Visualization of Analytical Processes

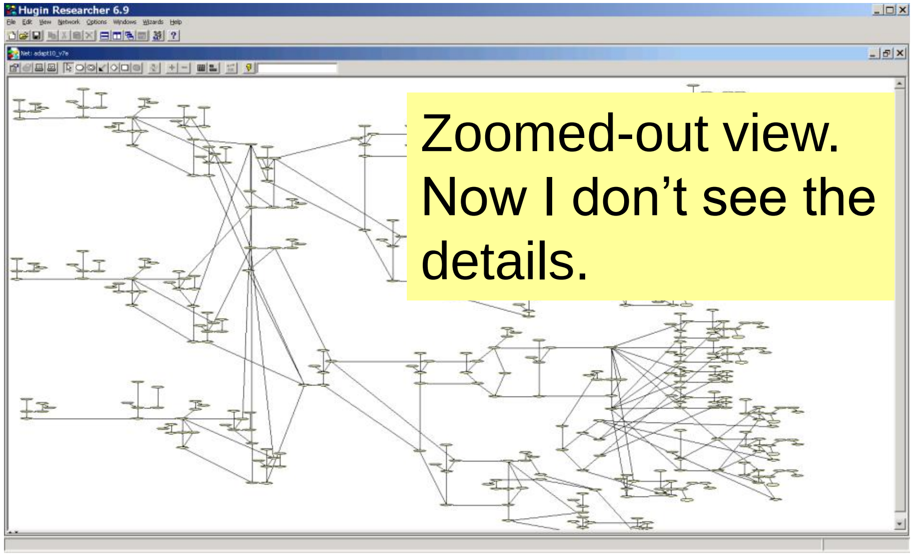
Ole J. Mengshoel, Ted Selker, and Marija Ilic

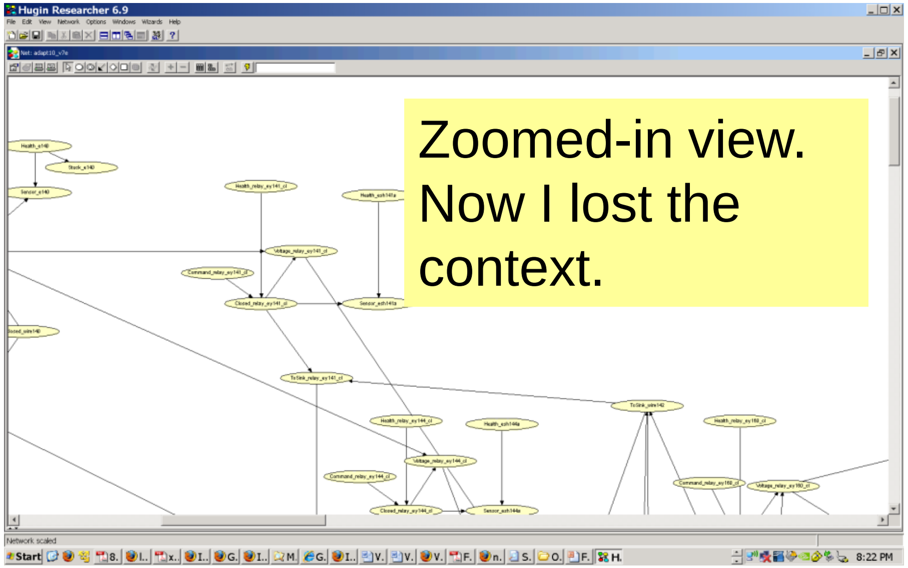
Carnegie Mellon University

NSF FODAVA Review, December 13th, 2012

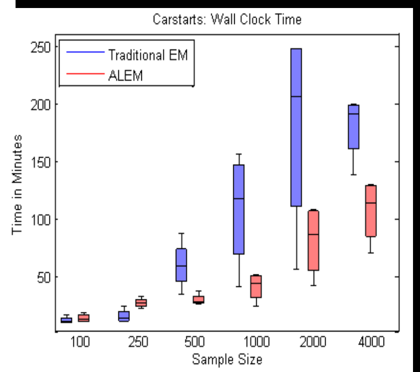
Project Background

- Focus on networks, including probabilistic graphical models (Bayesian networks) - and electrical power networks
- Bayesian network representations are natural and useful in computation, visualization, and human-computer interaction
- Difficulty for humans to reason under uncertainty, especially under time pressure and stress - benefit probabilistic graphical models
- Most current visualization and analytical methods for probabilistic reasoning target domains with “few” random variables, small data set, and non-interactive use

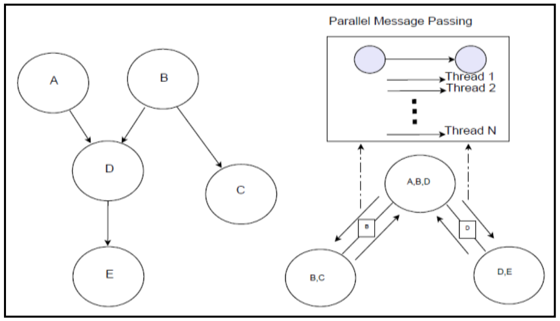




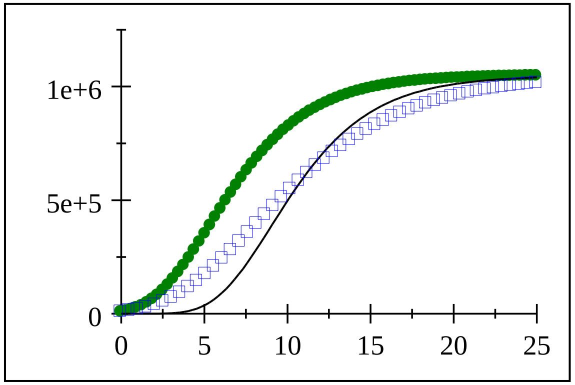
Project Accomplishments

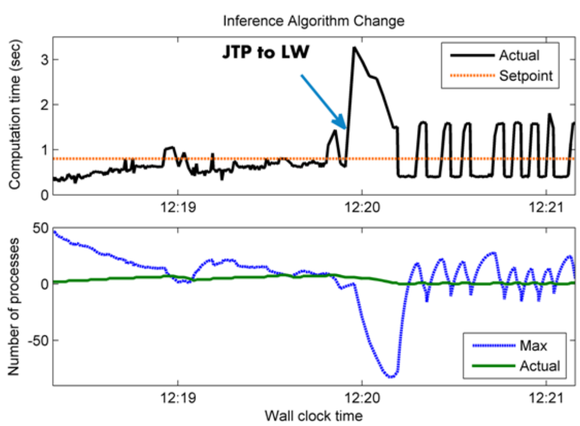


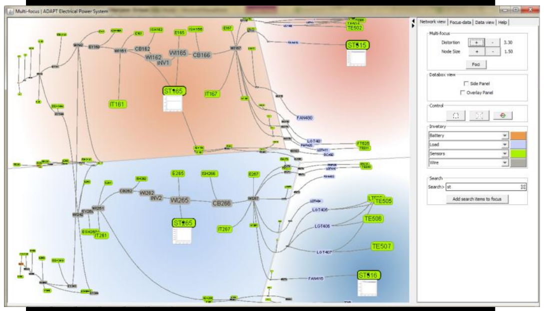
Fast Belief Propagation in Junction Trees: GPU Parallelization



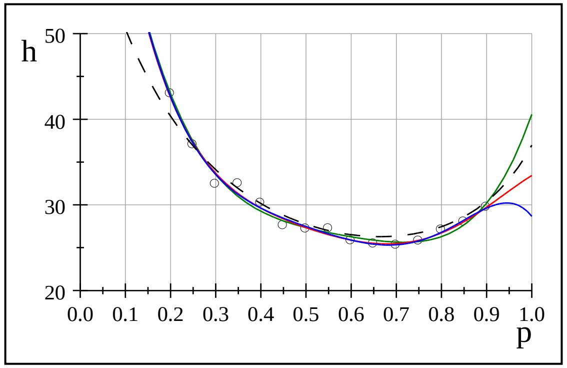
Understanding Scalability of Bayesian Network Computation using Junction Tree Growth Curves







