Graph-guided Sampling

Rina Dechter

Bren school of Information and Computer Sciences

University of California, Irvine

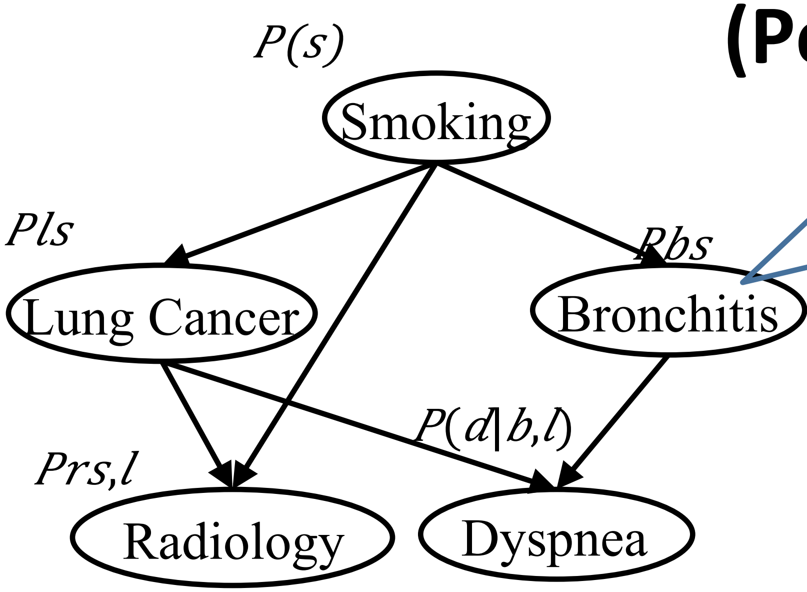
  

Joint work with Vibhav Gogate and Bozhena Bidyuk

Outline

- Background: Bayesian networks
- Importance Sampling
- AND/OR search space for grpahical Models
- AND/OR Sampling: exploiting structure
- AND/OR+cutset sampling
- UAI 2010 experience

Directed graphical models: Bayesian networks (Pearl 1988)



B	S	P(b|s)
b^0	s^0	0.1
b^1	s^0	0.9
b^0	s^1	0.7
b^1	s^1	0.3

Variables: $\{X_1, \ldots, X_n\}$

$$P(x) = \prod_i P(x_i \mid pa(x_i))$$

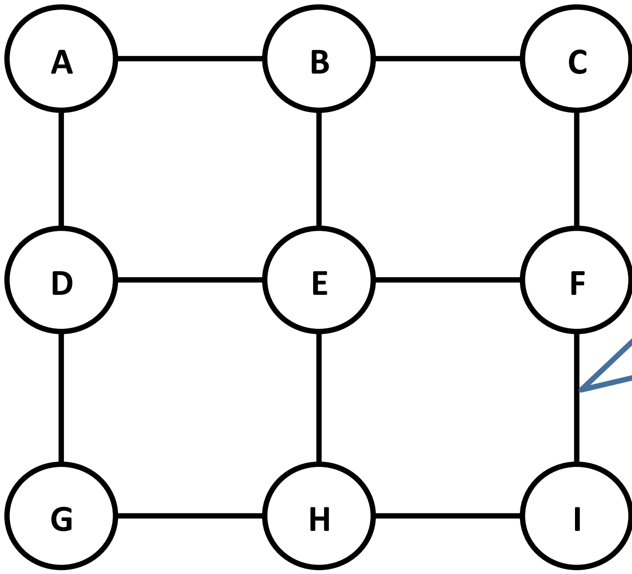
$$P(s,l,b,r,d) = P(s) P(l \mid s) P(b \mid s) P(r \mid s,l) P(d \mid b,l)$$

- Basic inference task: compute the probability of evidence (likelihood computation)

$$P(E=e) = \sum_z P(Z=z, E=e) = \sum_z \prod_i [P(z_i \mid pa(z_i)) \mid E=e]$$

$$P(S=s^0, D=d^1) = \sum_{l,b,r} P(s^0) P(l \mid s^0) P(b \mid s^0) P(d^1 \mid b,l) P(r \mid s^0, l)$$

Undirected graphical models: Markov networks



F	I	Value
f^0	i^0	23.01
f^1	i^0	42.9
f^0	i^1	34.7
f^1	i^1	0.002

Node potentials: $\phi_i(x)$
Edge potentials: $\phi_{i,j}(x)$

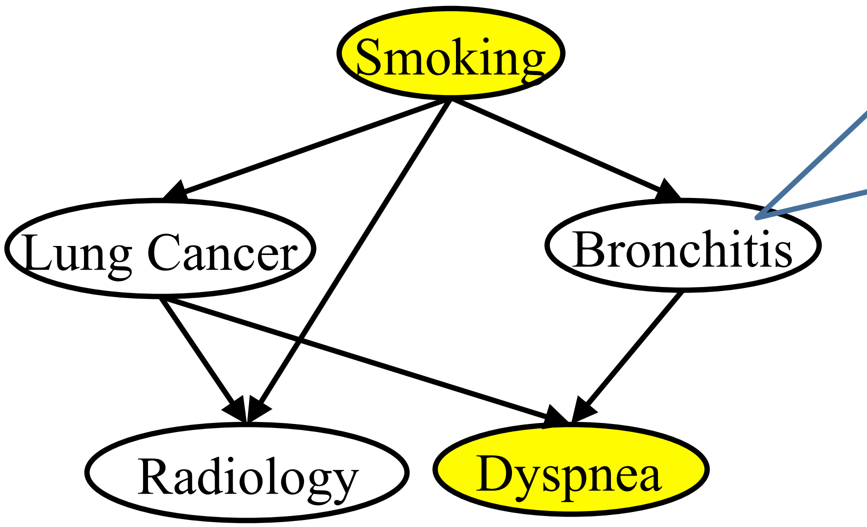
$$P(x) = 1/Z \prod_i \phi_i(x) \prod_{i,j} \phi_{i,j}(x)$$

$$Z = \sum_x \prod_i \phi_i(x) \prod_{i,j} \phi_{i,j}(x)$$

Z: Partition function

- Inference task: compute the partition function Z
  - Sum-product problem
  - Qualitatively same as computing P(e) in Bayesian networks
    - Counting solutions of a SAT or a CSP

