Carnegie Mellon University Silicon Valley

sv.cmu.edu

Visualization of Analytical Processes

Ole J. Mengshoel
Carnegie Mellon University
Silicon Valley Campus

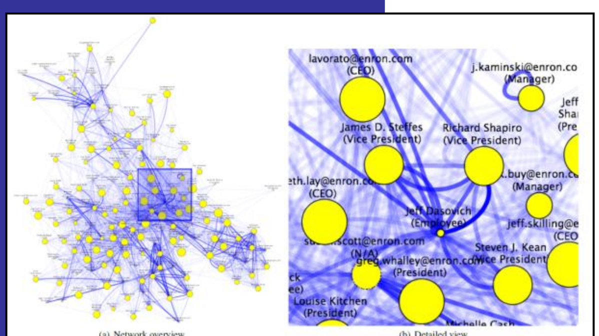
December 9, 2011
FODAVA Annual Review
Georgia Tech

Project Overview

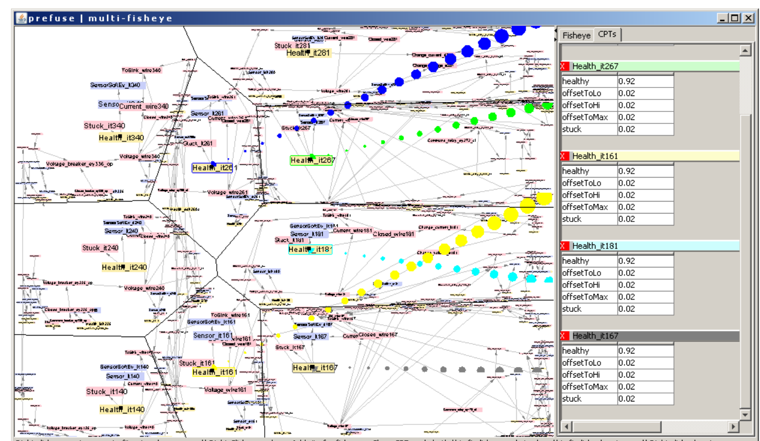
: Improve the visualization of analytical processes, in particular for probabilistic graphical (Bayesian network) models and other networks models.

: 2009-2011 (currently no-cost extension)

Faculty: Mengshoel, Selker, and Ilic



(a) Network overview (b) Detailed view



Areas of Research

Probabilistic and statistical models, algorithms:

- Probabilistic graphical models: Bayesian networks, …
- Inference: Diagnosis, prognosis, …
- Machine learning

Interactive visualization:

- Network visualization
- Multi-view, Multi-focus, …

Stochastic and randomized algorithms:

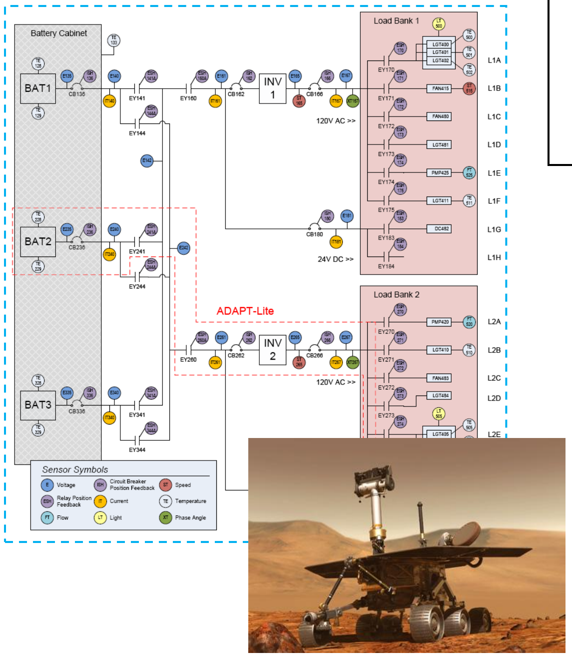
- Stochastic local search
- Evolutionary algorithms

Applications and demonstrations:

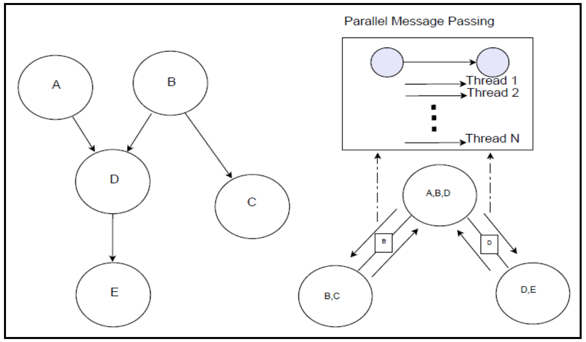
- Challenging and large-scale applications, multi-media data sets, …
- Scalability of algorithms, visualizations, user interactions, …

Research Directions

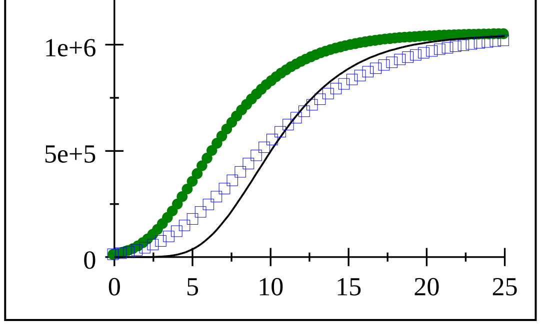
Electrical Power System Diagnosis using Probabilistic Computation



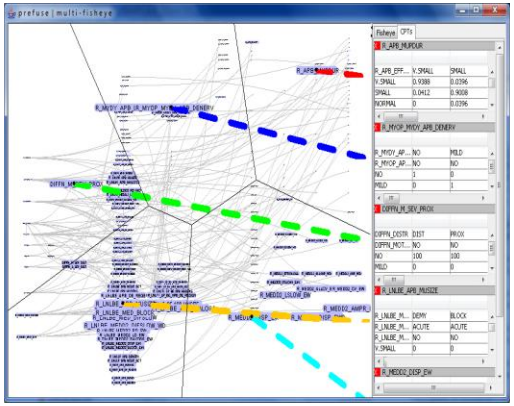
Fast Belief Propagation Using GPU Parallelization in Junction Trees



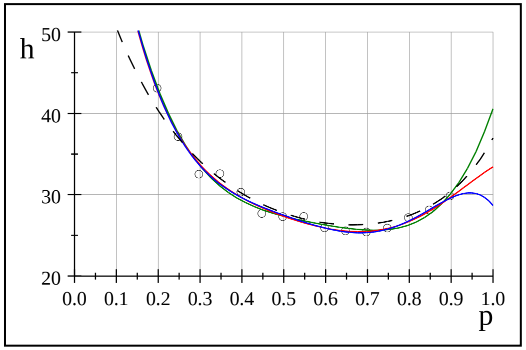
Understanding Scalability of Bayesian Network Computation using Junction Tree Growth Curves



Multi-Fisheye, Multi-View for Interactive Visualization of Large Networks

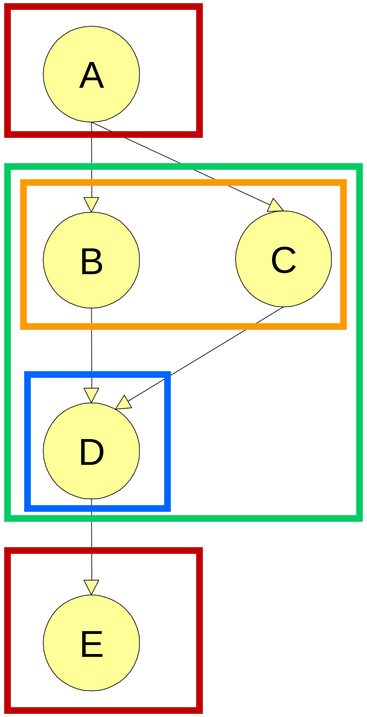


Stochastic Search for Computing Most Probable Explanations in Bayesian



Bayesian Network Inference

- Bayesian network inference answers these queries:
  - *Marginal/MLV*: Given evidence at some nodes, infer posterior probability/most likely value (MLV) over one node
  - *Most probable explanation (MPE)*: Given evidence, find explanation with greatest probability over remaining nodes
  - *Maximum aposteriori probability (MAP)*: Given evidence, find explanation with greatest probability over some nodes
- Computational hardness [Cooper, 1990; Shimony, 1994; Roth 1996]:
  - Care is needed, in modeling, machine learning, and inference
- Inference algorithms:
  - *Exact*: Clique tree propagation [Lauritzen & Spiegelhalter, 1988]; Arithmetic circuit evaluation [Darwiche, 2003; Darwiche & Chavira, 2007]; …
  - *Approximate*: Stochastic local search [Kask & Dechter, 1999; Mengshoel, 1999; Mengshoel 2008]; Variational inference; …



Need for Resilient Operations and System Health Management

On September 2, 1998, Swissair 111 crashed into the Atlantic Ocean, killing all 229 people onboard. Probably, *wires short-circuited* and led to a fire.

A *battery failure* occurred on the Mars Global Surveyor on November 2, 2006. A software error caused the battery to overheat due to over-exposure to sunlight.

In 1999, the Mars Polar Lander crashed into the surface of Mars, most likely due to a premature engine shutdown because of spurious lander leg signals.