Formalization of Stochastic Local Search

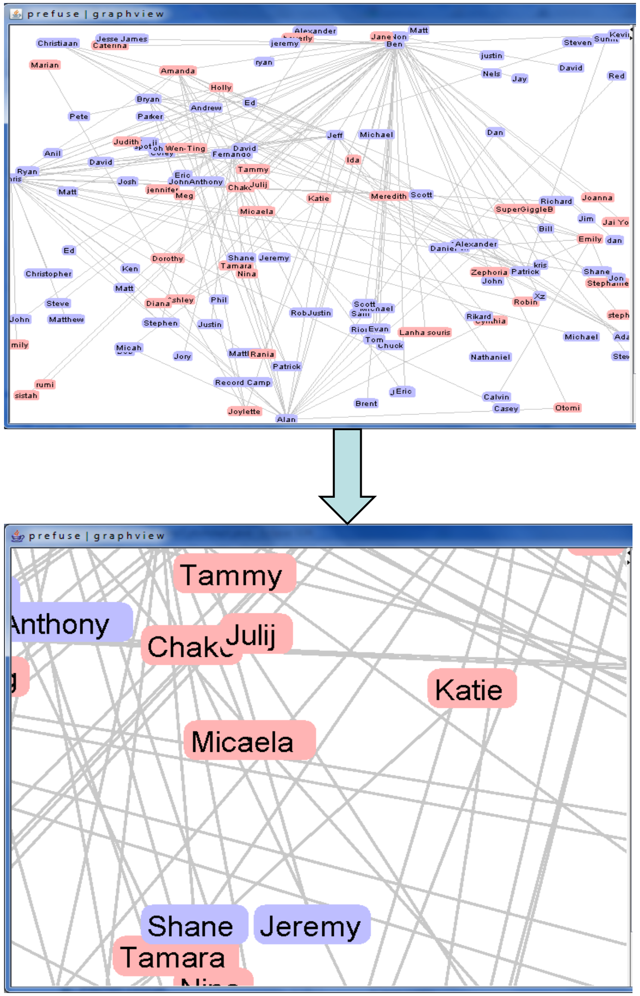


Multi-Fisheye, Multi-View for Interactive Visualization of Large Graphs

- P. K. Sundararajan, O. J. Mengshoel, and T. Selker. "Multi-Focus and Multi-Window Techniques for Interactive Network Exploration." In *Proc. of Visualization and Data Analysis (VDA 2013)*, San Francisco, Feb 2013.
- M. Cossalter, O. J. Mengshoel, and T. Selker. "Multi-Focus and Multi-Level Techniques for Visualization and Analysis of Networks with Thematic Data." In *Proc. of Visualization and Data Analysis (VDA 2013)*, San Francisco, Feb 2013.

Network Visualization Challenges

- Exploration of large graphs
- Context loss while zooming or scrolling
- Limitations of traditional fisheye:
  - Only one focus
  - Distorting whole layout
- Often no displaying of information (e.g., Bayesian network) along with the underlying data set (e.g., time series) or thematic data
- Need to perform visual search to locate thematic data or data set
  - Ex: Relationship between a node and its conditional probability table



Focus+Context: Multi-Fisheye and Multi-Window Visualization for Networks

OBJECTIVE

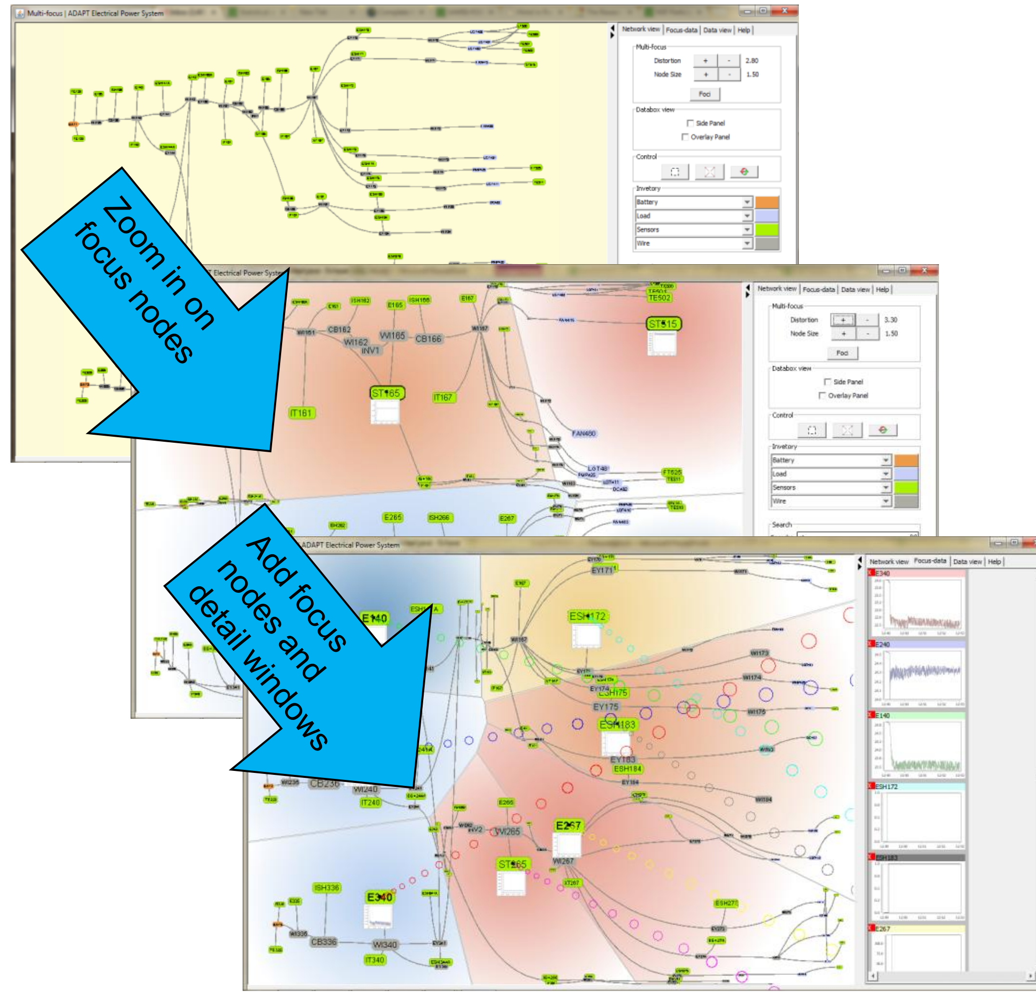
Improve the applicability of multi-fisheye to exploration of networks, including Bayesian network (BN) problem instances. Focus on large-scale but in-memory networks.

DESCRIPTION & FEATURES

A focus+context visualization tool that supports the interactive creation of multiple fisheyes (Bayesian network nodes, for example) with corresponding distortions. Voronoi edges separate the fisheyes, and multiple windows 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 networks. Node context, including network edge connection patterns and relative location, is preserved.