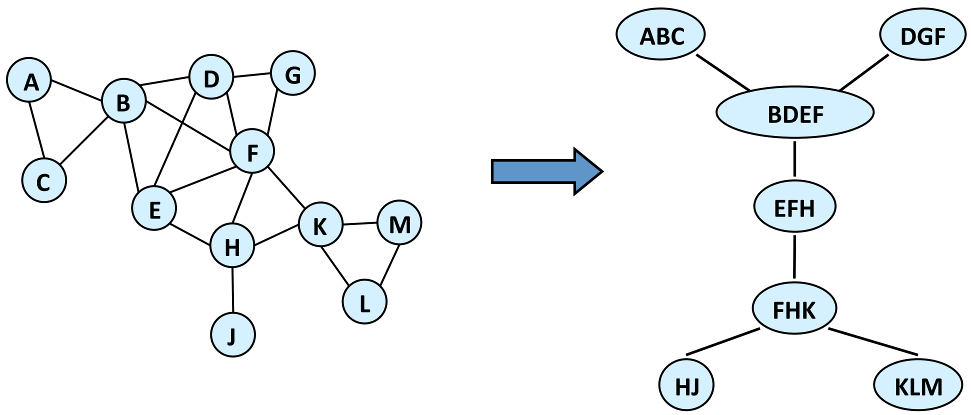
Other inference tasks



B	S	P(B|S)
b^0	s^0	0.1
b^1	s^0	0.9
b^0	s^1	0.7
b^1	s^1	0.3

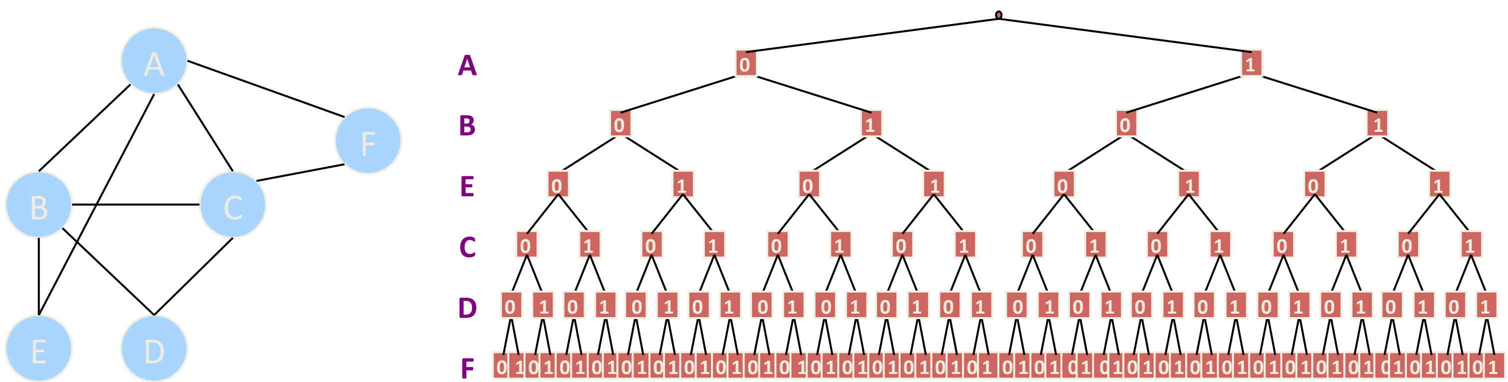
- Compute the conditional marginal probability given evidence (MAR task)
  – $P(L=l0|S=s0,D=d1)$
- Compute the most probable explanation (MPE)
  – Find a maximum probability assignment
  – Abductive inference

Inference vs conditioning-search



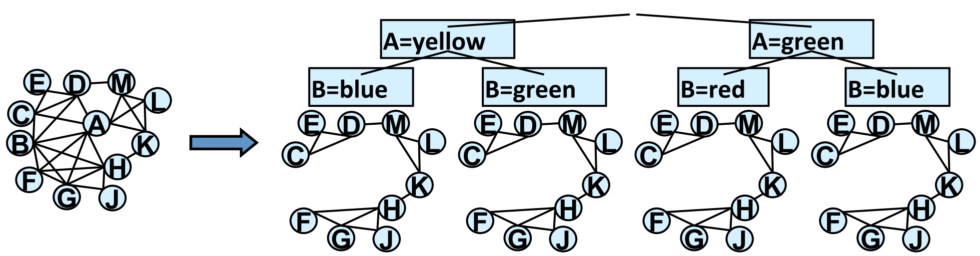
Inference

Exp(w*) time/space



Search

Exp(n) time
O(n) space



Search+inference:
Space: exp(w)
Time: exp(w+c(w))

W: user controlled

Approximation

- Since inference, search and hybrids are too expensive when graph is dense; (high treewidth) then:
- Bounding inference:
  - mini-bucket and mini-clustering
  - Belief propagation
- **Bounding search:**
  - **Sampling**
- Goal: an anytime scheme

Outline

- Background: Bayesian networks
- Importance Sampling
- AND/OR search space for grpahical Models
- AND/OR Sampling: exploiting structure
- AND/OR+cutset sampling
- UAI 2010 experience

Monte Carlo Estimate

- **Estimator:**
  - An estimator is a function of the samples.
  - It produces an estimate of the unknown parameter of the sampling *distribution.*

Given i.i.d. samples $S^1, S^2, \ldots S^T$ drawn from P,

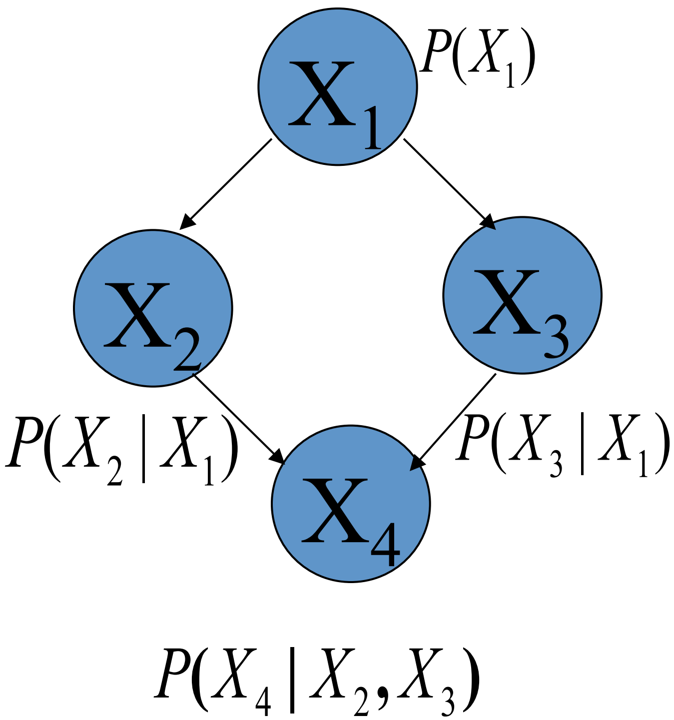
the Monte carlo estimate of $E_P[g(x)]$ is given by :

$$\hat{g} = \frac{1}{T}\sum_{t=1}^{T} g(S^t)$$

So the key is to generate samples from P

Logic sampling (example)

$$P(X_1, X_2, X_3, X_4) = P(X_1) \times P(X_2 \mid X_1) \times P(X_3 \mid X_1) \times P(X_4 \mid X_2, X_3)$$



No Evidence

// generate sample k

1. Sample x_1 from $P(x_1)$
2. Sample x_2 from $P(x_2 \mid X_1 = x_1)$
3. Sample x_3 from $P(x_3 \mid X_1 = x_1)$
4. Sample x_4 from $P(x_4 \mid X_2 = x_2, X_3 = x_3)$

But with evidence we have rejection…

In general it may be hard to generate a sample, especially on Markov networks.

