay (here *Closed_Relay1*) depends on the command given to it, *Command_Relay1*, as well as its health, *Health_Relay1*. In addition, the feedback message from the relay, *Sensor_Feedback1*, depends not only on the relay's status but also on the sensor's health, *Health_Feedback1*.

To solve the EPS health monitoring problem, we dynamically clamp nodes $\boldsymbol{E}_S$ and $\boldsymbol{E}_C$ in the BN using sensor readings and user commands respectively. We then pose a maximum a posteriori hypothesis query MAP($\boldsymbol{Q}$, $\boldsymbol{e}$) over nodes $\boldsymbol{Q}$ for evidence $\boldsymbol{e}$. Here, MAP($\boldsymbol{Q}$, $\boldsymbol{e}$) computes the joint explanation over $\boldsymbol{Q} \subseteq \boldsymbol{X} - \boldsymbol{E}$ with maximal probability, given $\boldsymbol{e}$ (Park & Darwiche 2004). Depending on how $\boldsymbol{Q}$ is chosen, we obtain three different diagnostic queries:

- Diagnosis of components MAP($\boldsymbol{H}_C$,$\boldsymbol{e}$): Query regarding the health of the EPS components $\boldsymbol{H}_C$
- Diagnosis of sensors MAP($\boldsymbol{H}_S$,$\boldsymbol{e}$): Query regarding the health of the EPS sensors $\boldsymbol{H}_S$
- EPS diagnosis MAP($\boldsymbol{H}_E$,$\boldsymbol{e}$): Query regarding the health status of the entire EPS $\boldsymbol{H}_E$ (both components $\boldsymbol{H}_C$ and sensors $\boldsymbol{H}_S$)

While algorithms for efficiently computing MAP have been developed (Park & Darwiche 2004), it can be useful to approximate MAP using MPE (most probable explanation) or MLV (most likely value, which can easily be computed from marginals) (Pearl 1988). We say $\text{MAP}_{\text{MPE}}(\boldsymbol{Q},\boldsymbol{e})$ and $\text{MAP}_{\text{MLV}}(\boldsymbol{Q},\boldsymbol{e})$ respectively for these two approximations.

Returning to Figure 3, we consider $\boldsymbol{H}_E = \{$*Health_Relay1*, *Health_Feedback1*$\}$ and $\boldsymbol{e} = \{$*Command_Relay1* = *cmdClose*, *Sensor_Feedback1* = *readClosed*$\}$. Using computation of marginals, as illustrated in Figure 3, we obtain $\text{MAP}_{\text{MLV}}(\boldsymbol{H}_E,\boldsymbol{e}) = \{$*Health_Relay1* = *healthy*, *Health_Feedback1* = *healthy*$\}$. In other words, given a command to close `Relay1`, and a confirming feedback message from `Feedback1`, it is inferred that both the relay and the feedback mechanism are healthy.

Meeting The Real-Time Challenge

Musliner and his coauthors identified three approaches to real-time AI (Musliner *et al.* 1995); we employ what they call "embedding AI into a real-time system". Specifically, we consider the real-time operating systems (RTOSs) used in current aircraft and spacecraft avionics. These RTOSs are typically based on priority-based preemptive scheduling, where higher-priority tasks preempt lower-priority tasks. Each periodic RTOS task has a priority, a period, a deadline, and a worst-case execution time (WCET). A periodic diagnostic task, when designed as a periodic RTOS task, needs to adhere to these hard real-time requirements (Musliner *et al.* 1995; Mengshoel 2007a).

At the same time, the computational hardness of most BN inference problems is well-known (Cooper 1990; Shimony 1994; Park & Darwiche 2004). In addition, empirical studies have established the difficulty of relatively small application BNs (Shwe *et al.* 1991) as well as synthetic BNs (Mengshoel, Wilkins, & Roth 2006; Mengshoel 2007b).