Multi-Focus, Multi-Window User Study

OBJECTIVE

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

DESCRIPTION

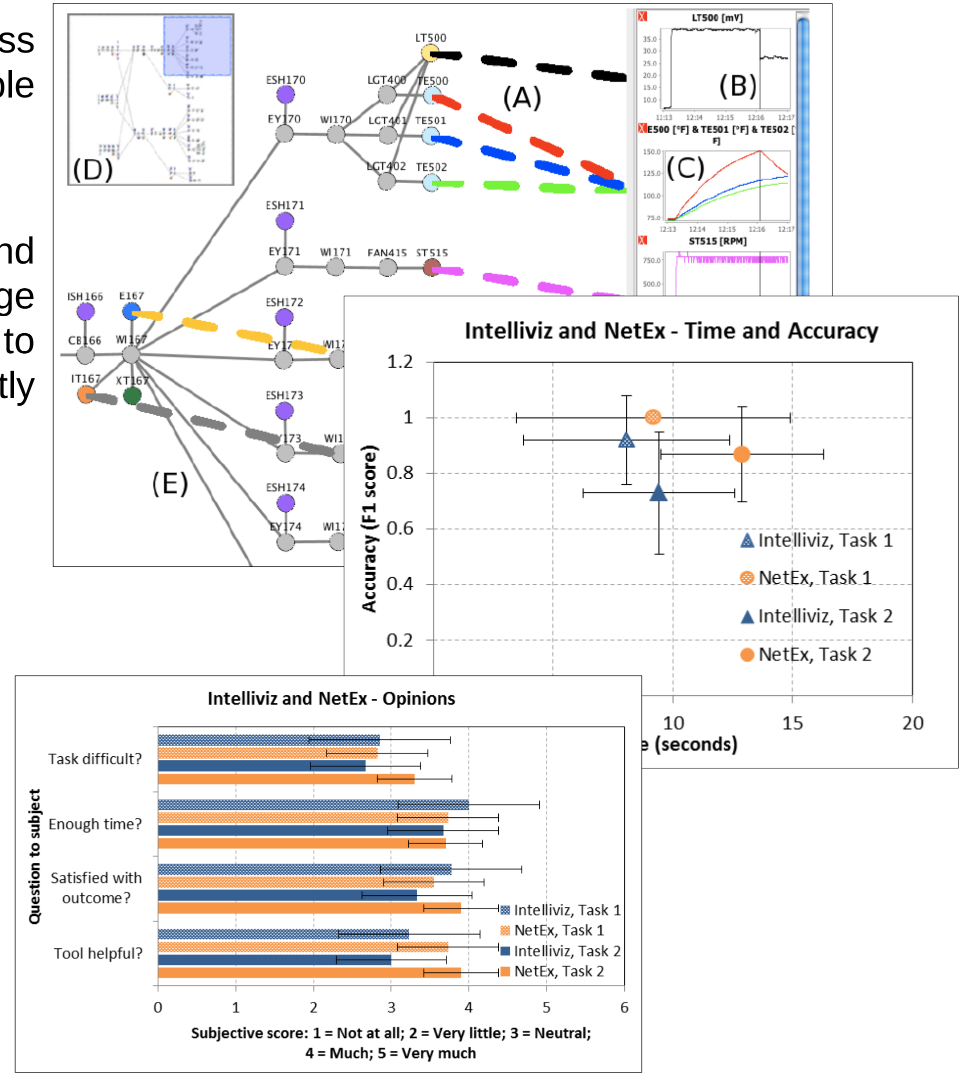
Two network visualization and analysis tool, **NetEx** and **NetEy,** that enrich the traditional node-edge visualization of networks by providing easy access to other aspects of data that can not be directly encapsulated in the graph structure.

FEATURES OF NetEx TOOL:

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

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



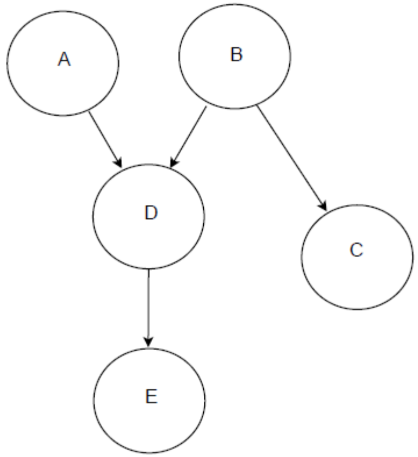
Carnegie Mellon University Silicon Valley

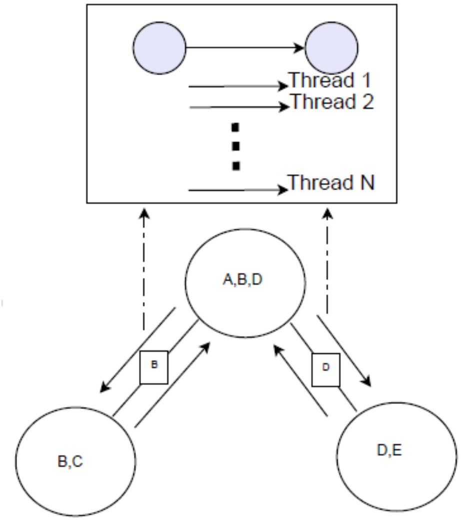
Speeding up Bayesian Network Computation using Parallel and Distributed Methods

- A. Basak, I. Brinster, X. Ma and O. J. Mengshoel. “Accelerating Bayesian Network Parameter Learning Using Hadoop and MapReduce.” In *Proc. 1st International Workshop on Big Data, Streams and Heterogeneous Source Mining @ KDD-12*, Beijing, China, August 2012.
- A. Saluja, P. Sundararajan, and O. J. Mengshoel. “Age-Layered Expectation Maximization for Parameter Learning in Bayesian Networks.” In *Proc. Artificial Intelligence & Statistics (AIStats)*, La Palma, Spain, April 2012.
- L. Zheng, O. J. Mengshoel, and J. Chong. “Belief Propagation by Message Passing in Junction Trees: Computing Each Message Faster Using GPU Parallelization.” In *Proc. Uncertainty in Artificial Intelligence (UAI-11)*, Barcelona, Spain, July 2011.

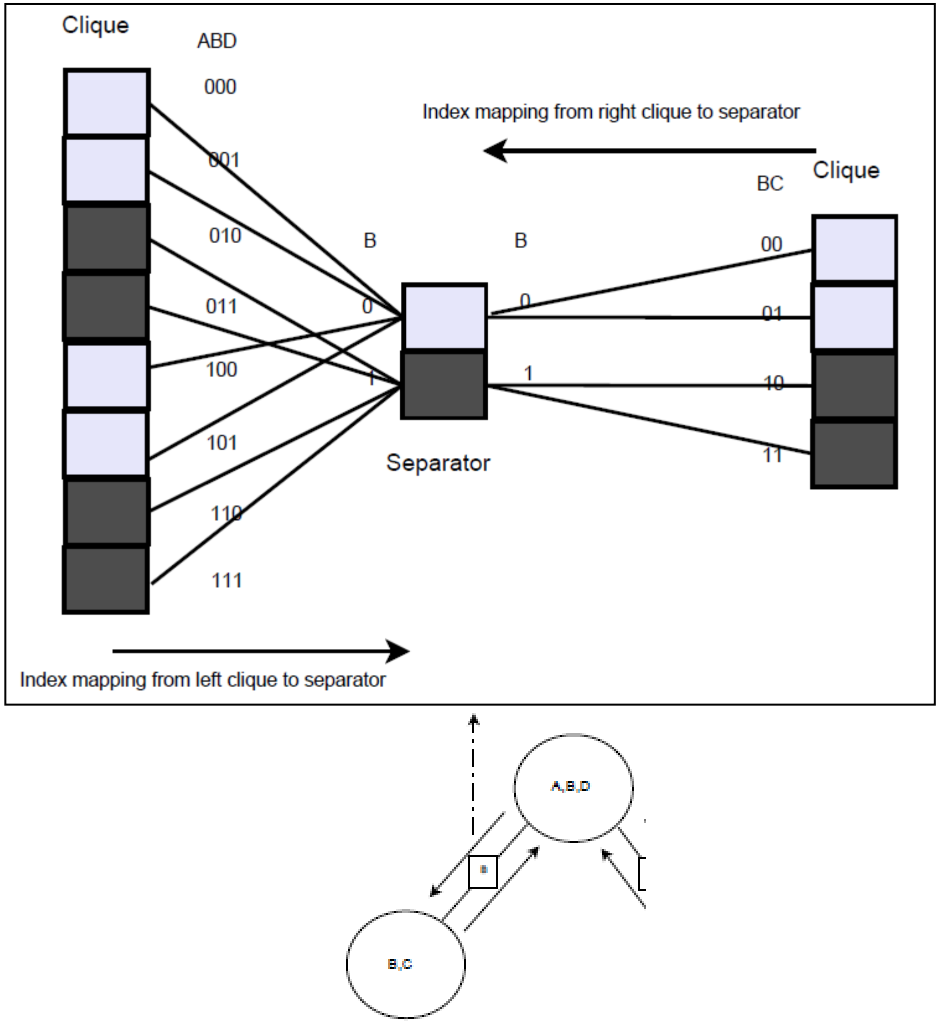
Motivation and Approach - GPU

- 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





Parallel Message Passing, Junction Trees



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

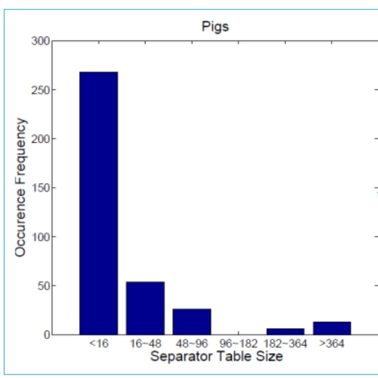
Speed up Bayesian network computation when junction trees are being used; exploit many-core parallelism such as graphics processing units (GPUs).