For the Mars rover SPIRIT, a full on-board file system caused reboot-loop after landing.

On June 4, 1996, software on the Ariane V rocket, reused from Ariane IV, overflowed and lead to its destruction.

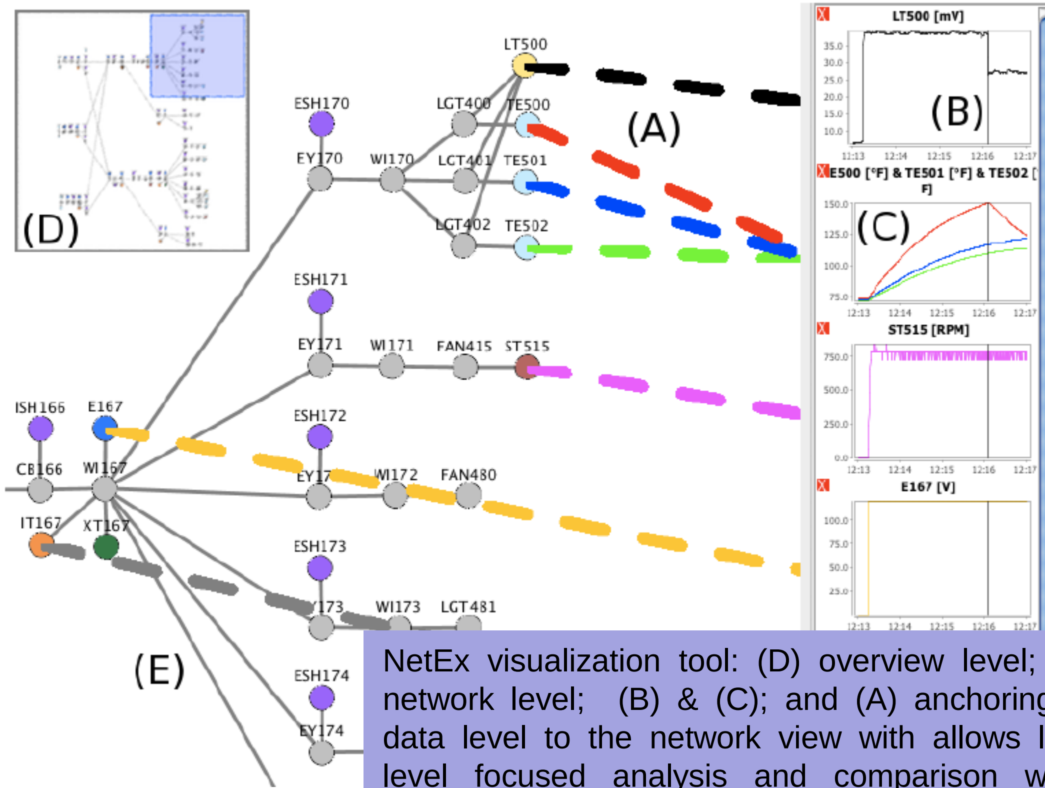
Multi-View Overview+Detail for Networks

OBJECTIVE

Making multiple and multi-step comparisons across different parts of a data corpus and across multiple representational levels in a complex data set.

FEATURES

- Visual encoding of data properties
- Overview + detail
- Multi-focus + context
- Bubbles anchoring node information to the network



NetEx visualization tool: (D) overview level; (E) network level; (B) & (C); and (A) anchoring of data level to the network view with allows low-level focused analysis and comparison while preserving network structure.

RESULTS

In experiments with data from an electrical power network we demonstrated how NetEx makes fault diagnosis easier. Results from a **user study** with 25 subjects suggests that NetEx enables more accurate isolation of faults in multi-fault situations.

Carnegie Mellon University
Silicon Valley

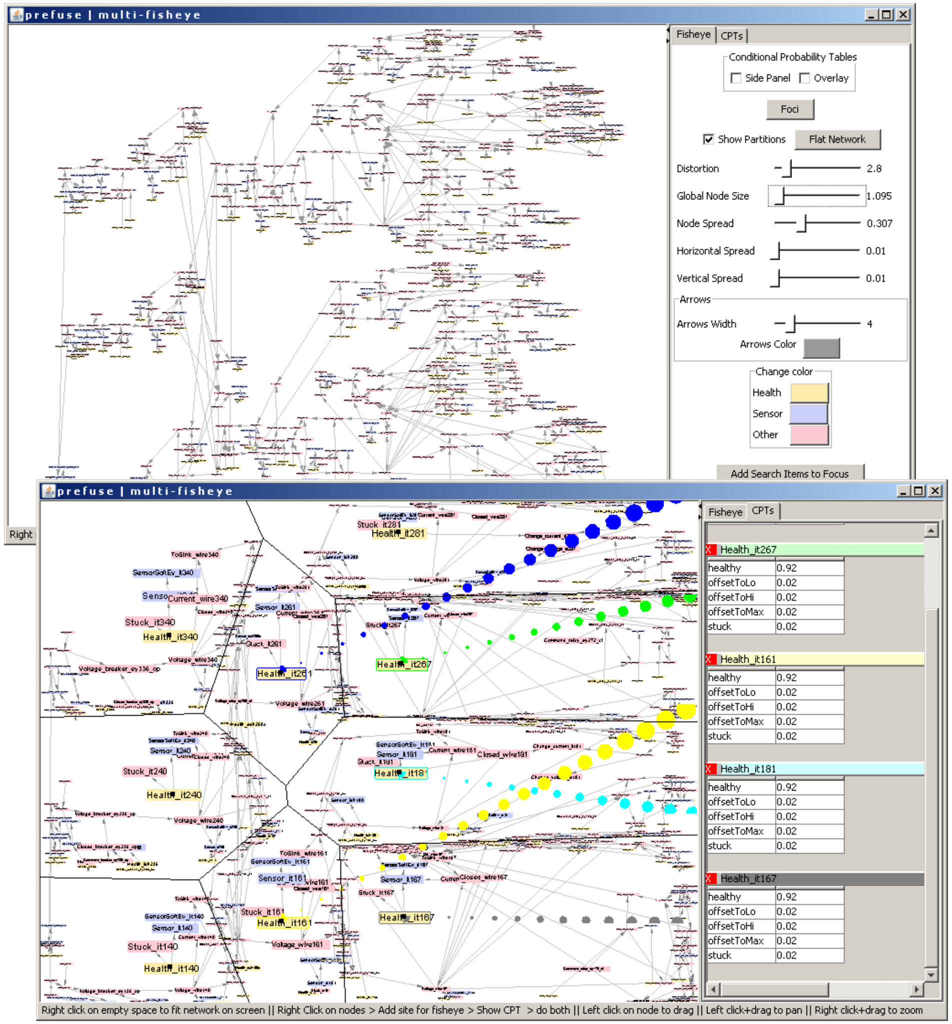
Multi-View Focus+Context for Networks

OBJECTIVE

Improve the applicability of multi-fisheye to exploration of labeled networks, including Bayesian network (BN) problem instances. Focus on large-scale but in-memory networks.

DESCRIPTION

A focus+context visualization tool that supports visualization of multiple fisheye distortions in network (Bayesian networks, for example). Voronoi edges separate the fisheyes, and data boxes with details (such as Bayesian network conditional probability tables) are created for fisheyes and their neighboring nodes.



RESULTS

The tool supports interactive and simultaneous creation of up to 10-20 readable node labels by means of fisheye distortion in large-scale (Bayesian) networks. Node context, including network edge connection patterns and relative location, is preserved.

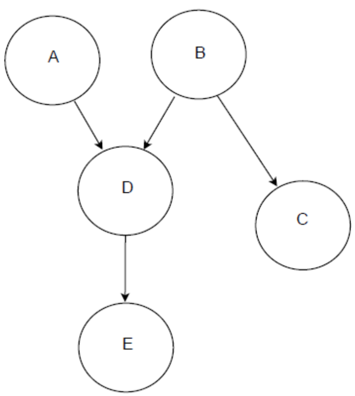
Belief Propagation by Fast GPU Message Passing in Junction Trees

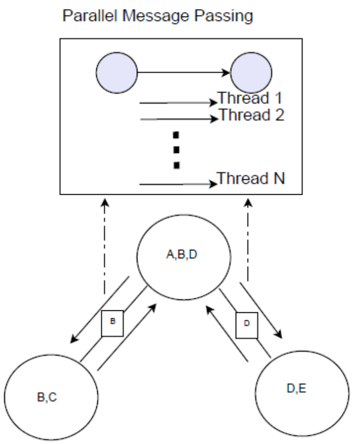
OBJECTIVE

Speed up Bayesian network computation when junction trees are being used; use graphics processing units (GPUs).