Importance Sampling: Overview

$$P(e) = \sum_{X \setminus E} P_B(X, E = e)$$

$$M = \sum_{z \in Z} f(z) = \mathrm{P(e)} \; where \; Z = X \setminus E$$

- Given a proposal or importance distribution Q(z) such that f(z)>0 implies Q(z)>0, rewrite

$$M = \sum_{z \in Z} f(z) = \sum_{z \in Z} f(z) \frac{Q(z)}{Q(z)} = E_Q\left[\frac{f(z)}{Q(z)}\right]$$

- Given i.i.d. samples $z_1,..,z_N$ from Q(z),

unbiased

$$\hat{M} = \frac{1}{N} \sum_{j=1}^{N} \frac{f(z_j)}{Q(z_j)} = \frac{1}{N} \sum_{j=1}^{N} w(z_j) \quad \mathrm{E_Q[\hat{M}] = M = P(e)}$$

Variance reduction

- MSE of estimator Z is equals its variance when bias is zero.

$$Var_Q\left[\hat{Z}\right] = Var_Q\left[\frac{1}{N}\sum_{i=1}^{N} w(x_i)\right] = \frac{Var_Q[w(x)]}{N}$$

- Performance depends on the **variance** of the estimate
  - How close Q(Z) is to P(Z|E).
  - In practice, we can compute Q from the output of Generalized Belief Propagation **(UAI, 2005)**
- The number of samples;

AND/OR sampling: motivation

$$Z = \sum_{x} f(x) = \sum_{x1,x2,x3,x4} f_A(x_1, x_2, x_3) f_B(x_2, x_3, x_4) f_C(x_4)$$

$$Z = \sum_{x} f(x) = \sum_{x1,x2,x3,x4} f_A(x_1) f_B(x_2) f_C(x_3, x_4)$$

Total disregard for the structure of f(x) while approximating it.

$$Z = \sum_{x \in X} f(x) = \sum_{x \in X} f(x) \frac{Q(x)}{Q(x)} = E_Q\left[\frac{f(x)}{Q(x)}\right]$$

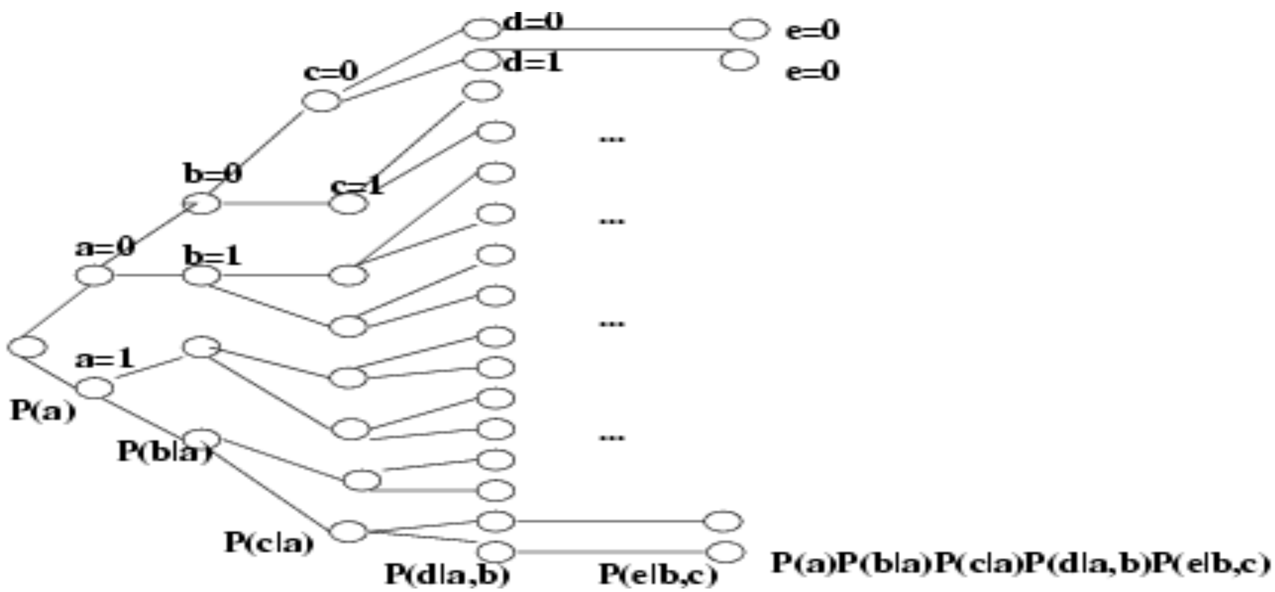
Use AND/OR search to better exploit the structure

Outline

- Background: Bayesian networks
- Importance Sampling
- AND/OR search space for grpahical Models
- AND/OR Sampling: exploiting structure
- AND/OR+w-cutset sampling
- UAI 2010 experience