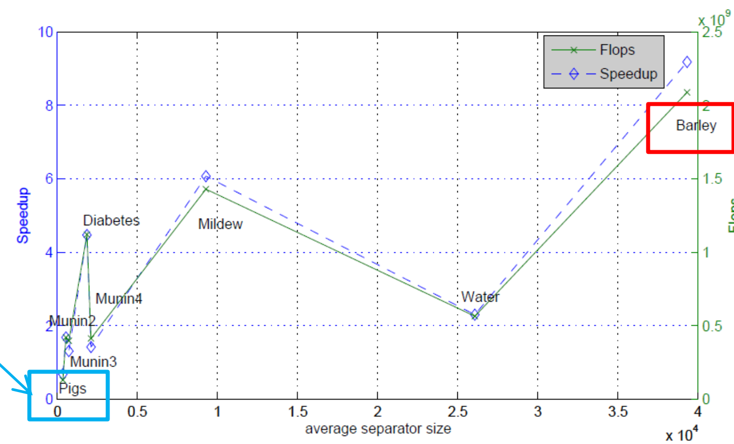
Message passing is performed in parallel, benefiting situations with large cliques and large separators

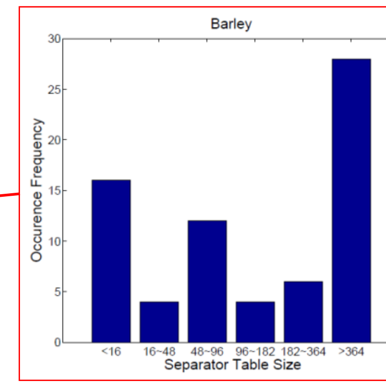
$$Speedup = \frac{\sum_i \sum_{k \in Ne(\mathcal{C}_i)} (|\phi_{\mathcal{X}_i}| + |\phi_{\mathcal{X}_k}|)}{2(n-1)\tau + \sum_i \sum_{k \in Ne(\mathcal{C}_i)} \frac{(|\phi_{\mathcal{X}_i}| + |\phi_{\mathcal{X}_k}|)}{|\phi_{\mathcal{S}_{ik}}|}}.$$

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





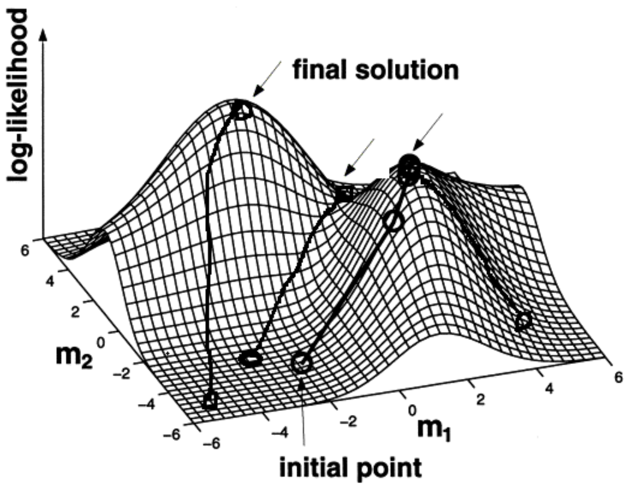


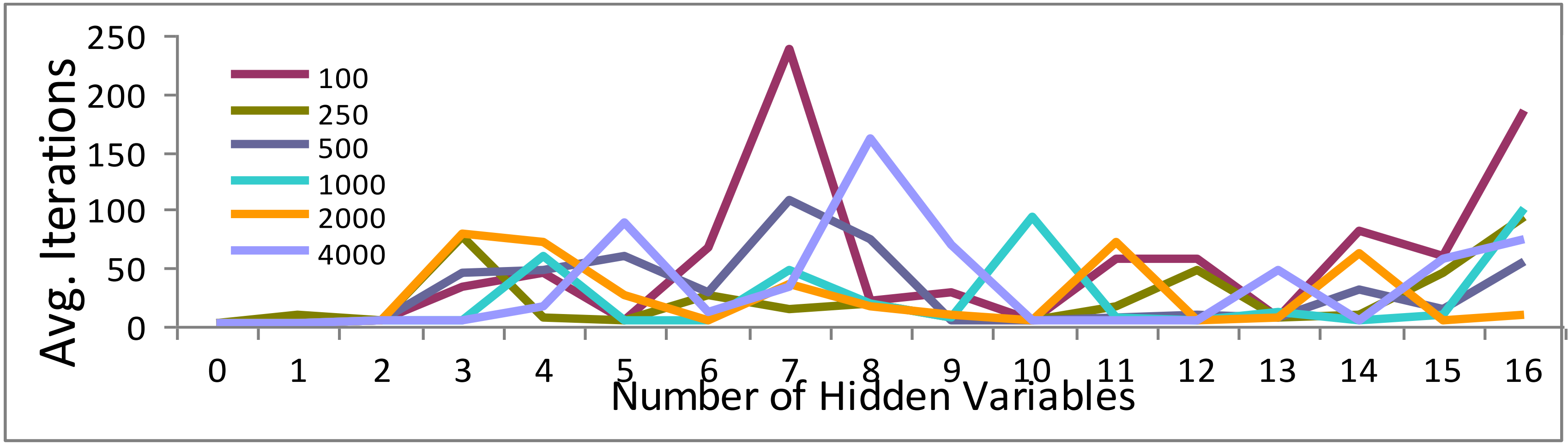
Best published experimental speed up result to date is 918%.

Problems of EM Algorithm

Time consuming:

- Computing E-step
- Problem of local maxima
  - Multiple random starting points
- Large number of iterations
- Big Data