A designer of BN-based diagnostic systems must carefully align resource consumption with the resource bounds imposed by the computational platform. The compilation approach to probabilistic inference is attractive in such settings. We mention two compilation paradigms, namely compilation to clique trees (Lauritzen & Spiegelhalter 1988; Andersen *et al.* 1989) and compilation to arithmetic circuits (Darwiche 2003; Chavira & Darwiche 2007). The arithmetic circuit paradigm is based on the observation that a BN can be represented as a multi-variate polynomial (MVP) in which terms consist of probabilities from the BN's CPTs and indicators take into account evidence. Unfortunately, an MVP grows exponentially with the size of a BN, hence one compiles a BN into an equivalent but (typically) more compact arithmetic circuit. An example is shown in Figure 4. In many cases, sparse arithmetic circuits exist for BNs with 100s or 1000s of nodes. The arithmetic circuit's size depends on a BN's graphical and local structure: if BN has local structure, the arithmetic circuit may be small despite large treewidth. A range of probabilistic queries — including MAP, MPE, and marginals/MLVs — can be computed using arithmetic circuits.

We now very briefly summarize the compilation to arithmetic circuits. Prior to compilation, we modify the CPTs to store pointers to AC nodes rather than numbers. For example, if 0.1 is stored in a particular slot of some CPT, then this number would be replaced with a pointer to a single AC node (sink) labeled with 0.1. Also prior to compilation, for each BN variable, we add a new table over just that variable

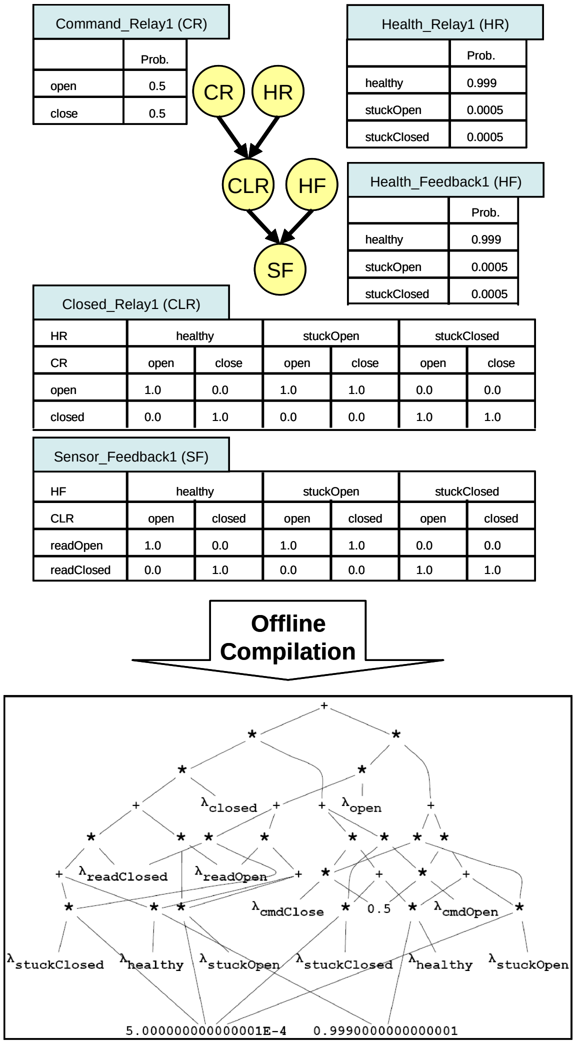

Figure 4: Automatic compilation of a BN representing an electrical power system relay (top) into an equivalent arithmetic circuit (bottom). This BN consists of the five nodes in the upper right corner of Figure 2.

representing the values of that variable. For example, variable X with values 0 and 1 would generate a table over X where the first slot contains a pointer to an AC node (sink) labeled with λ_0 and the second slot contains a pointer to an AC node (sink) labeled with λ_1.