Conditioning generates the probability tree

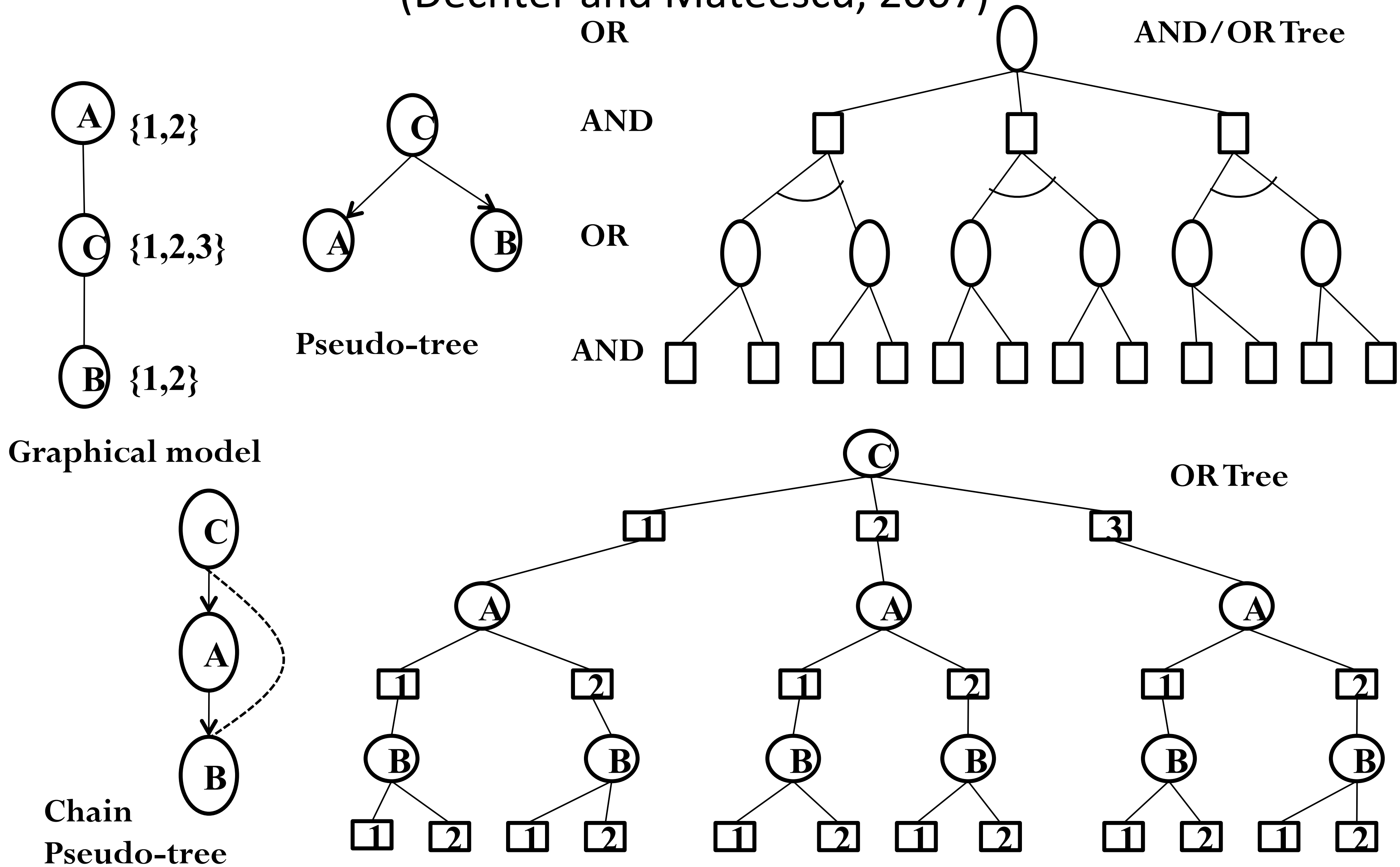
$$P(a,e=0)=P(a)\sum_{b}P(b\,|\,a)\sum_{c}P(c\,|\,a)\sum_{b}P(d\,|\,a,b)\sum_{e=0}P(e\,|\,b,c)$$



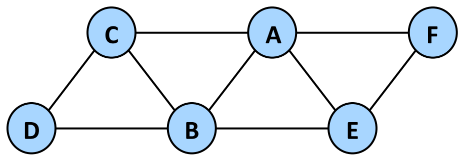
Complexity of conditioning: exponential time, linear space

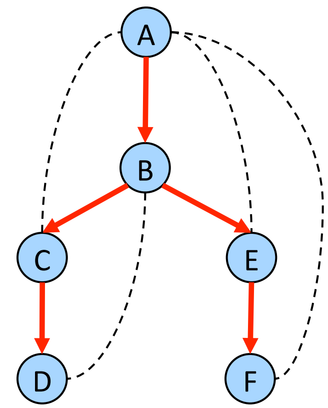
AND/OR search spaces for exact inference

(Dechter and Mateescu, 2007)

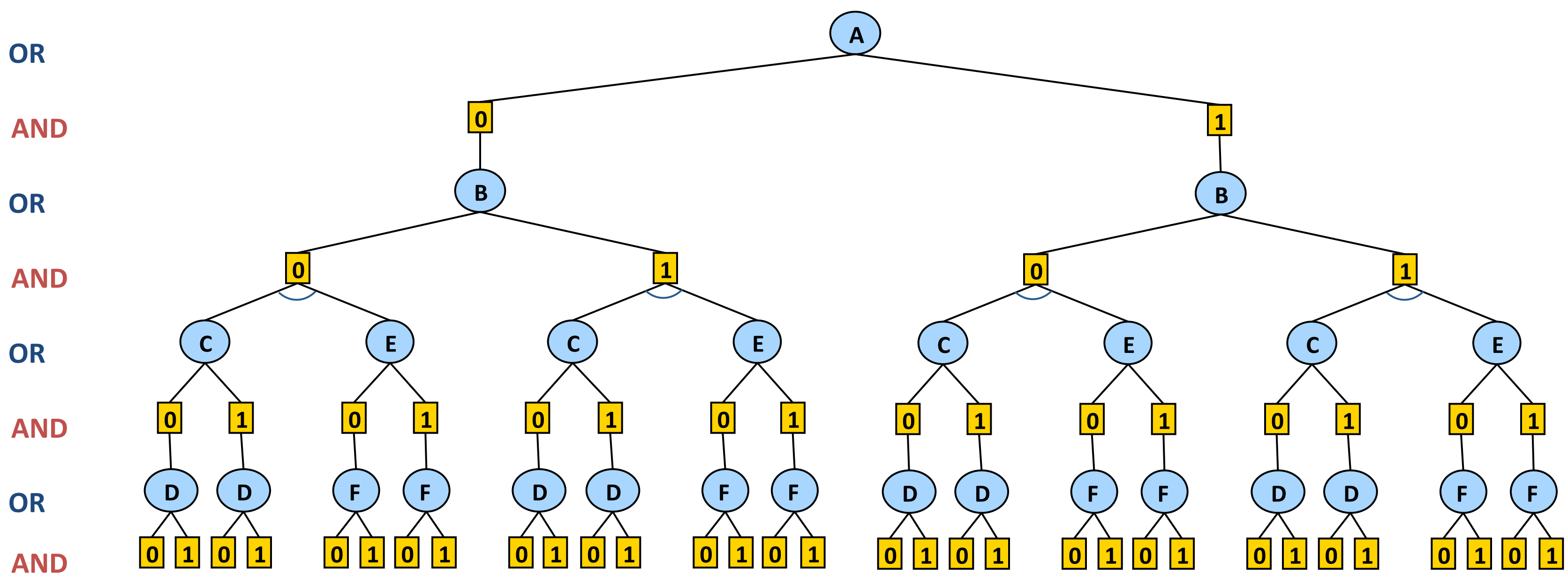


The AND/OR search tree





Pseudo tree



A solution subtree is (A=0, B=1, C=0, D=0, E=1, F=1)