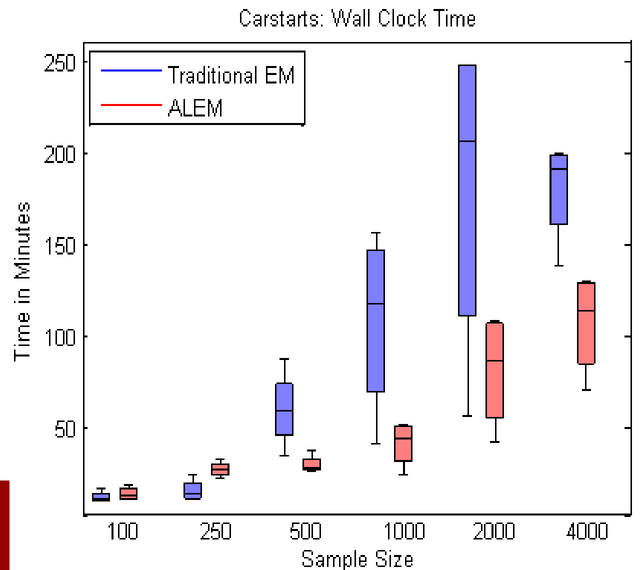


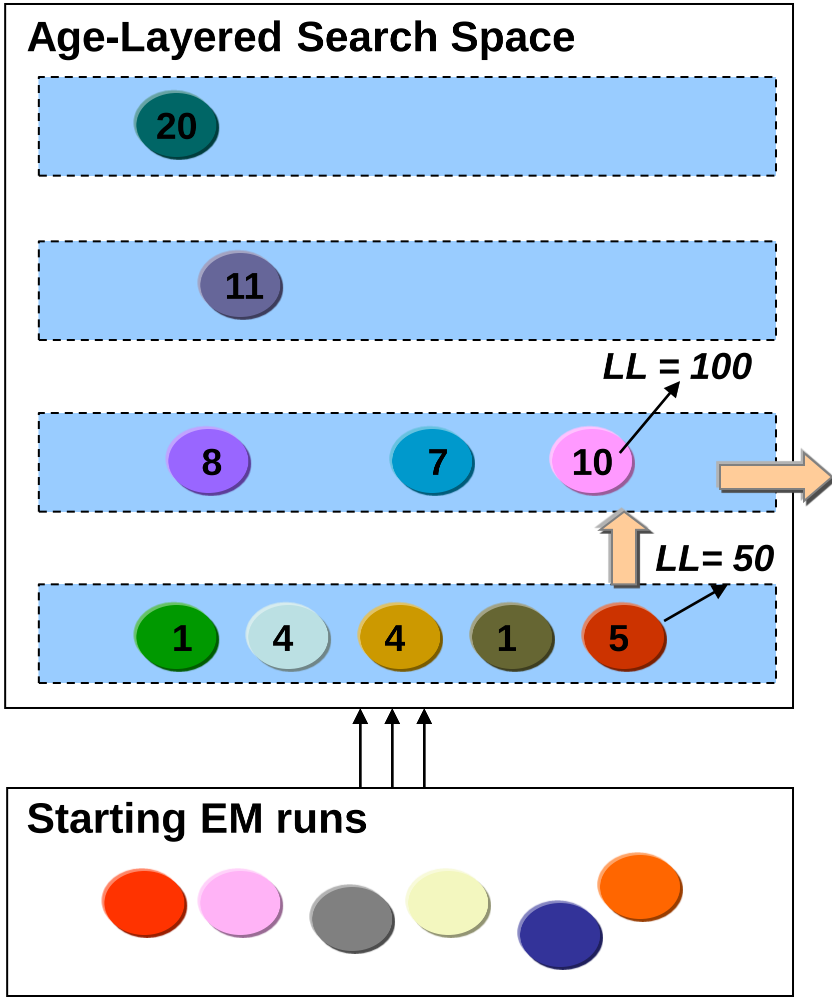


Age-Layered Expectation Maximization

Objective: Speed up convergence

1. EM runs are allotted ages (number of iterations).
2. Each layer has an age limit allotted to it.
3. When an EM run reaches the maximum age for that layer, it may need to competes (log-likelihood - LL) with runs in the next layer.





Bayesian Network Learning using MapReduce

Motivation:

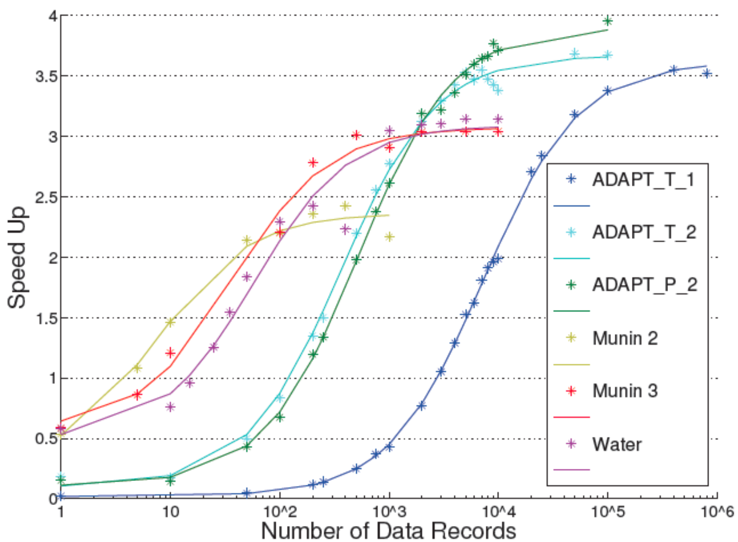
- Computational cost of parameter learning increases with network complexity and data set size.
- Additional bottlenecks for incomplete data:
  - Junction tree inference (E-step)
  - Iteration
  - Local optima

Approach:

- Formulate BN machine learning algorithms within MapReduce (MR)
- Implement algorithms using Hadoop

MR BN Learning: Incomplete Data

Curve fits, of the form (An + B)/(Cn + D):



Time to process one data record in M-step

Time to process one data record in E-step

Number of BN parameters

$$Speed-Up \approx \frac{t_{e-step}\, n + t_{m-step}\, |\theta|}{t_{start-up}\, n + t_{e-step}\left\lceil \frac{n}{M} \right\rceil + t_{aggreg}\, |\theta|} = \frac{An+B}{Cn+D}$$

Time to start up the cluster/distribute data, etc.

Time to update BN parameters in HDFS

Number of mappers

Experimental Bayesian networks

Bayesian Network (BN)	Nodes N	Edges E	Number of parameters $\|\theta\|$	Junction Tree (JT) Size	Cross-over points ($\Psi = 1$)
ADAPT_T1	120	136	1,504	1,690	2,800
ADAPT_P2	493	602	10,913	32,805	160
ADAPT_T2	671	789	13,281	36,396	130
Water	32	66	13,484	3,465,948	20
Munin3	1,044	1,315	85,855	3,113,174	10
Munin2	1,003	1,244	83,920	4,861,824	5

Integration of Bayesian Network Computation and Feedback Control

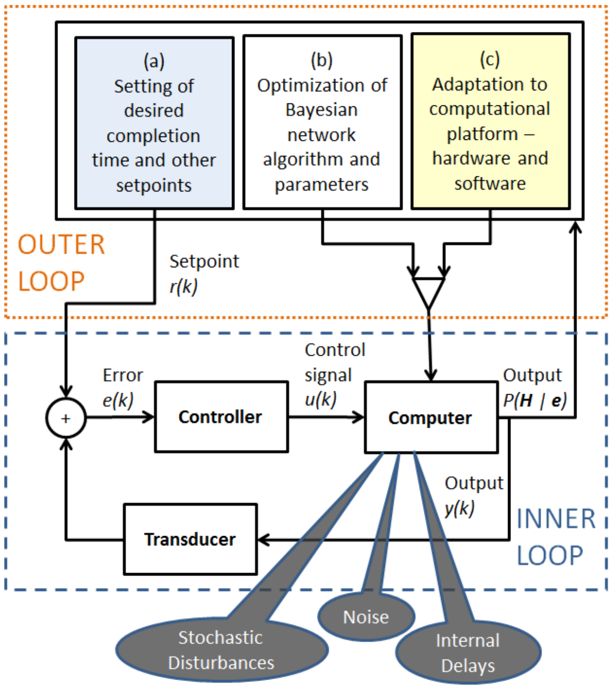
- E. Reed, A. Ishihara, and O. J. Mengshoel. "Adaptive Control Of Bayesian Network Computation." In *Proc. of the International Symposium of Resilient Control Systems (ISRCS-12)*, Salt Lake City, Utah, August 2012.
- O. J. Mengshoel, A. Ishihara, and E. Reed. "Reactive Bayesian Network Computation using Feedback Control: An Empirical Study." In *Proc. of Bayesian Modelling Applications Workshop (BMAW-12) @ UAI-12*, Catalina, CA, August 2012.

Motivation and Approach

- The power of "analytics technologies" is steadily improving
- Many BN inference algorithms, with varying resource requirements and performance:
    - likelihood weighting (LW)
    - junction tree propagation (JTP)
    - loopy belief propagation (LBP)
    - Pearl's belief propagation (PBP)
    - variable elimination (VE)
- BN inference in unpredictable SW and HW environments
- User behavior varies substantially
- Need for reactive computation
    - "soft real-time"

Approach: "middleware" supporting a range of reactive applications – based on feedback control

Feedback Control Architecture



Types of processes:

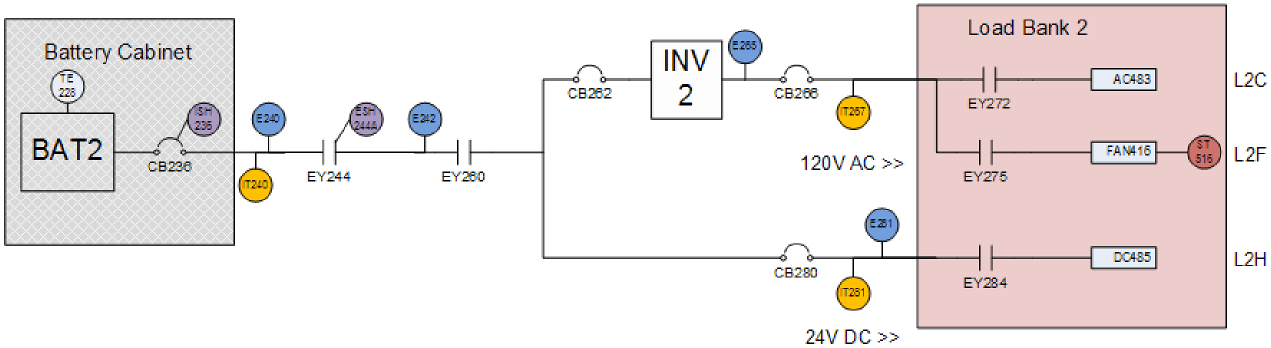
- High-criticality: BN diagnosis
- Medium-criticality: Undisturbed
- Low-criticality: Subject to control

Low-criticality processes are controlled – currently terminated – when BN computation time *y(k)* exceeds a set point *r(k)*.