After these two preprocessing steps, we run a slightly modified version of the standard variable elimination (VE) algorithm (Zhang & Poole 1996; Dechter 1999). The only difference occurs when the standard version wishes to add or multiply two numbers. In each of these situations, the standard algorithm will identify two slots A and B in tables, add (multiply) the two numbers residing there, and store the result back into some slot C of some table. When the modified algorithm looks into A and B, it finds pointers to AC nodes α and β rather than numbers. Instead of performing the arithmetic operation, the modified algorithm creates a new AC node γ labeled with "+" or "*", makes α and β children of γ, and stores a pointer to γ into C. Upon completion, standard VE yields a single table containing a single slot containing a number. The modified algorithm will be the same, except that rather than a number, we will have a pointer to an AC node, which is the root of the compiled arithmetic circuit. By exploiting local structure, this modified VE algorithm has compiled BNs with prohibitively large treewidths (Chavira & Darwiche 2007; Chavira 2007). Using our Offline Generation approach, we auto-generate BNs such that the structure of the underlying EPSs is maintained to a great degree; in addition a high proportion of CPTs generated are deterministic (see Figure 4 for examples). The BNs developed for ADAPT in other words have substantial local structure, and AC compilation has worked well on similar BNs in the past.

Experimental Results

We now turn to experiments using ADAPT and different inference algorithms. Experiments are divided into two sets: hand crafted, real-world scenarios from ADAPT and simulated scenarios that were automatically generated from an ADAPT BN. In both cases, we executed probabilistic queries over the health variables $\boldsymbol{H}_E$ in order to find out which components or sensors, if any, were in non-healthy states. Figure 3 presents the inputs and outputs in terms of the ADAPT BN along with a small example.

The ACE system was used to (i) compile an ADAPT BN into an arithmetic circuit and (ii) evaluate that arithmetic circuit (see `http://reasoning.cs.ucla.edu/ace/` for details on ACE). The timing measurements reported here were made on a PC with an Intel 4 1.83 Ghz processor, 1 GB RAM, and Windows XP.

Experiments using Real-World Data

For experimentation using real-world data, EPS failure scenarios were generated using the ADAPT EPS at NASA Ames. These scenarios cover both component failures (experiments 304, 306, 309, and 310 in Table 1) and sensor failures (experiments 305, 308, and 311); many previous efforts have only considered one type of failure. In each of these experiments, ADAPT's initial state was as follows: Circuit breakers were commanded closed; they had evidence $\boldsymbol{e}$ clamped to *cmdClose*. Relays were commanded open; they had evidence clamped to *cmdOpen* in $\boldsymbol{e}$. In this initial state, all health nodes $\boldsymbol{H}_E$ are deemed healthy when computing MAP, MAP_{MPE}, and MAP_{MLV}.

After ADAPT system reconfigurations and fault insertion (for example insertion of "Relay EY260 failed open" – see ID 304 in Table 1), the ADAPT BN or an arithmetic circuit compiled from it is used to compute a diagnosis. The variant of the ADAPT BN used here was largely auto-generated and contains 434 nodes and 482 edges; the BN node cardinalities range from 2 to 4 with mean 2.27. ACE was used to compute MPEs and marginals/MLVs. We report here on the queries $\text{MAP}_{\text{MPE}}(\boldsymbol{H}_E, \boldsymbol{e})$ and $\text{MAP}_{\text{MLV}}(\boldsymbol{H}_E, \boldsymbol{e})$ computed by ACE. To compute $\text{MAP}(\boldsymbol{H}_E, \boldsymbol{e})$, SamIam was used (see `http://reasoning.cs.ucla.edu/samiam/` for details).

ID	Fault Description	Diagnosis: MAP, MAP_{MPE}, and MAP_{MLV}	Correct
304	Relay EY260 failed open	*Health_relay_ey260_cl = stuckOpen*	Yes
305	Relay feedback sensor ESH175 failed	*Health_relay_ey175_cl = stuckOpen*	Yes
306	Circuit breaker ISH262 tripped	*Health_breaker_ey262_op = stuckOpen*	Yes
308	Voltage sensor E261 failed	*Health_e261 = stuckVoltageLo*	Yes
309	Battery BATT1 voltage low	*Health_battery1 = stuckDisabled*	Yes
310	Inverter INV1 failed off	*Health_inv1 = stuckOpen*	Yes
311	Load sensor LT500 failed	*Health_LT500 = stuckLow*	Yes

Table 1: Diagnostic results for different fault scenarios (with IDs 304, 305, ...) for the electrical power system testbed ADAPT.