AND/OR search spaces for exact reasoning

Task: compute the value of a root node

$P(E \mid A,B)$

A	B	E=0	E=1
0	0	.4	.6
0	1	.5	.5
1	0	.7	.3
1	1	.2	.8

Evidence: E=0

$P(B \mid A)$

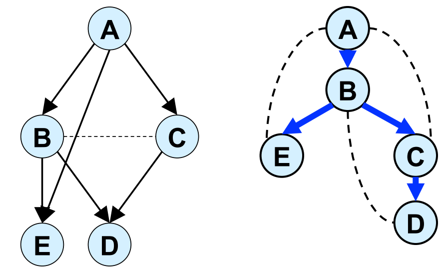
A	B=0	B=1
0	.4	.6
1	.1	.9

$P(C \mid A)$

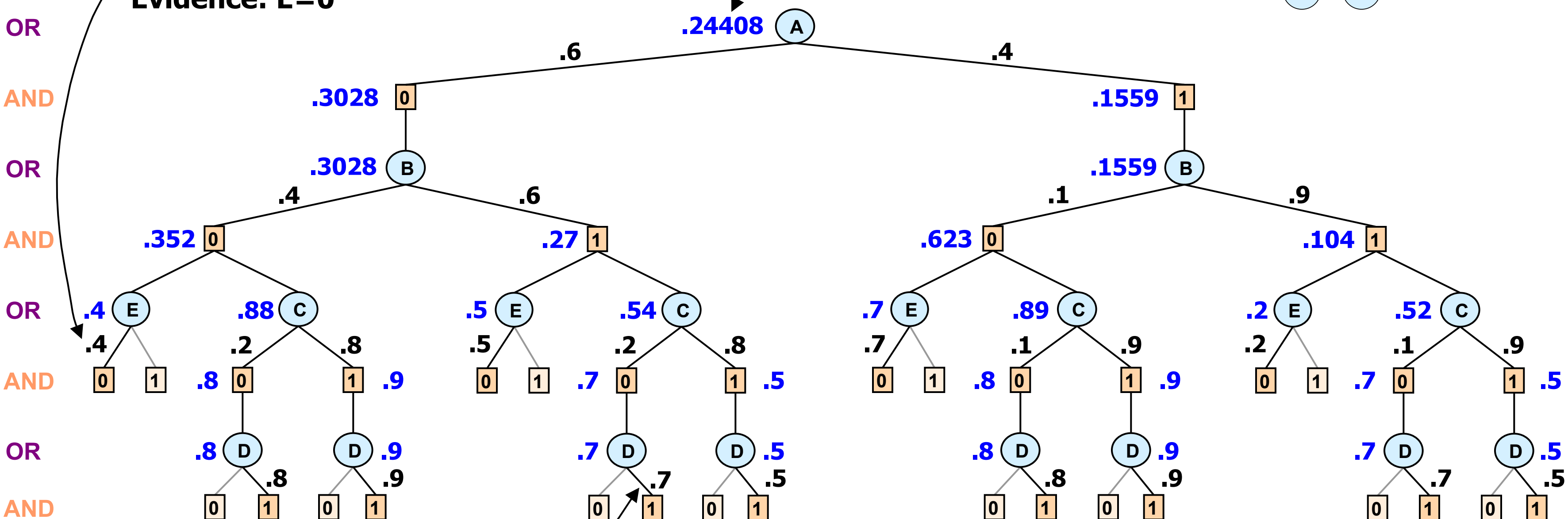
A	C=0	C=1
0	.2	.8
1	.7	.3

$P(A)$

A	P(A)
0	.6
1	.4



Result: P(D=1,E=0)



$P(D \mid B,C)$

B	C	D=0	D=1
0	0	.2	.8
0	1	.1	.9
1	0	.3	.7
1	1	.5	.5

Evidence: D=1

OR node: Marginalization operator (summation)

AND node: Combination operator (product)

Value of node = updated belief for sub-problem below

Complexity of AND/OR tree search

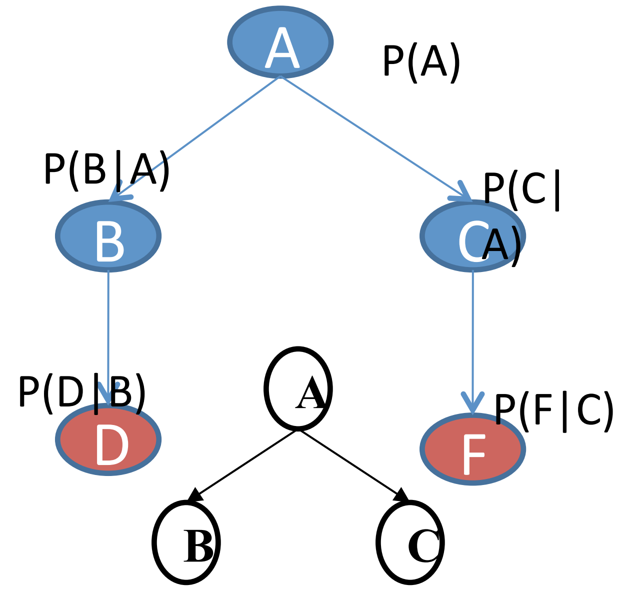
	AND/OR tree	OR tree
Space	$O(n)$	$O(n)$
Time	$O(n\ d^t)$ $O(n\ d^{w^* \log n})$ (Freuder & Quinn85), (Collin, Dechter & Katz91), (Bayardo & Miranker95), (Darwiche01)	$O(d^n)$

d = domain size
t = depth of pseudo-tree
n = number of variables
w*= treewidth

Outline

- Background: Bayesian networks
- Importance Sampling
- AND/OR Sampling: exploiting structure
- AND/OR+w-cutset sampling
- UAI 2010 experience

AND/OR importance sampling (general idea)



Pseudo-tree

- Decompose Expectation

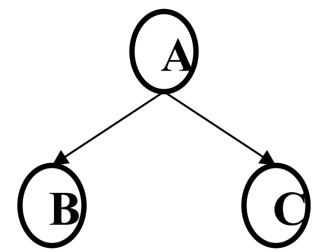
$$P(d,f) = \sum_{a,b,c} P(a)P(c|a)P(b|a)P(d|b)P(f|c)$$

$$Q(A,B,C) = Q(A)Q(B|A)Q(C|A)$$

$$P(d,f) = \sum_{a,b,c} \frac{P(a)P(c|a)P(b|a)P(d|b)P(f|c)}{Q(a)Q(b|a)Q(c|a)} Q(a)Q(b|a)Q(c|a)$$

$$= E_Q\left[\frac{P(a)P(c|a)P(b|a)P(d|b)P(f|c)}{Q(a)Q(b|a)Q(c|a)}\right]$$

AND/OR importance sampling (general idea)



Pseudo-tree

- Decompose Expectation

$$P(d,f)=\sum_{a,b,c}\frac{P(a)P(c|a)P(b|a)P(d|b)P(f|c)}{Q(a)Q(b|a)Q(c|a)}Q(a)Q(b|a)Q(c|a)$$