A Linear Controller is given by the following z-transfer function:

$$H(z) = \frac{K}{1 - \alpha z^{-1}}$$

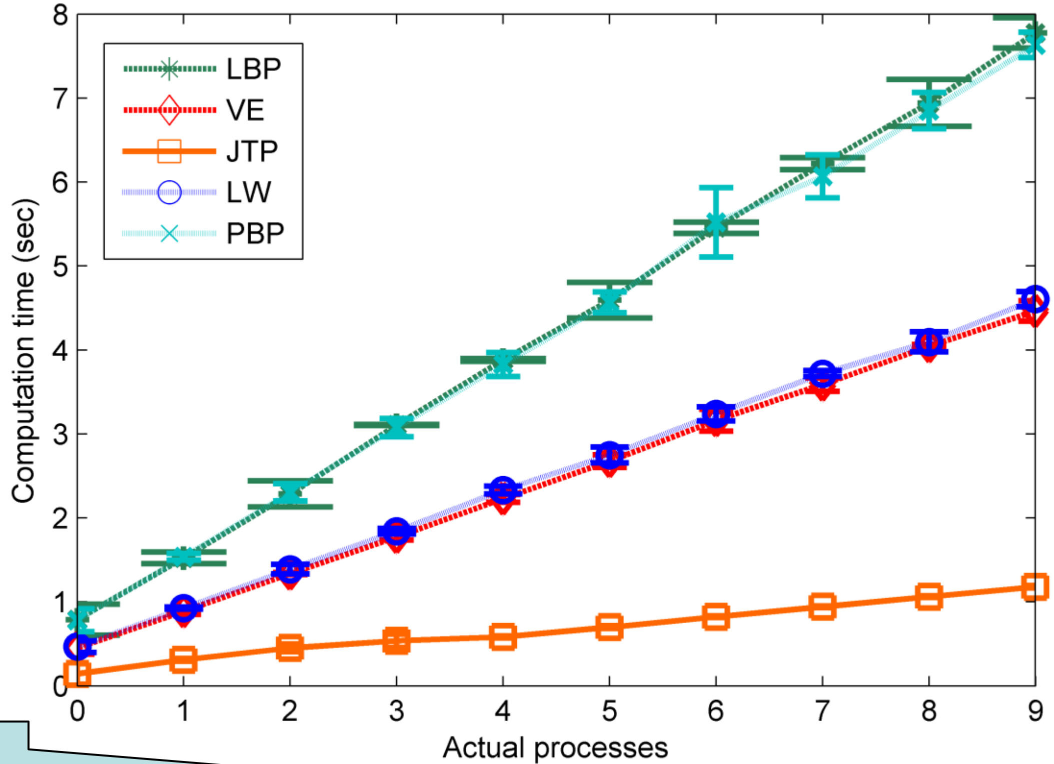
Electrical Power System



- We used data from ADAPT (above), an electrical power system testbed, and a BN model of ADAPT
- During each simulation timestep, the posterior P(***H*** | ***e***) over ADAPT's health nodes ***H*** is calculated, based on evidence ***e***
- We varied:
  - BN inference algorithms: **JTP**, **LW**, LBP, PBP, VE
  - Computer and OS: See paper
- Low criticality processes: generated using Poisson process
  - CPU-intensive, executing math operations in tight loop

Actual Computation Time as a Function of Actual Number of Processes

Actual computation time, high-criticality (BN) process



Actual number of low-criticality processes

Estimation of Model Parameters

Open loop model fitting with first order least squares using this model:

$$\hat{y}(t) = c_1 \hat{y}(t-1) + c_2 u(t-1) + c_3$$

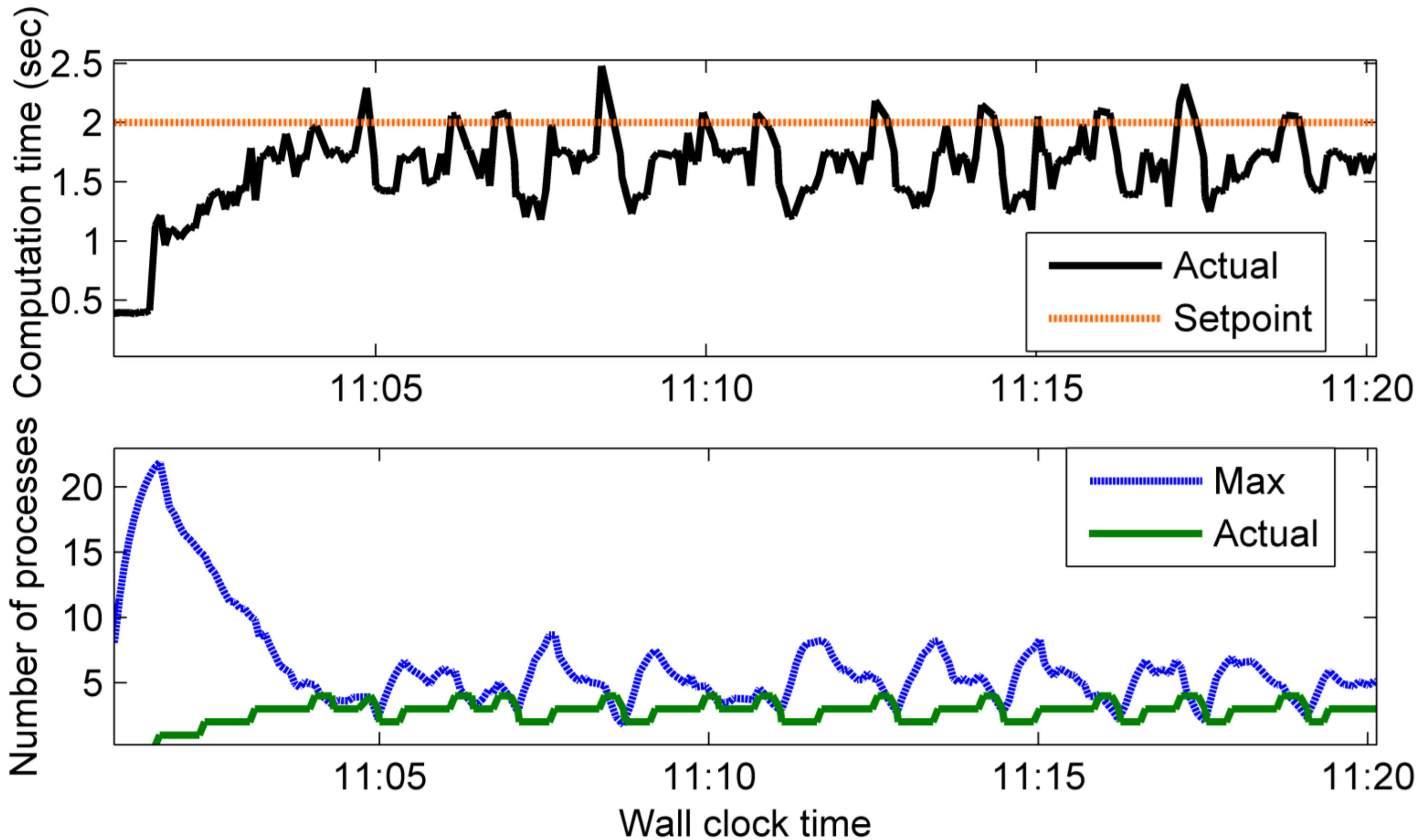
where:

- Parameters C_1, C_2, C_3 are estimated
- *u(t)* is the max number of low-criticality processes
- A random square-wave is used for *u(t)* to ensure sufficient excitation for parameter estimation

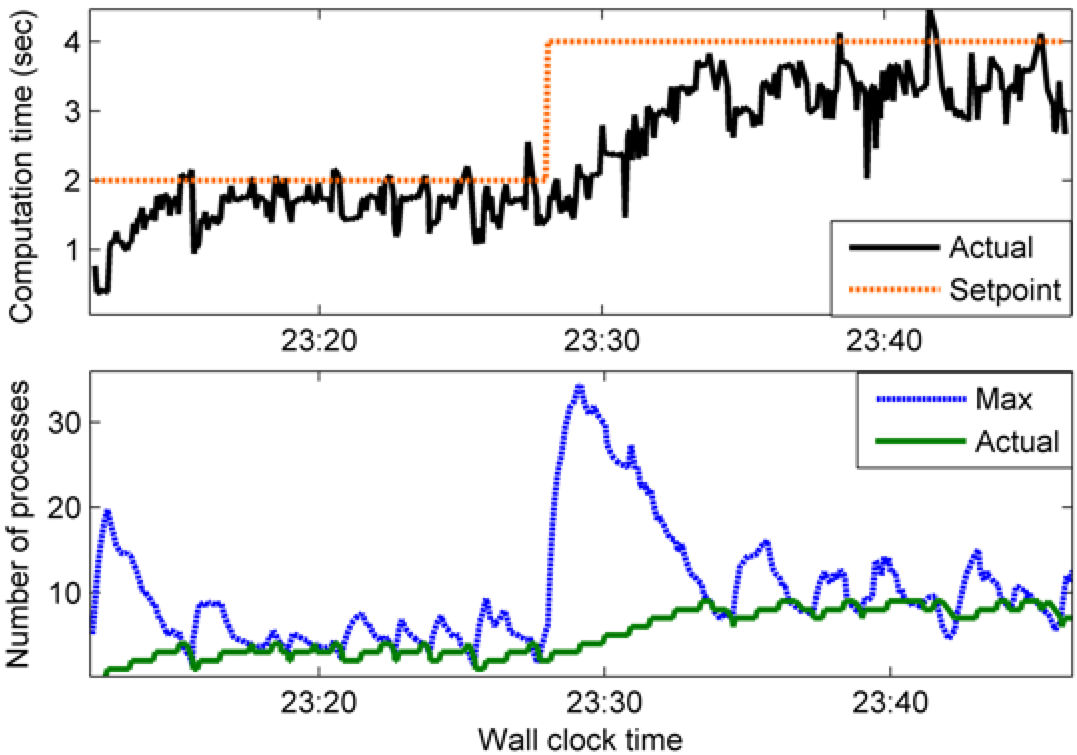
Experiments - Summary

		Algorithm	
		Fix	**Change**
Set-point	**Fix**	**(A)**	**(C)**
	Change	**(B)**	**(D)**

(A) Control: Fixed Setpoint, Fixed BN Inference Algorithm

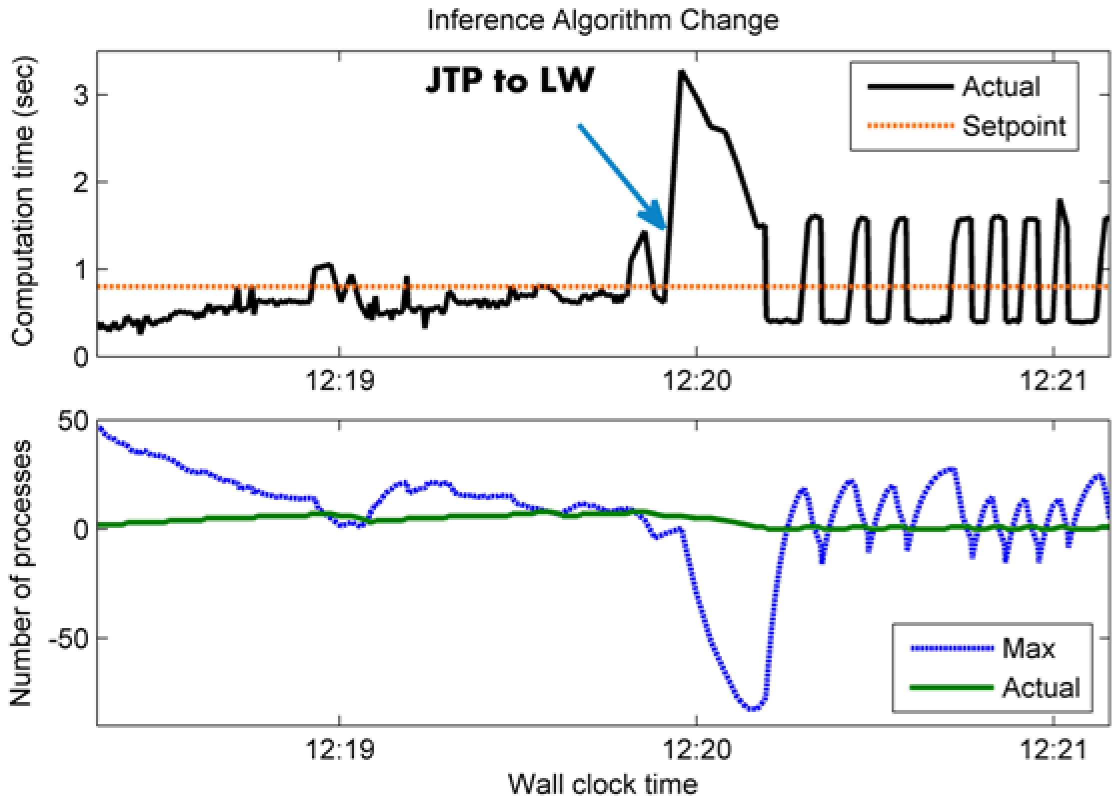


(B) Control: Change of Setpoint



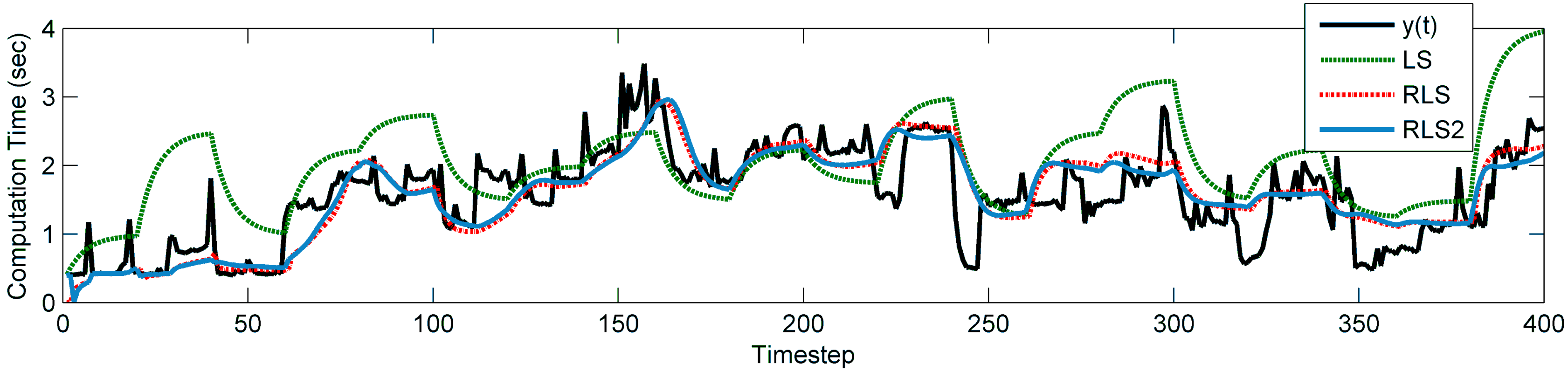
- Setpoint is changed midway through a simulation
- The controller reacts and increases the maximum number of low-criticality processes.