DESCRIPTION

An algorithm in which message passing in performed in parallel, benefiting situations with large cliques and large separators

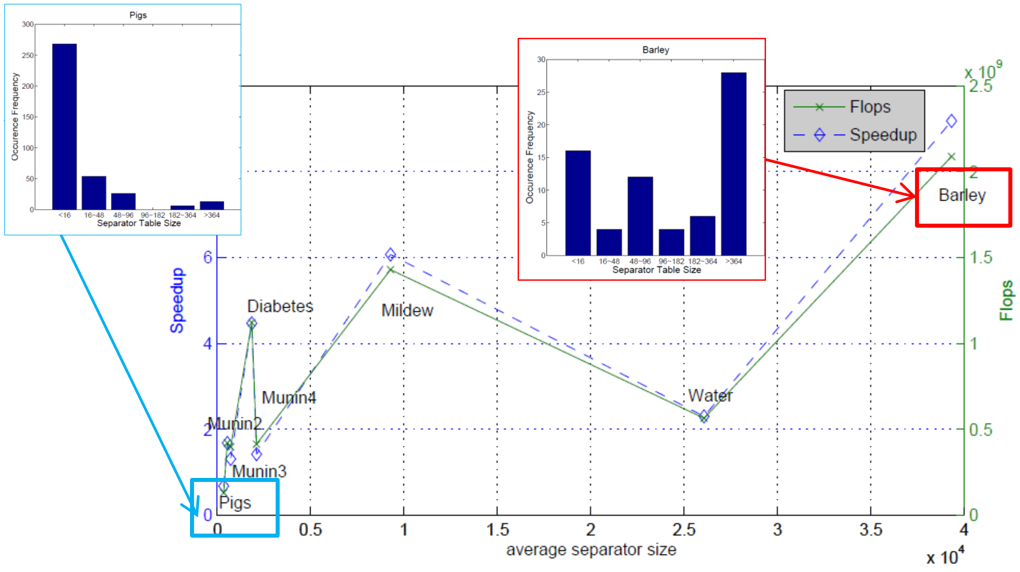




RESULTS

Analytical and experimental speed up – best experimental speed up result to date is 918%.

$$Speedup = \frac{\sum_i \sum_{k \in Ne(\mathcal{C}_i)} (|\phi \mathcal{X}_i| + |\phi \mathcal{X}_k|)}{2(n-1)\tau + \sum_i \sum_{k \in Ne(\mathcal{C}_i)} \frac{(|\phi \mathcal{X}_i| + |\phi \mathcal{X}_k|)}{|\phi s_{ik}|}}.$$

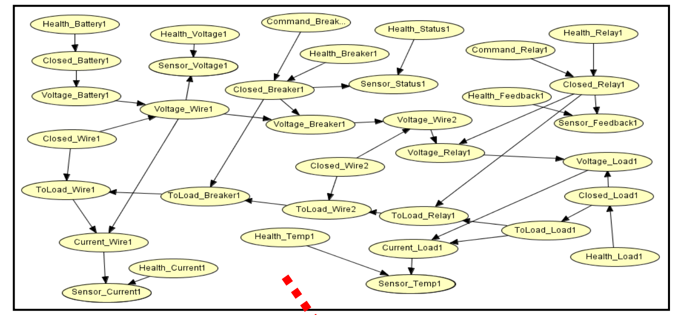


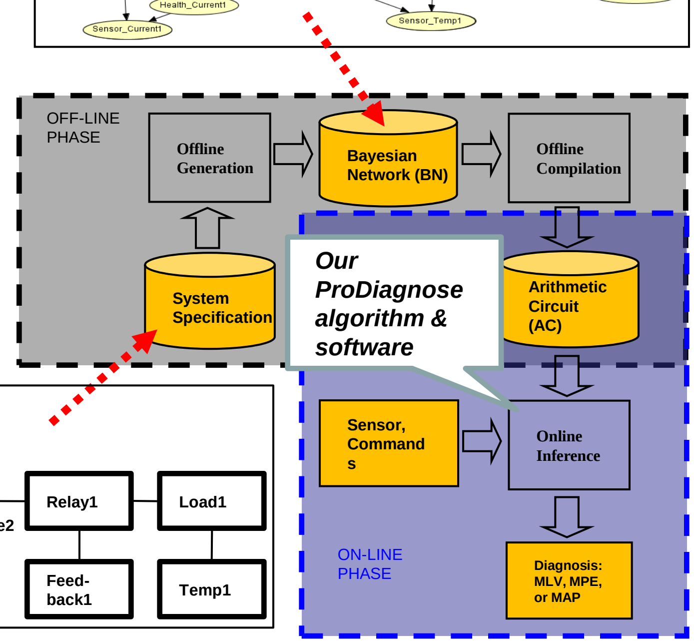
Carnegie Mellon University
Silicon Valley

Bayesian Methods for Diagnostics (1)

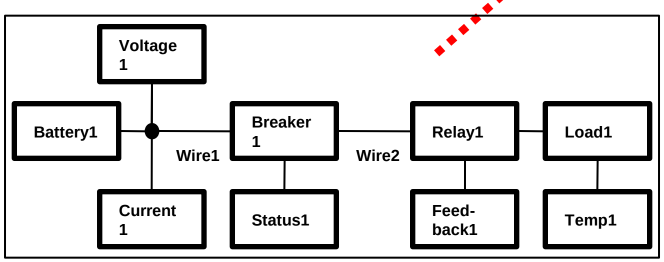
: Tackle system health management and diagnostic challenges: Large & complex systems; Hybrid systems (discrete & continuous behavior); Hard diagnostic problems; Real time requirements.

: Develop probabilistic diagnosis approach, ProDiagnose: Auto-generation of Bayesian network; Compilation of Bayesian networks to real-time arithmetic circuits; Diagnose discrete and continuous faults on-line.



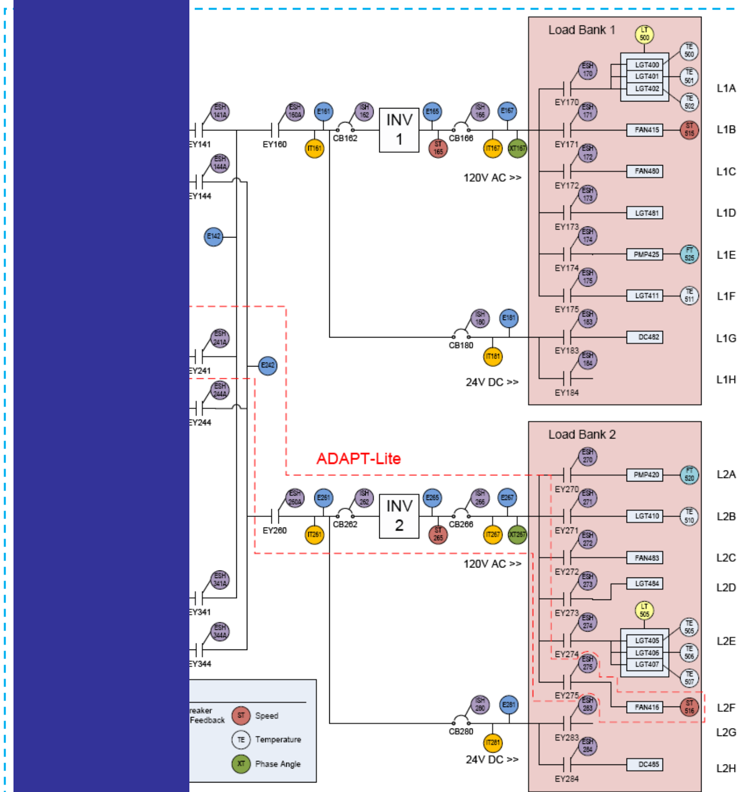


Electrical Power System



Bayesian Methods for Diagnostics (2)

[redacted]*. ADAPT – Electrical power system testbed at NASA ARC.*



[redacted]*. Two conditions: Our novel cumulative sum (CUSUM) technique (i) enabled or (ii) disabled.*

	CUSUM	
Metric	**Enabled**	**Disabled**
Detection Accuracy	92.31%	46.15%
False Positives Rate	0%	0%
False Negatives Rate	8.82%	61.76%
Mean Time To Detect	17.97 s	28.36 s
Mean Time To Isolate	72.27 s	51.14 s

Bayesian Reasoning for Diagnostics:
Operates in a state space of size > 2^{500} in time < 1 ms.

GPUs for Speeding up Bayesian Network Computation

Parallel and Distributed Computing

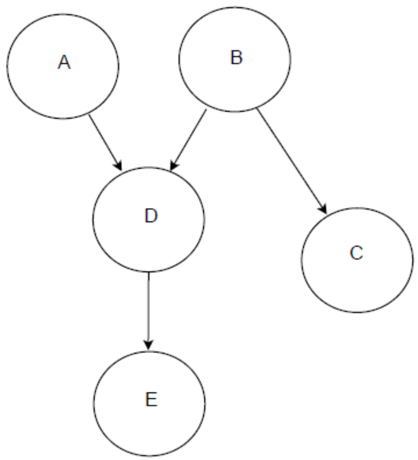
Graphics processing units (GPUs): Promise to dramatically up the performance of processing in the cloud and on the mobile device.

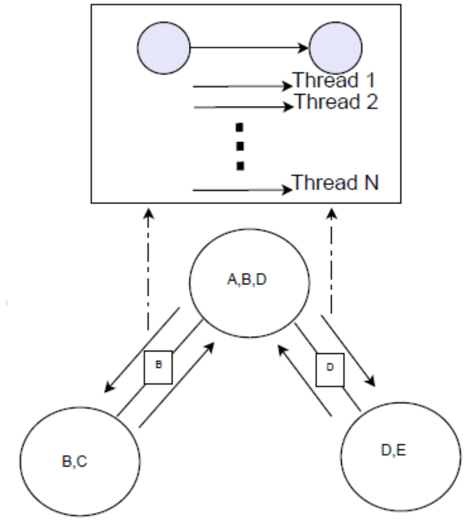
Speed up performance of processing in the cloud – integration with analytics software.