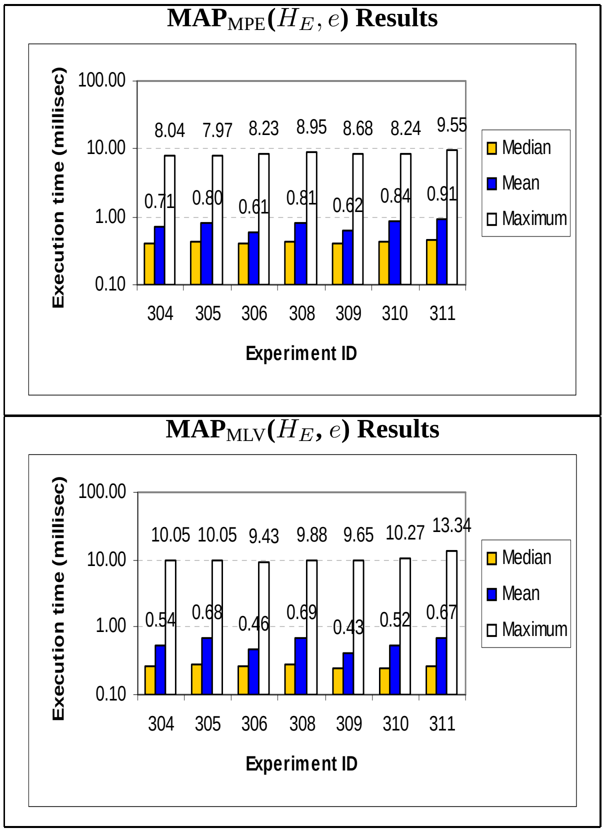

Figure 5: Execution time results for ACE for the ADAPT testbed. *Top:* Results for the most probable explanation (MPE); *Bottom:* Results for the most likely value (MLV).

The results of the ADAPT experiments are provided in Table 1 and in Figure 5. Since $\boldsymbol{H}_E$ contains over 120 nodes, we only show the variables deemed to be non-healthy in Table 1. Further, the diagnostic results of the queries $\text{MAP}_{\text{MPE}}(\boldsymbol{H}_E, \boldsymbol{e})$, $\text{MAP}_{\text{MLV}}(\boldsymbol{H}_E, \boldsymbol{e})$, and $\text{MAP}(\boldsymbol{H}_E, \boldsymbol{e})$ turned out to be the same, hence we consolidate them into one column in Table 1. ADAPT uses a 2 Hz sampling rate, and a probabilistic query was posed to ACE after each sample in an experimental run. The execution time statistics displayed in Figure 5 are based on the execution times for all probabilistic queries during an experimental run. Each execution time is for an entire inference step, i.e. translating measurements to evidence, committing evidence to the

Inference Time (ms)	MPE		Marginals	
	VE	ACE	CTP	ACE
Minimum	17.25	0.1967	8.527	0.4934
Maximum	38.45	2.779	54.51	5.605
Median	17.63	0.1995	9.204	0.5624
Mean	17.79	0.2370	10.02	0.6981
St. Dev.	1.513	0.2137	4.451	0.6669

Table 2: Results for different inference algorithms (VE, ACE, and CTP) when computing MPEs and marginals using data generated from the ADAPT BN.

arithmetic circuit, and evaluating the arithmetic circuit.

Our main observations regarding these experiments are as follows. First, we see in Table 1 that the different diagnostic queries correctly diagnose all these component and sensor failure scenarios. Second, we emphasize the fast and predictable inference times for the ACs in Figure 5. These are both very important factors in real-time electrical power system health management.

Experiments using Simulated Data

The variant of the ADAPT BN used here was completely auto-generated and contains 453 nodes and 509 edges; the BN node cardinalities range from 2 to 4 with mean 2.28. Simulated data was created by a program that (i) generated a set of failure scenarios according to the probabilities of the ADAPT BN's health nodes $\boldsymbol{H}_E$, and (ii) for each failure scenario, generated an evidence set on sensor nodes. This large number of evidence sets was then run through different inference systems. In addition to arithmetic circuit evaluation (ACE), we performed experiments with variable elimination (VE) and clique tree propagation (CTP).