(C) Control: Change of BN Inference Algorithm



- Controller reacts to an inference algorithm change from junction tree propagation (JTP) to likelihood weighting (LW) midway through a simulation
- This results in a change of maximum number of low-criticality processes
- Tracking not as good after change

Adaptive Control: Online System Identification

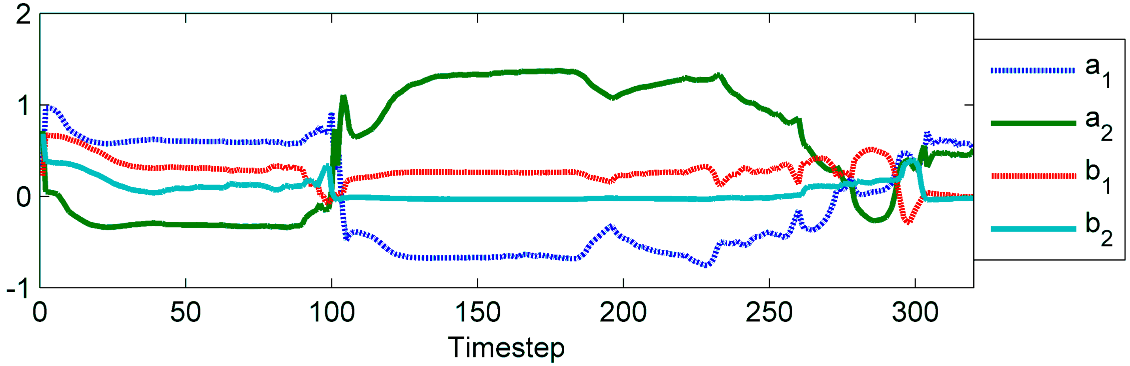


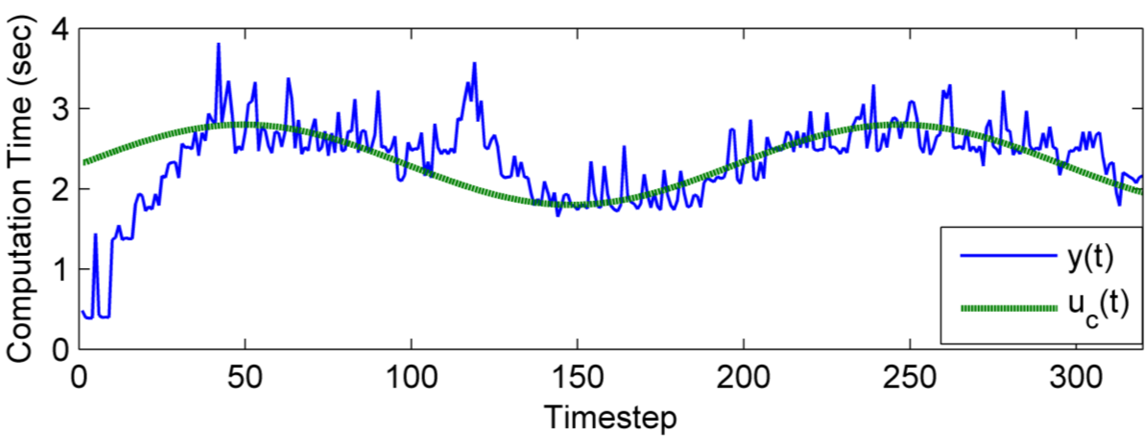
Realtime system identification with linear least squares (LS), recursive least squares (RLS), and 2nd order recursive least squares (RLS2) using the model:

$$\hat{y}(t) = a_1\hat{y}(t-1) + a_2\hat{y}(t-2) + b_1 u(t-1) + b_2 u(t-2)$$

Adaptive Control: Results

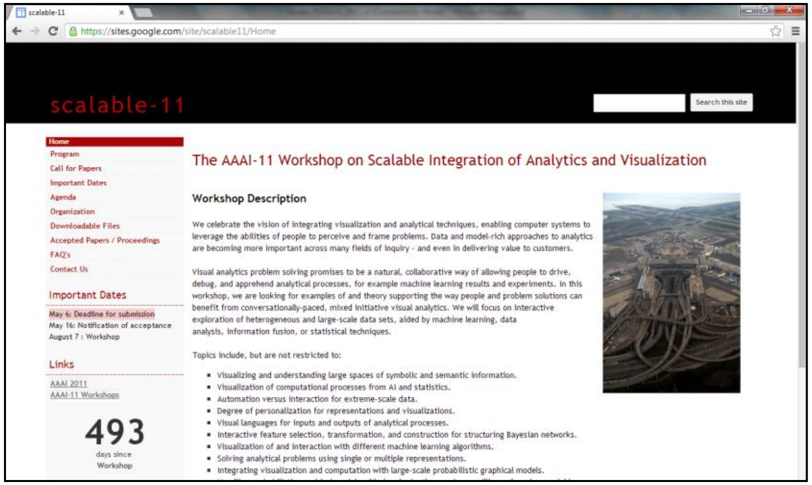
$$\hat{y}(t)=a_1\hat{y}(t-1)+a_2\hat{y}(t-2)+$$
$$b_1u(t-1)+b_2u(t-2)$$





- Adaptive parameters with minimum degree pole placement is used [Reed et al., 2012]
- Here $u_c(t)$ is the setpoint. The 2nd order RLS model was used.

Extensions and Outreach



- Organizer/Tutorials:
  - VisWeek BoF: Scalable Interactive Visualizations for Visual Analytics
  - AAAI Workshop: Scalable Integration of Analytics and Visualization
  - GTC Tutorial: Speeding up Computation in Probabilistic Graphical Models using GPGPUs
- Participation: AAAI, AISTATS, CHI, KDD, NIPS, PHM, UAI, VAC meeting, VDA, VisWeek/VAST, …
- Competitions: VAST dataset, DX dataset (ADAPT)
- New Graduate Level Class:
  - Visual Analytics, Summer 2011 & Summer 2012
- Software: NetEx, NetEy, …
- Follow-on Projects: ARPA-E, Industry

Selected References

– [Sundararajan et al., 2013] P. K. Sundararajan, O. J. Mengshoel, and T. Selker. "Multi-Focus and Multi-Window Techniques for Interactive Network Exploration." In Proc. of Visualization and Data Analysis (VDA 2013), San Francisco, Feb 2013.

– [Cossalter et al., 2013] M. Cossalter, O. J. Mengshoel, and T. Selker. "Multi-Focus and Multi-Level Techniques for Visualization and Analysis of Networks with Thematic Data." In Proc. of Visualization and Data Analysis (VDA 2013), San Francisco, Feb 2013.

– [Reed et al., 2012] E. Reed, A. Ishihara, and O. J. Mengshoel. "Adaptive Control of Bayesian Network Computation," in Proc. of 5th International Symposium on Resilient Control Systems, August 2012.

– [Ricks et al., 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.

– [Mengshoel et al., 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.

– [Mengshoel, 2010] O. J. Mengshoel. Understanding the scalability of Bayesian network inference using clique tree growth curves. *Articial Intelligence*, 174:984–1006, 2010.

– [Basak et al., 2012] A. Basak, I. Brinster, X. Ma and O. J. Mengshoel. "Accelerating Bayesian Network Parameter Learning Using Hadoop and MapReduce." In *Proc. 1st International Workshop on Big Data, Streams and Heterogeneous Source Mining @ KDD-12*, Beijing, China, August 2012.

– [Saluja et al., 2012] A. Saluja, P. Sundararajan, and O. J. Mengshoel. "Age-Layered Expectation Maximization for Parameter Learning in Bayesian Networks." In *Proc. Artificial Intelligence & Statistics (AIStats)*, La Palma, Spain, April 2012.

– [Zheng et al., 20121 L. Zheng, O. J. Mengshoel, and J. Chong. "Belief Propagation by Message Passing in Junction Trees: Computing Each Message Faster Using GPU Parallelization." In *Proc. Uncertainty in Artificial Intelligence (UAI-11)*, Barcelona, Spain, July 2011.