GPUs are moving onto mobile devices, and within the next year or two we expect them to be programmable through CUDA or other programming languages.

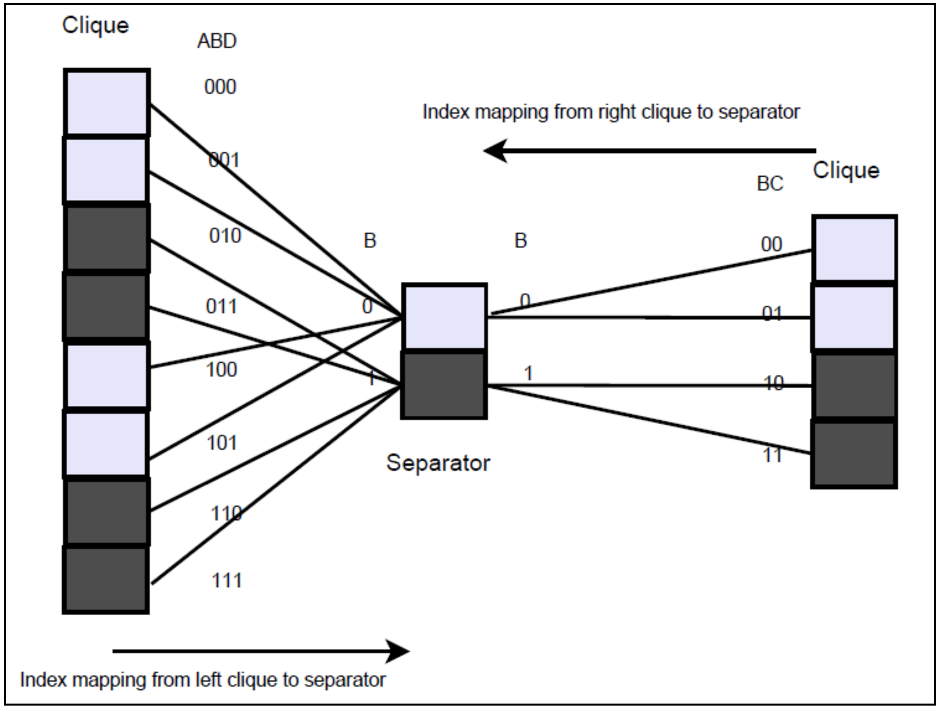
Motivation and Approach

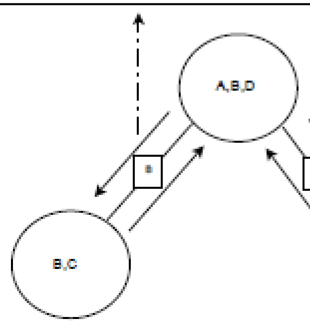
- Belief propagation in junction trees may be computationally intensive due to:
  - The topology and connectedness of Bayesian networks
  - High cardinality of one or more nodes in cliques with sufficiently high number of nodes
- Observations:
  - During message passing, computations associated with different separator tables are independent
  - Some junction trees contain large cliques and separators
- Our approach:
  - Compute each message in parallel
  - Substantial parallelism opportunity when neighboring cliques and separators are large
  - Non-invasive embedding in original junction tree message passing algorithms





Fast Message Passing





Step 1: Marginalization

$$\phi^*_{\mathcal{S}_{ik}} = \sum_{\mathcal{X}_i / \mathcal{S}_{ik}} \phi_{\mathcal{X}_i}$$

Step 2: Scattering

$$\phi^*_{\mathcal{X}_k} = \phi_{\mathcal{X}_k} \frac{\phi^*_{\mathcal{S}_{ik}}}{\phi_{\mathcal{S}_{ik}}}$$

GPU Message Computation and Speedup

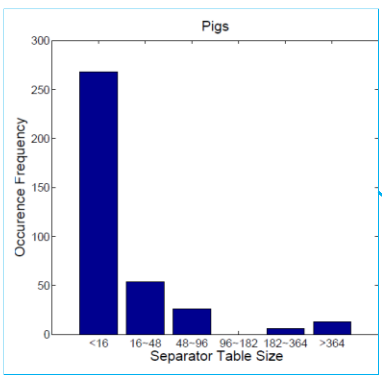
Algorithm 1 $Message\_Passing(\phi_{\mathcal{X}_i}, \phi_{\mathcal{X}_k}, \phi_{\mathcal{S}_{ik}})$

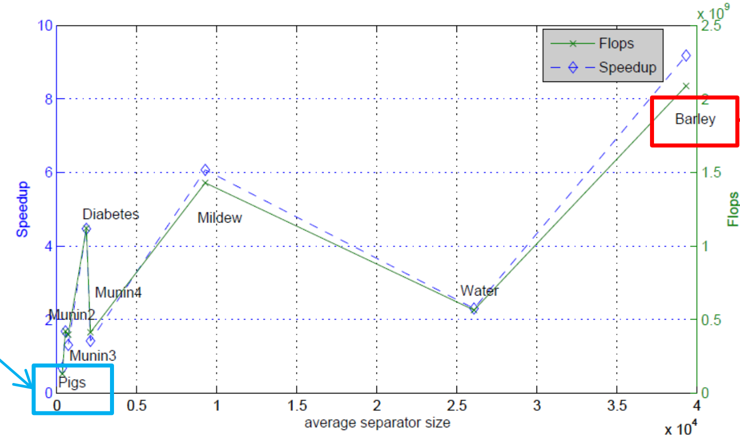
Input: $\phi_{\mathcal{X}_i}, \phi_{\mathcal{X}_k}, \phi_{\mathcal{S}_{ik}}$.
for $j = 1$ *to* $|\phi_{\mathcal{S}_{ik}}|$ **in parallel do**
  sep_star=0;
  for $n = 1$ *to* $|\mu_{\mathcal{X}_i, s_j}|$ **do**
    sep_star[j] $=$ sep_star[j]$+\phi_{\mathcal{X}_i}(\mu_{\mathcal{X}_i, s_j}[n])$
  end for
  for $n = 1$ *to* $|\mu_{\mathcal{X}_k, s_j}|$ **do**
    $\phi_{\mathcal{X}_k}(\mu_{\mathcal{X}_k, s_j}[n]) = \frac{\text{sep\_star[j]}}{\phi_{\mathcal{S}_{ik}}[j]} \phi_{\mathcal{X}_k}(\mu_{\mathcal{X}_k, s_j}[n])$
  end for
end for

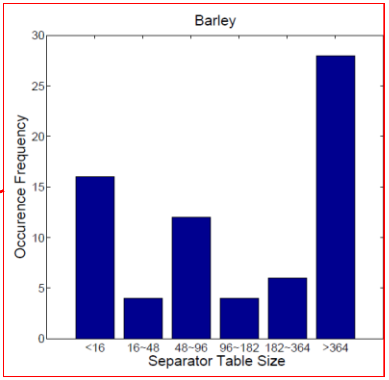
$$Speedup = \frac{\sum_i \sum_{k \in Ne(\mathcal{C}_i)} (|\phi_{\mathcal{X}_i}| + |\phi_{\mathcal{X}_k}|)}{2(n-1)\tau + \sum_i \sum_{k \in Ne(\mathcal{C}_i)} \frac{(|\phi_{\mathcal{X}_i}| + |\phi_{\mathcal{X}_k}|)}{|\phi_{\mathcal{S}_{ik}}|}}.$$

GPU Parallelization: Experiments

Dataset	Mildew	Diabetes	Barley	Pigs	Munin2	Munin3	Munin4	Water
# of JT nodes	28	337	36	368	860	904	872	20
Max. CPT size	4,372,480	190,080	7,257,600	177,147	504,000	156,800	784,000	995,328
Min. CPT size	336	495	216	27	4	4	4	9
Ave. CPT size	341,651	32,443	512,044	1,927	5,653	3,443	16,444	173,297
Max. SPT size	71,680	11,880	907,200	59,049	72,000	22,400	112,000	147,456
Min. SPT size	72	16	7	3	2	2	2	3
Ave. SPT size	9,273	1,845	39,318	339	713	553	2,099	26,065
BP on GPU [ms]	53	94	106	75	125	104	342	52
BP on CPU [ms]	355	397	974	51	210	137	473	120
Speedup	6.70	4.22	9.19	0.68	1.68	1.32	1.38	2.31





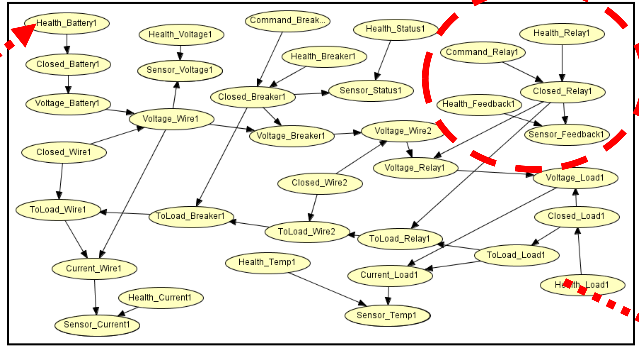


System Health Management using Bayesian Networks

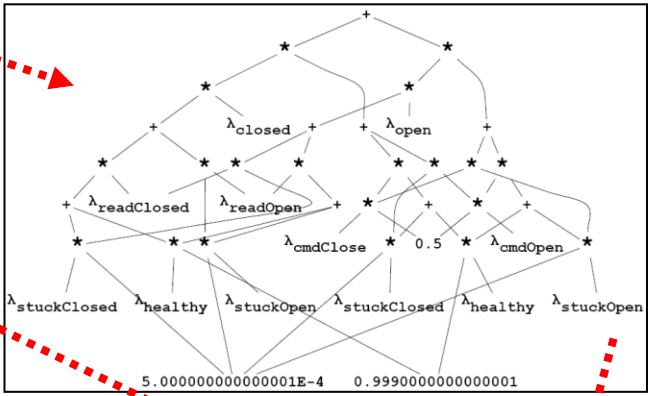
Architecture using Bayesian Networks

Each health variable has at least two states (healthy and faulty), thus enabling the diagnoses of zero, one, two, or more faults.