Table 2 summarizes the results of experiments with 200 simulated evidence sets generated from the ADAPT BN. ACE is, on average, over 75 times faster than VE when computing MPEs (see Table 2). In addition, we note how ACE can compute all marginals, or $\text{MAP}_{\text{MLV}}(\boldsymbol{Q},\boldsymbol{e})$, using just slightly more time than what is used for MPEs, or $\text{MAP}_{\text{MPE}}(\boldsymbol{Q},\boldsymbol{e})$. In other words, ACE can compute over 400 probabilities 25 times faster than VE computes a single probability. CTP can be used to compute marginals in order to overcome VE's limitation of computing only one probability at a time, but even CTP is over 14 times slower and has higher standard deviation than AC.

In summary, VE, CTP, and ACE all run quite efficiently on the ADAPT system, but ACE is one or two orders of magnitude more efficient than the other algorithms, while having lower standard deviation. Diagnostic inference for ADAPT is therefore very efficient for two reasons. First, the BN was carefully generated, using our novel auto-generation algorithm, in manner that supports efficient inference using any reasonable exact inference algorithm. Second, the particular arithmetic circuit algorithms we have emphasized here, as implemented in ACE, provides very large additional gains.

Conclusion

Electrical power systems (EPSs) are crucially important in spacecraft and aircraft. In this paper, we have presented a probabilistic approach to fault diagnosis in such systems. Specifically, we have discussed how ADAPT, an electrical power system testbed at NASA, can be represented as Bayesian networks and arithmetic circuits which are used to answer diagnostic queries. We have highlighted two challenges, the modelling and real-time reasoning challenges, often associated with the development of model-based diagnostic engines for spacecraft and aircraft, and shown how they are overcome in our setting.

Meeting the modelling challenge, we have discussed how EPS BNs can easily be described in a novel, easy-to-use specification language that is component-based and reflects EPS structure. This language is the basis for our auto-generation of EPS BNs, including ADAPT BNs with over 400 nodes. We have also considered how to meet the real-time requirements of spacecraft and aircraft. Our approach meets this challenge by compilation into arithmetic circuits, where inference is fast and predictable, thereby enabling embedding into real-time settings. Our probabilistic fault diagnosis methodology has been successfully evaluated using real-world data from ADAPT.

Acknowledgments This material is based upon work supported by NASA under awards NCC2-1426 and NNA07BB97C. We would also like to thank Ann Patterson-Hine (NASA Ames Research Center) for her central role in the development of the ADAPT testbed.

References

Andersen, S. K.; Olesen, K. G.; Jensen, F. V.; and Jensen, F. 1989. HUGIN—a shell for building Bayesian belief universes for expert systems. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, volume 2, 1080–1085.

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

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

Chavira, M. 2007. *Beyond Treewidth in Probabilistic Inference*. Ph.D. Dissertation, University of California, Los Angeles.

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

Cooper, F. G. 1990. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence* 42:393–405.

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

Dechter, R. 1999. Bucket elimination: A unifying framework for reasoning. *Artificial Intelligence* 113(1-2):41–85.

Lauritzen, S., and Spiegelhalter, D. J. 1988. 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.

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

Mengshoel, O. J. 2007a. 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)*, 330–337.

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

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

Park, J. D., and Darwiche, A. 2004. Complexity results and approximation strategies for MAP explanations. *Journal of Artificial Intelligence Research (JAIR)* 21:101–133.

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

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

Shimony, E. 1994. Finding MAPs for belief networks is NP-hard. *Artificial Intelligence* 68:399–410.

Shwe, M.; Middleton, B.; Heckerman, D.; Henrion, M.; Horvitz, E.; Lehmann, H.; and Cooper, G. 1991. Probabilistic diagnosis using a reformulation of the INTERNIST-1/QMR knowledge base: I. The probabilistic model and inference algorithms. *Methods of Information in Medicine* 30(4):241–255.

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

Zhang, N. L., and Poole, D. 1996. Exploiting causal independence in Bayesian network inference. *Journal of Artificial Intelligence Research* 5:301–328.