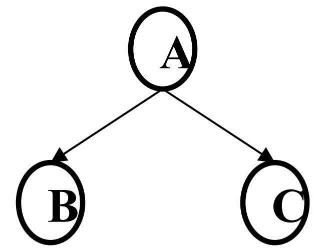
$$P(d,f)=\sum_{a}\frac{P(a)}{Q(a)}Q(a)\sum_{b}\frac{P(b|a)P(d|b)}{Q(b|a)}Q(b|a)\sum_{c}\frac{P(c|a)P(f|c)}{Q(c|a)}Q(c|a)$$

$$P(d,f)=E_Q\left[\frac{P(a)}{Q(a)}E_Q\left[\frac{P(b|a)P(d|b)}{Q(b|a)}|a\right]E_Q\left[\frac{P(c|a)P(f|c)}{Q(c|a)}|a\right]\right]$$

AND/OR importance sampling (general idea)

$$P(d,f) = E_Q\left[\frac{P(a)}{Q(a)} E_Q\left[\frac{P(b|a)P(d|b)}{Q(b|a)} \,|\, a\right] E_Q\left[\frac{P(c|a)P(f|c)}{Q(c|a)} \,|\, a\right]\right]$$

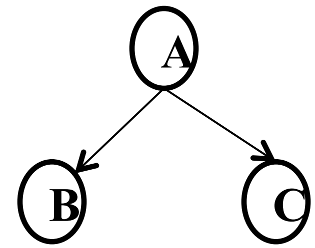
- Compute all expectations separately
- How?
  - Record all samples
  - For each sample that has A=a
    - Estimate the conditional expectations separately using the generated samples
    - Combine the results



Pseudo-tree

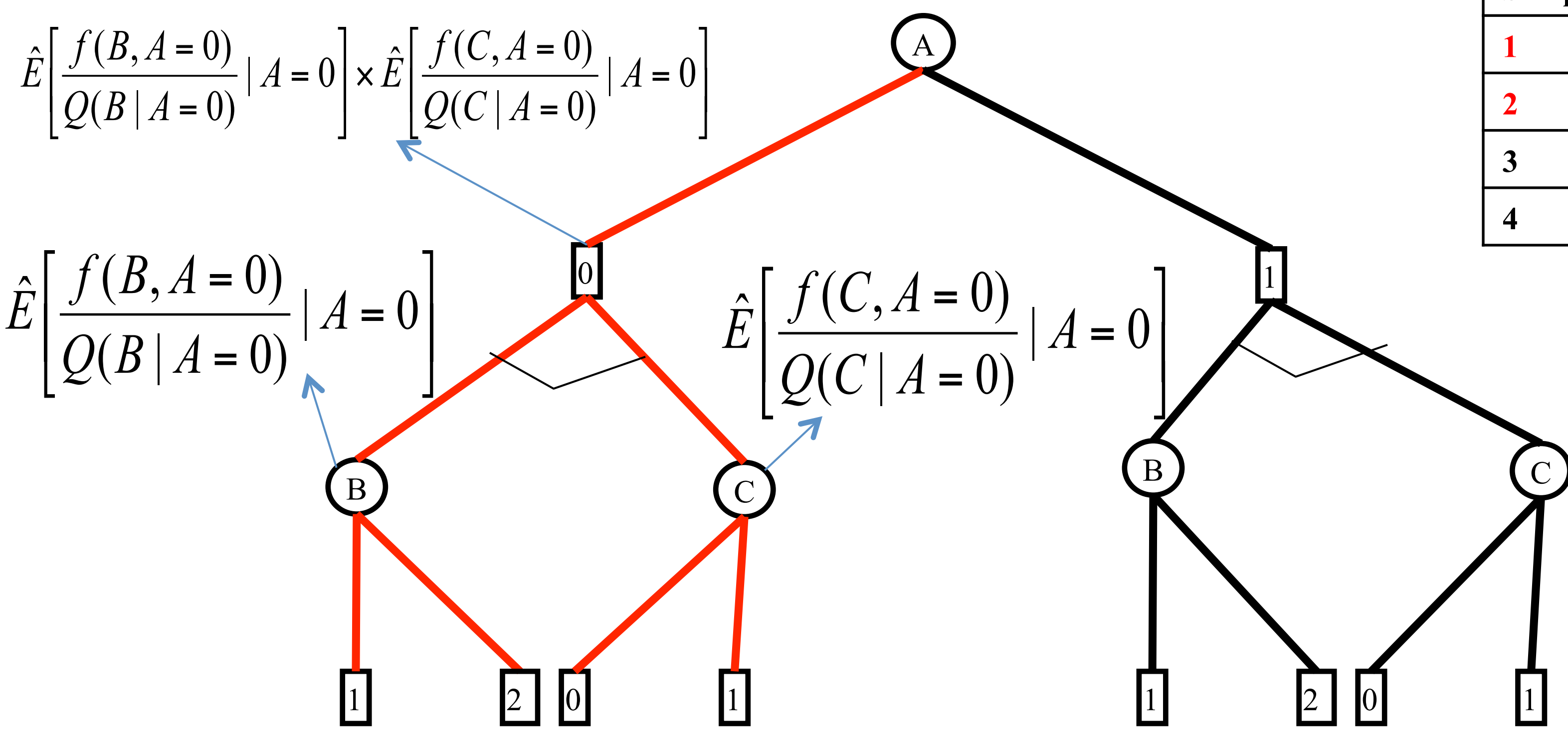
AND/OR tree importance sampling

$$Z = E_Q\left[\frac{f(a)}{Q(a)} E_Q\left[\frac{f(b,a)}{Q(b|a)} | a\right] E_Q\left[\frac{f(c,a)}{Q(c|a)} | a\right]\right]$$