Bayesian network

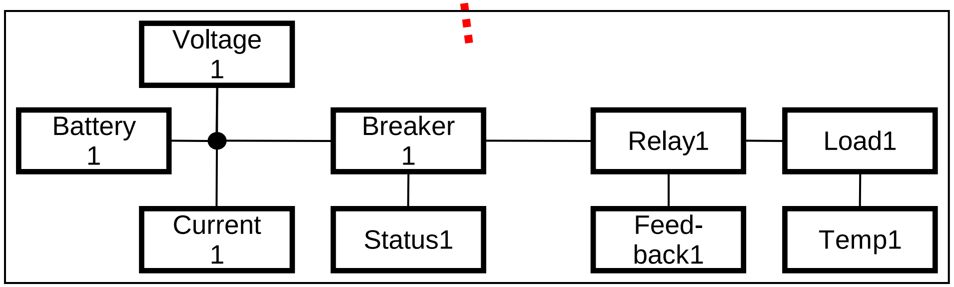


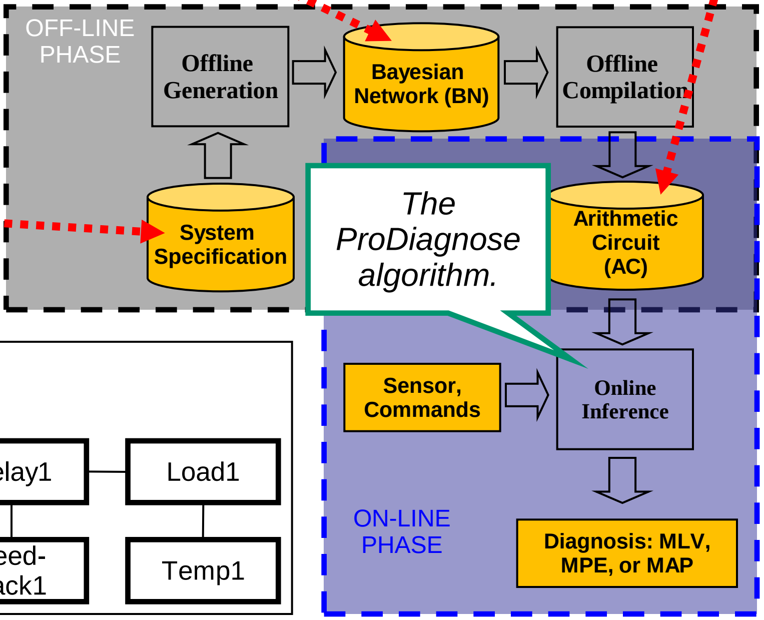
Arithmetic circuit



Specification language

```
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 ;
```





Fault Types

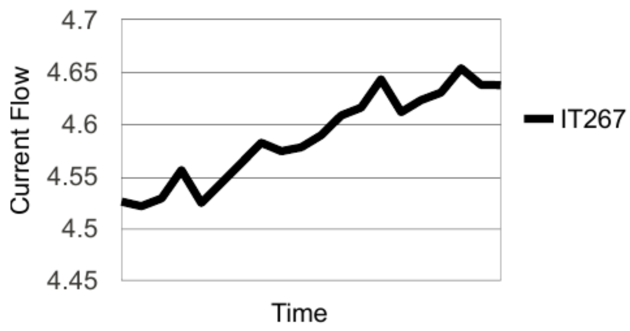
Independent faults
- Abrupt
  - Permanent
    - Discrete
    - Continuous (parametric)
  - Intermittent
- Drift (incipient)

Dependent faults
- Common cause
- Cascading

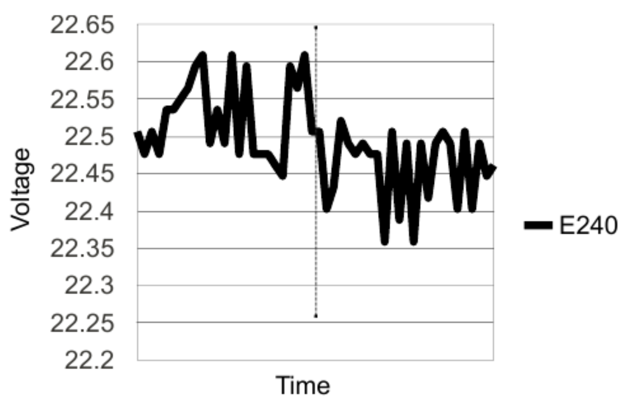
Bayesian networks in general

Problem-1 (DP1) and Problem-2 (DP2) of diagnostics challenge

Continuous drift fault



Continuous abrupt (offset) fault



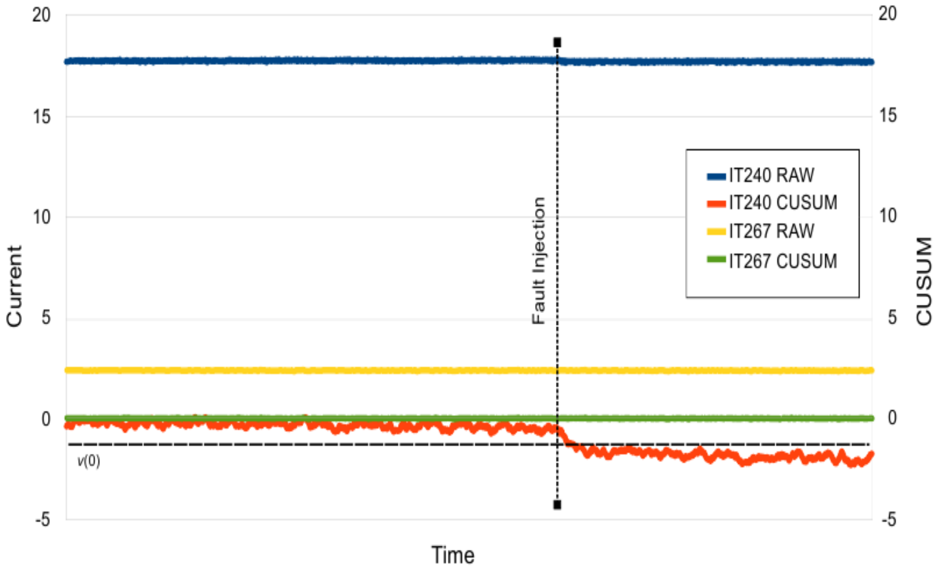
Cumulative Sum (CUSUM)

Mathematical definition of CUSUM:

$$\delta_p(t) = [s_p(t) - s_p(t-1)] + \delta_p(t-1)$$

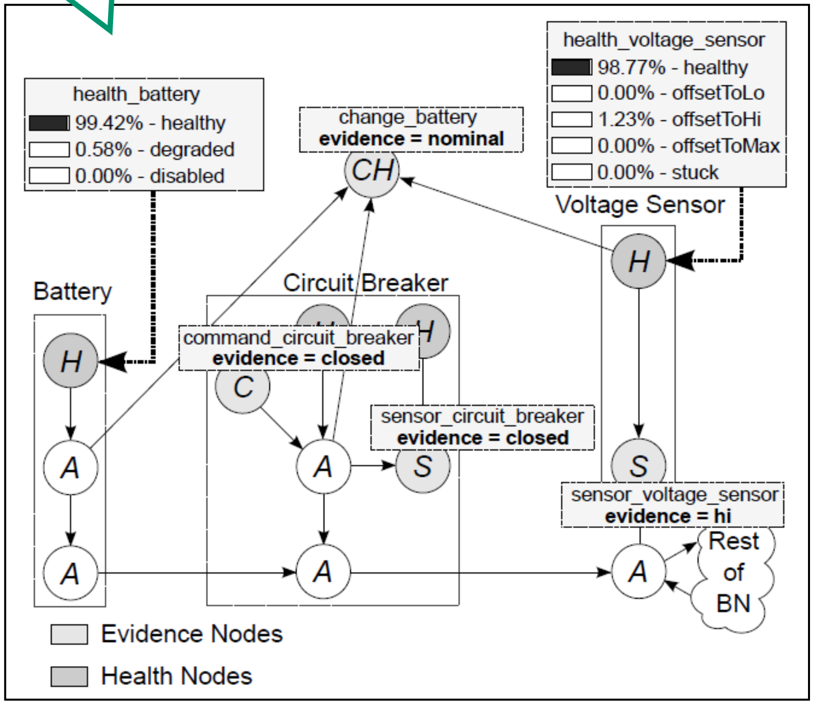
Graph illustrating CUSUM on current readings:

- The blue and orange plots represent the raw sensor readings (span of 4 minutes)
- The red and green plots represent the CUSUM values of these respective raw sensor readings
- The vertical dotted line represents the time of fault injection.
- Benefit of CUSUM: It [redacted] and [redacted] signals of [redacted] faulty

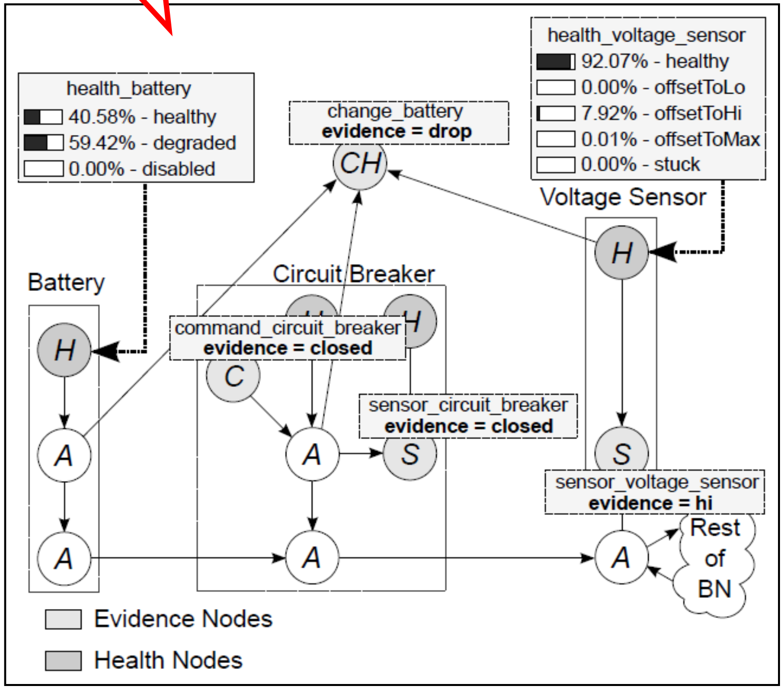


CUSUM – Continuous Offset Faults

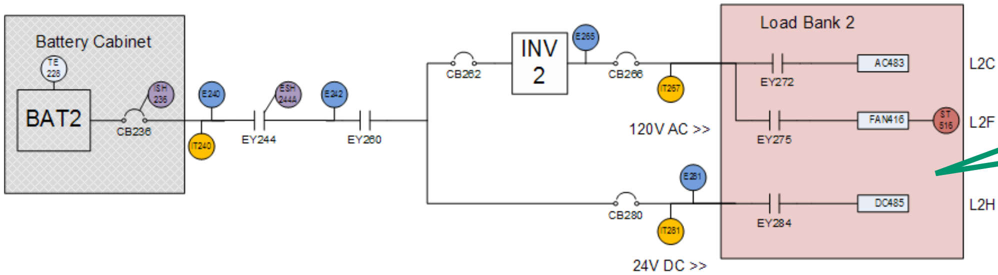
Nominal case



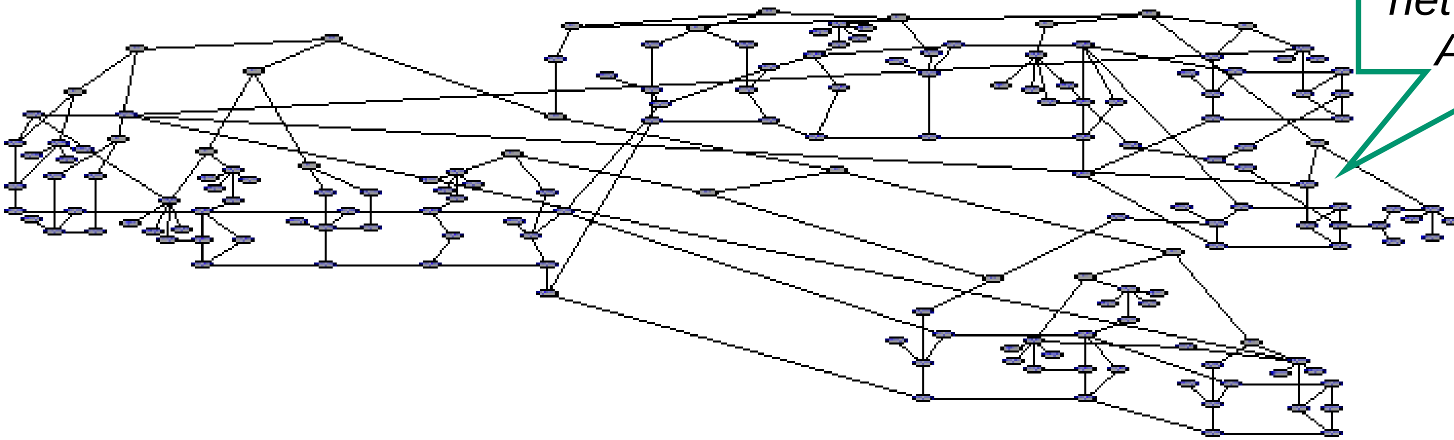
Fault case



Experimental Bayesian Network



ADAPT DP1



The Bayesian network model of ADAPT DP1

Summary Statistics:

- DP1 Bayesian network:
  - Nodes: 148
  - Edges: 176
  - Cardinality: [2, 10]

Hypothesis: Similar networks can be constructed (by expert, machine learning, or combination) to detect, diagnose, predict, and mitigate in a broad range of systems.

Experiments, Simulated ADAPT Data

Inference Time (ms)	MPE		Marginals	
	VE	ACE	JTP	ACE
Minimum	19.30	0.2235	9.792	0.5721
Maximum	40.21	2.5411	65.34	5.9228
Median	19.81	0.2260	10.52	0.6006
Mean	20.13	0.2625	11.01	0.7854
St. Dev.	1.554	0.2028	4.101	0.6970

ACE is the approach used in ProDiagnose

Comparison between Arithmetic Circuit Evaluation (ACE), Variable Elimination (VE) and Clique Tree Propagation (CTP)

Main conclusions:

- –All three inference algorithms are quite efficient, thanks to auto-generation algorithm
- –ACE outperforms VE (for MPE) and CTP (for marginals), both in Mean and St. Dev.

Experiments, ADAPT Power System

Results summary (CUSUM enabled):

- DXC-10 training set
- Detection accuracy doubled
- False negative rate greatly improved
- Improvement in average detection time
- Average isolation time increased
  - The DX competition specifies that no isolation time be recorded for an incorrect mis-diagnosis.

	CUSUM	
Metric	**Enabled**	**Disabled**
Detection Accuracy	92.31%	46.15%
False Positives Rate	0%	0%
False Negatives Rate	8.82%	61.76%
Mean Time To Detect	17.97 s	28.36 s
Mean Time To Isolate	72.27 s	51.14 s

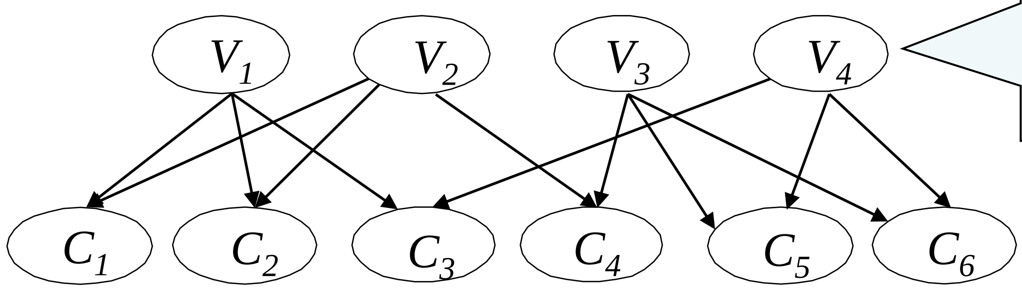
	ADAPT DXC Tier 1			ADAPT DXC Tier 2		
Metric	**ProADAPT**	**RODON**	**HyDE-S**	**ProADAPT**	**Stanford**	**RODON**
False positives (FP) rate	*0.0333*	0.0645	0.2000	*0.0732*	0.3256	0.5417
False negatives (FN) rate	*0.0313*	0.0968	0.0741	*0.1392*	0.0519	0.0972
Detection accuracy	*0.9677*	0.9194	0.8548	*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

9 competitors in Tier 1.

6 competitors in Tier 2.

Scalability of Bayesian Network Computation

Bipartite Bayesian Networks

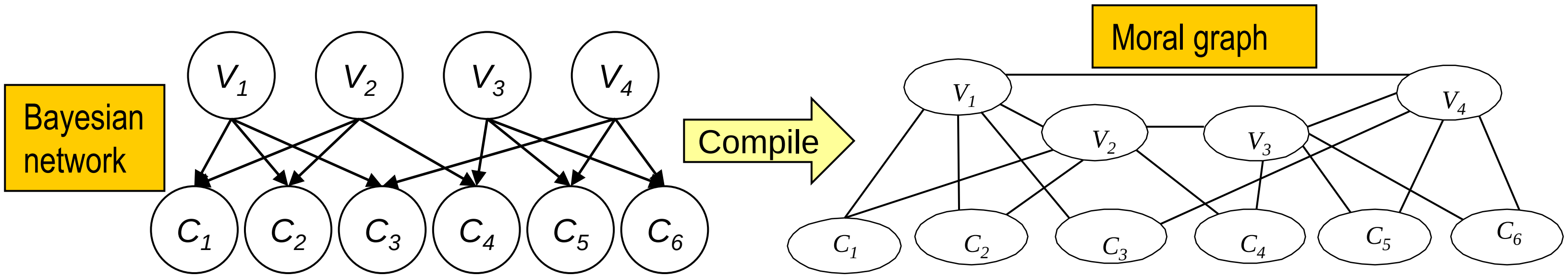


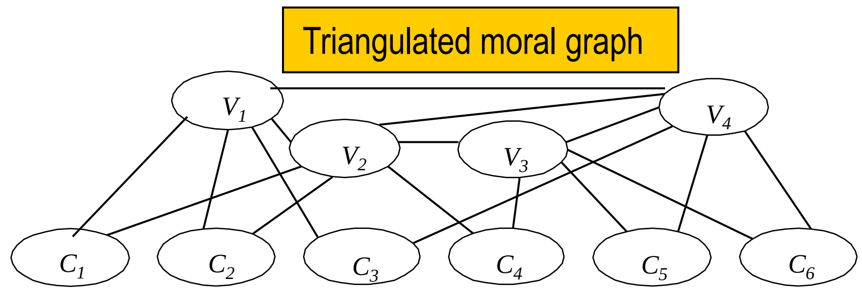
Root nodes are estimated (output nodes): represent whether components or systems are working or failing; type of user behavior (fraudulent or legitimate); etc.

Leaf nodes are observed (input nodes): represent detectors / sensors / observables / tests

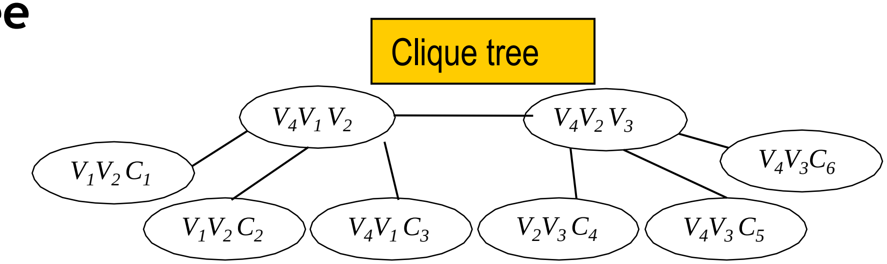
The number of sensors in mobile devices and infrastructure have increased dramatically. Are we taking full advantage of them, to understand the behavior of users as well as the communication and computation infrastructure?

Clique Tree Clustering



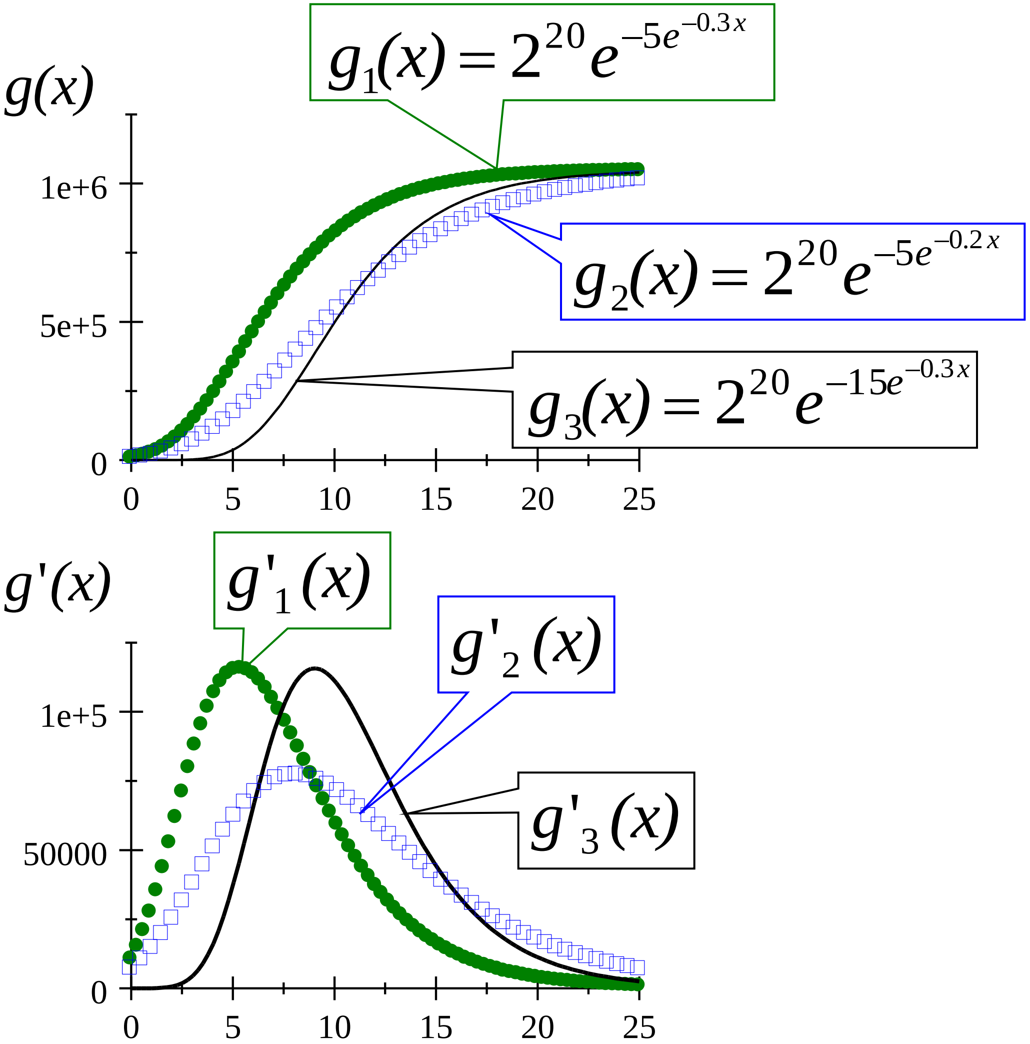


- **Tree clustering: a major approach to BN inference**
- **Tree clustering algorithms employ two phases:**
  - **Compilation: generate clique tree *β'''* from BN *β***
  - **Propagation: do belief revision (MPEs) or belief updating (marginals) by propagation of evidence in *β'''***



Details in [Lauritzen & Spiegelhalter, 88].

Gompertz Growth Curves



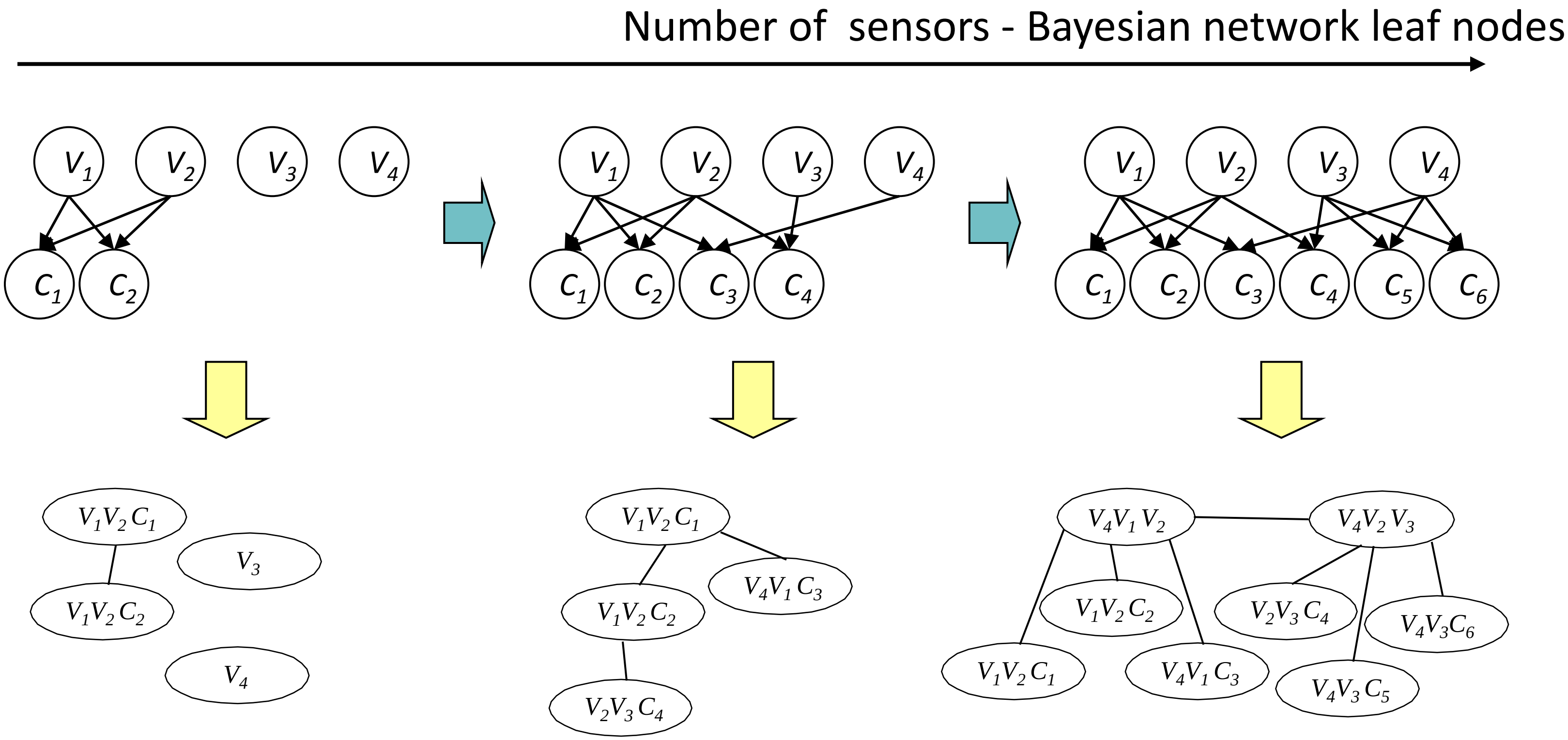
Gompertz growth curve:

$$g(x) = g(\infty) e^{-\zeta e^{-\gamma x}}$$

$g_1(x)$ to $g_3(x)$: Shift growth curve to right by increasing ζ from $\zeta = 5$ to $\zeta = 15$.