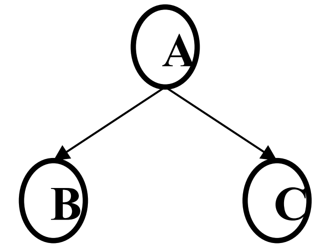


Pseudo-tree

Sample #	A	B	C
1	0	1	0
2	0	2	1
3	1	1	1
4	1	2	0

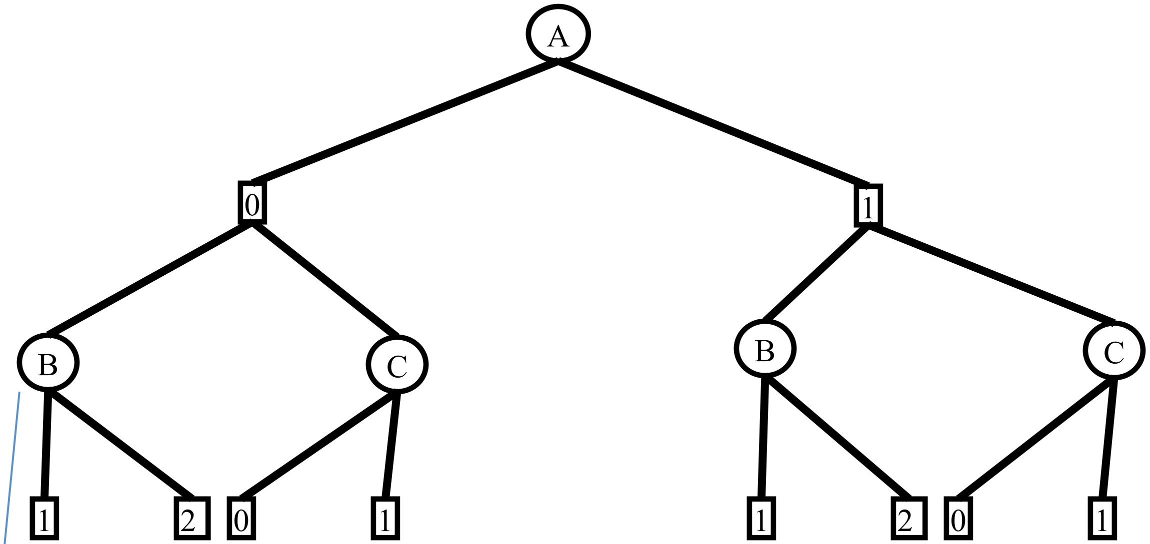


AND/OR importance sampling



Pseudo-tree

Sample #	A	B	C
1	0	1	0
2	0	2	1
3	1	1	1
4	1	2	0



$$\text{Estimate of } E\left[\frac{P(b \mid A=0)P(d \mid b)}{Q(b \mid A=0)} \mid A=0\right]$$

$$= \text{Average Weight of samples of B having A} = 0$$

$$= \frac{\text{w}(\text{B}=1, \text{A}=0) + \text{w}(\text{B}=2, A=0)}{2}$$

AND/OR sample tree

- Given a pseudo-tree and a set of samples, **S**
- 1. Generate the AND/OR search tree
- 2. Remove nodes/edges not appearing in **S**.
- 3. Assign weights (w(n,m) ,#(n,m))
  - w(n,m) = the weights (p/q)
  - #(n,m) = frequency of the partial sample in **S**
- The AOT estimate is the value of the root node.

Algorithm AND/OR importance sampling

1. Generate samples $\mathbf{x}_1, \ldots, \mathbf{x}_N$ *from Q along O.*
2. Build a AND/OR sample tree f*or the samples* $x_1, \ldots, x_N$ along the ordering *O.*

3. FOR all leaf nodes *i of AND-OR tree do*

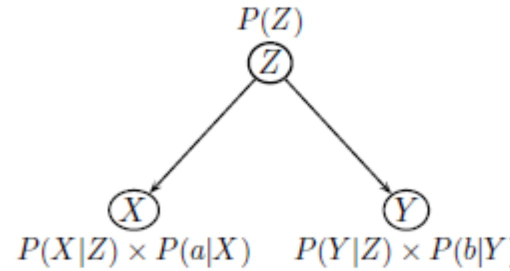
1. IF AND-node v(i)= 1 ELSE v(i)=0

4. FOR every node *n from leaves to the root do*

1. IF AND-node v(n)=product of children

2. IF OR-node v(n) = Average of children

A Bayesian network

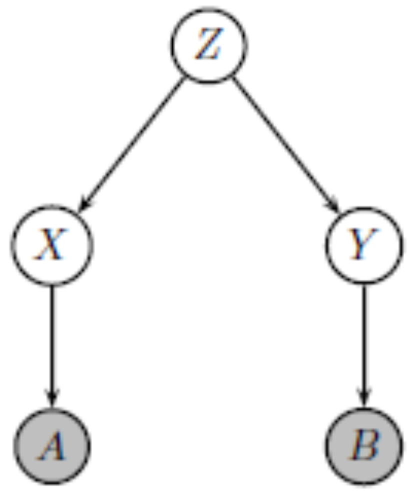


P(X|Z)

Z	X=0	X=1	X=2
0	.3	.4	.3
1	.2	.7	.1

P(A|X)

X	A=0	A=1
0	.1	.9
1	.2	.8
2	.6	.4



Evidence: $A = 0, B = 0$

P(Z)

Z=0	Z=1
.8	.2

P(Y|Z)

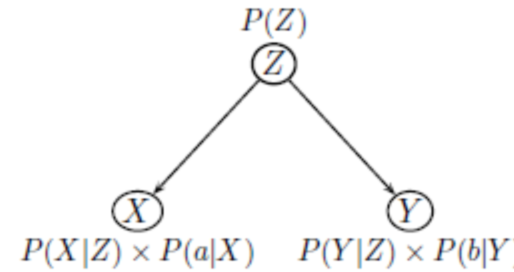
Z	Y=0	Y=1	Y=2
0	.5	.1	.4
1	.2	.6	.2

P(B|Y)

Y	B=0	B=1
0	.2	.8
1	.7	.3
2	.1	.9

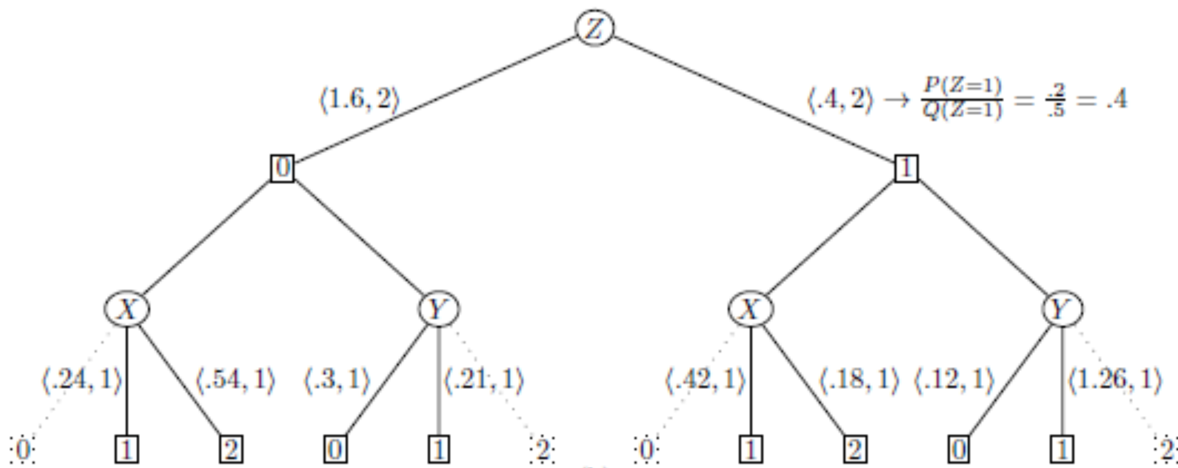
Figure 3: A Bayesian network and its CPTs

Example of AND/OR sample tree



Sample #	Z	X	Y
1	0	1	0
2	0	2	1
3	1	1	1
4	1	2	0

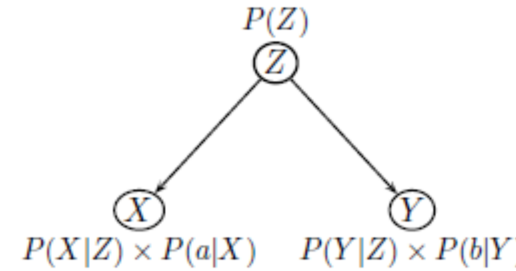
(a)



(b)

Figure 5: (a) Four samples drawn from a uniform proposal distribution $Q(X, Y, Z) = Q(X)Q(Y)Q(Z)$ where $Q(Z = 0) = Q(Z = 1) = 1/2$, $Q(X = 0) = Q(X = 1) = Q(X = 2) = 1/3$ and $Q(Y = 0) = Q(Y = 1)$ $= Q(Y = 2) = 1/3$, (b) The samples in (a) arranged on a full AND/OR search tree. Dotted edges and nodes are not sampled. Each arc from an OR node to an AND node is labeled by its weight and frequency (see Definition 14).