$g_1(x)$ to $g_2(x)$: Decrease maximal growth rate by decreasing γ from $\gamma = 0.3$ to $\gamma = 0.2$.

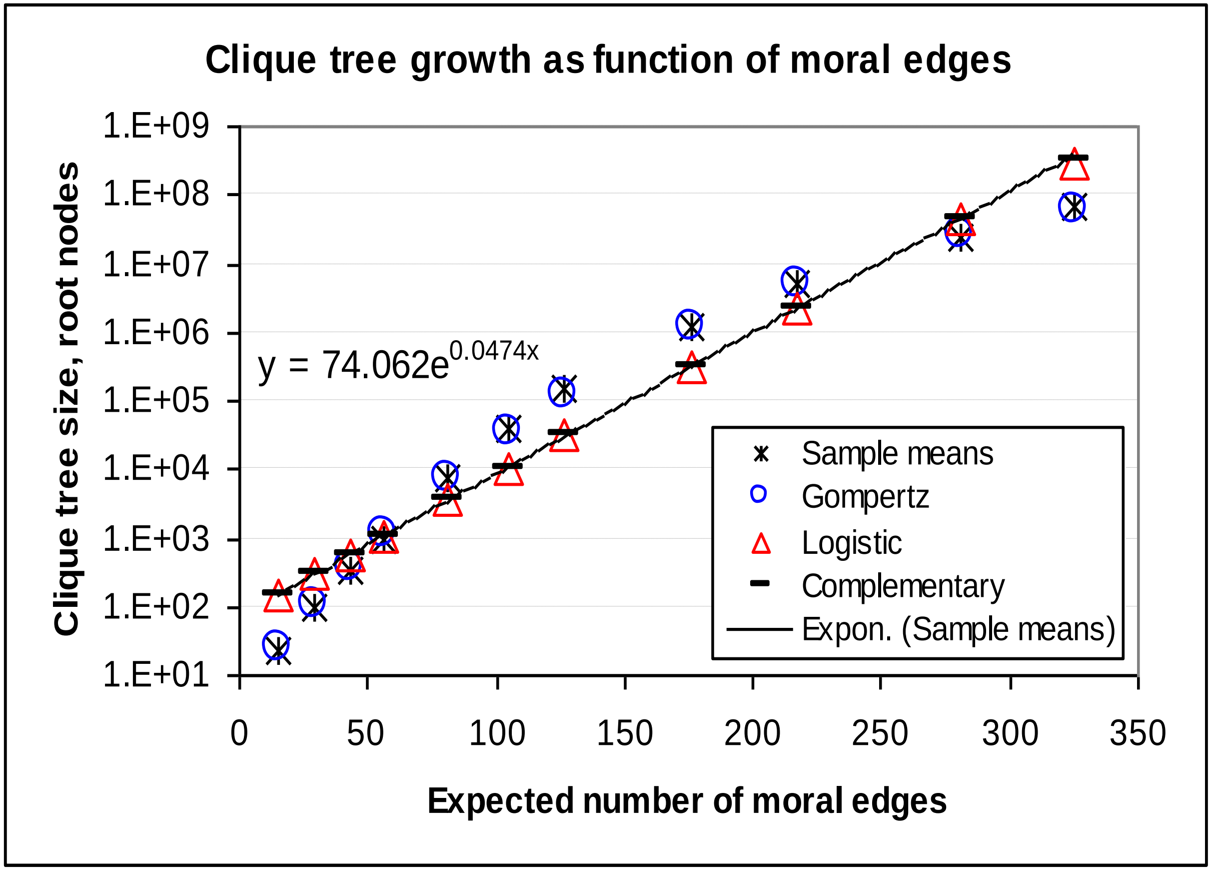
Growth of Bayesian Networks



Total Gompertz growth curve for BPART(*V*, *C*, *P*, *S*):

$$g_T(x) = S^V e^{-\zeta e^{-\gamma x}} + xS^{P+1}$$

Comparing Growth Curves



BNs of varying hardness generated with parameters *V=30, S=2, P=2*, and varying *C*.

Gompertz growth curve:

$$g(x) = 2^{30} \times \exp(-19.14 \times \exp(-0.005874x))$$

Current and Planned Work

Analytics:

- Improve Expectation Maximization (EM) algorithms for Bayesian network parameter estimation – exploit parallelism in modern hardware and software architectures
- Approach 1: Develop EM layer ”on top of” improved GPU-based approach to junction tree propagation
- Approach 2: Use MapReduce to explore data parallelism in Bayesian network parameter

Visualization:

- Improvements to current multi-focus, multi-view network visualizations
- Integration of novel and existing of analytics and visualization techniques

Experiments, demonstrations, and software:

- ADAPT datasets and Bayesian networks
- Synthetic Bayesian networks (“similar to ADAPT”) and other Bayesian networks
- Other network data sets: VAST challenge; disaster and emergency management, social network data, …
- Hardening and distribution of Java software

Publications

- A. Choi, L. Zheng, A. Darwiche, and O. J. Mengshoel. “A Tutorial on Bayesian Networks for System Health Management’,” In *Machine Learning and Knowledge Discovery for Engineering Systems Health Management,* Chapman & Hall/CRC Data Mining and Knowledge Discovery Series, A. N. Srivastava and J. Han (Editors) , 2011.
- B. Ricks, C. Harrison, O. J. Mengshoel. “Integrating Probabilistic Reasoning and Statistical Quality Control Techniques for Fault Diagnosis in Hybrid Domains,” In Proc. International Conference on Prognostics and Health Management (PHM-11) Montreal, September 27, 2011.
- O. J. Mengshoel, D. C. Wilkins, and D. Roth, “Initialization and Restart in Stochastic Local Search: Computing a Most Probable Explanation in Bayesian Networks”, IEEE Trans. on Knowledge and Data Engineering, 2011.
- L. Zheng, O. J. Mengshoel, and J. Chong. “Belief Propagation by Message Passing in Junction Trees: Computing Each Message Faster Using GPU Parallelization”. Proc. Uncertainty in Artificial Intelligence (UAI-11), Barcelona, Spain, July 2011.
- O. J. Mengshoel, "Understanding the Scalability of Bayesian Network Inference using Clique Tree Growth Curves“, *Artificial Intelligence*, 174 (2010) pp. 984–1006.
- B. W. Ricks and O. J. Mengshoel. “Diagnosing Intermittent and Persistent Faults Using Static Bayesian Networks”, In *Proc. 21st International Workshop on Principles of Diagnosis (DX-10)*, Portland, OR, October 2010.
- 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, Part A*, Vol. 40, No. 5, September 2010.
- B. W. Ricks and O. J. Mengshoel. “Methods for Probabilistic Fault Diagnosis: An Electrical Power System Case Study.” In *Proc. of Annual Conference of the PHM Society*, 2009.
- O.J. Mengshoel, S. Poll, and T. Kurtoglu, “Developing Large-Scale Bayesian Networks by Composition: Fault Diagnosis of Electrical Power Systems in Aircraft and Spacecraft,” In *Proc. of the IJCAI-09 Workshop on Self-* and Autonomous Systems (SAS): Reasoning and Integration Challenges*, 2009.
- B. W. Ricks and O. J. Mengshoel, “The Diagnostic Challenge Competition: Probabilistic Techniques for Fault Diagnosis in Electrical Power Systems.” In *Proc. of the 20th Intern. Workshop on Principles of Diagnosis (DX-09),* Stockholm, Sweden, 2009.
- O. J. Mengshoel, "Understanding the role of noise in stochastic local search: Analysis and experiments“, *Artificial Intelligence*, Volume 172, Issue 8-9, Pages 955-990, 2008.
- O. J. Mengshoel and D. E. Goldberg, “The Crowding Approach to Niching in Genetic Algorithms.” Evolutionary Computation, 16(3), 2008, pages 315-354.
- 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 *Proc. of the Twentieth Innovative Applications of AI Conference (IAAI-08)*, Chicago, IL, 2008.

Questions?