Example of value cor



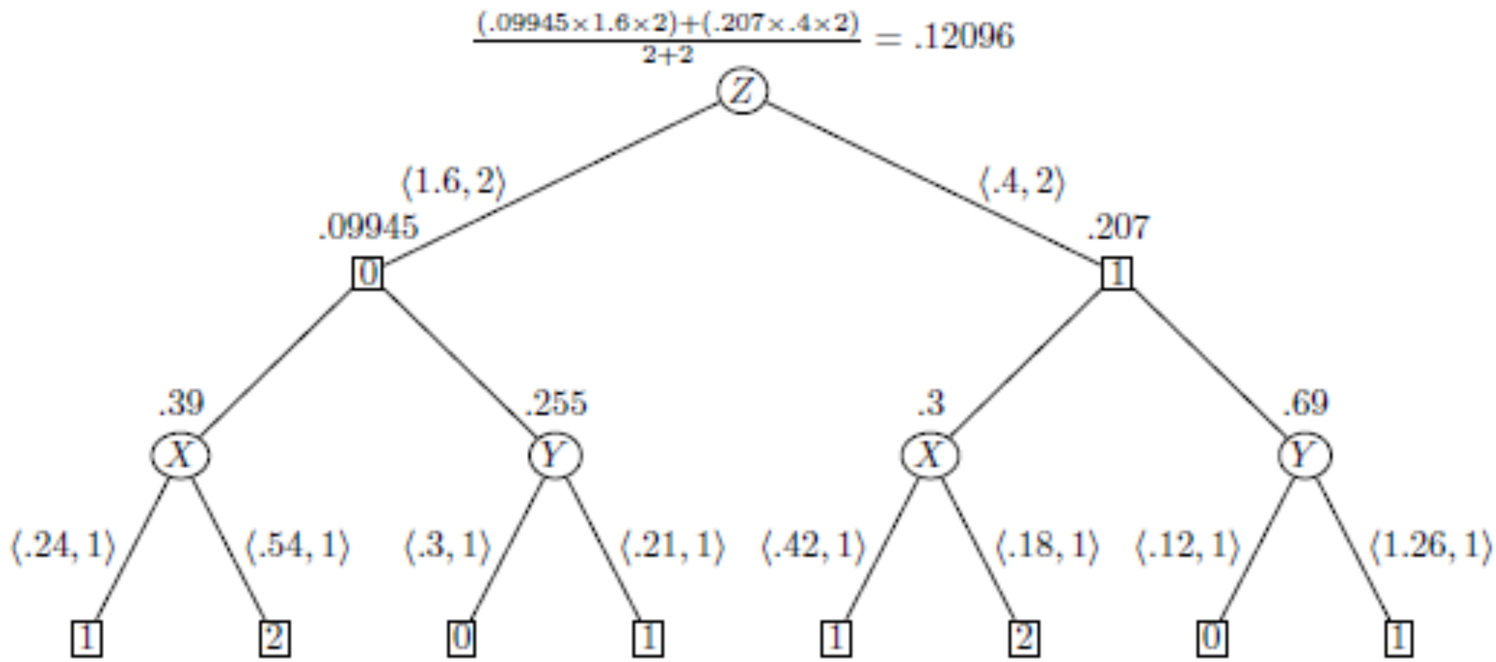
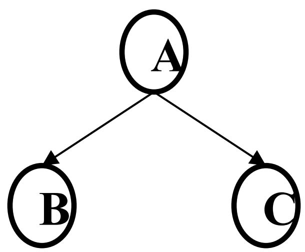
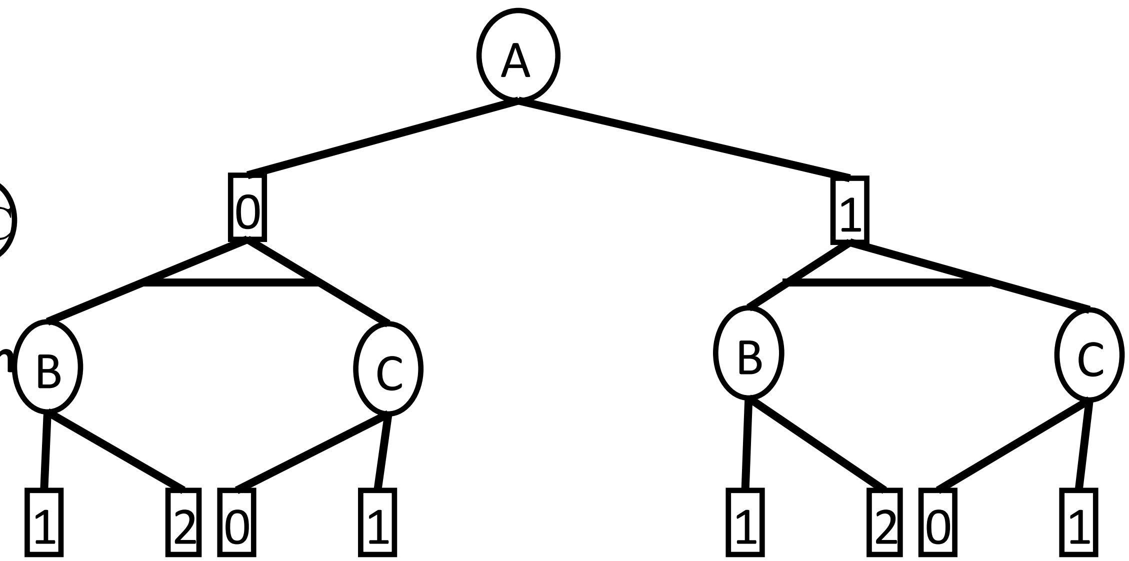


Figure 6: Value computation on an AND/OR sample tree (see Definition 15). Each OR and AND node is annotated with its value. By definition, the value of a leaf AND node is 1 and is not shown to avoid clutter. The AND/OR sample tree mean (which equals the value of the root OR node Z) is 0.12096.

AND/OR sampling: more virtual samples



Problem Graph



A	B	C
0	1	0
0	2	1
1	1	1
1	2	0

4 samples

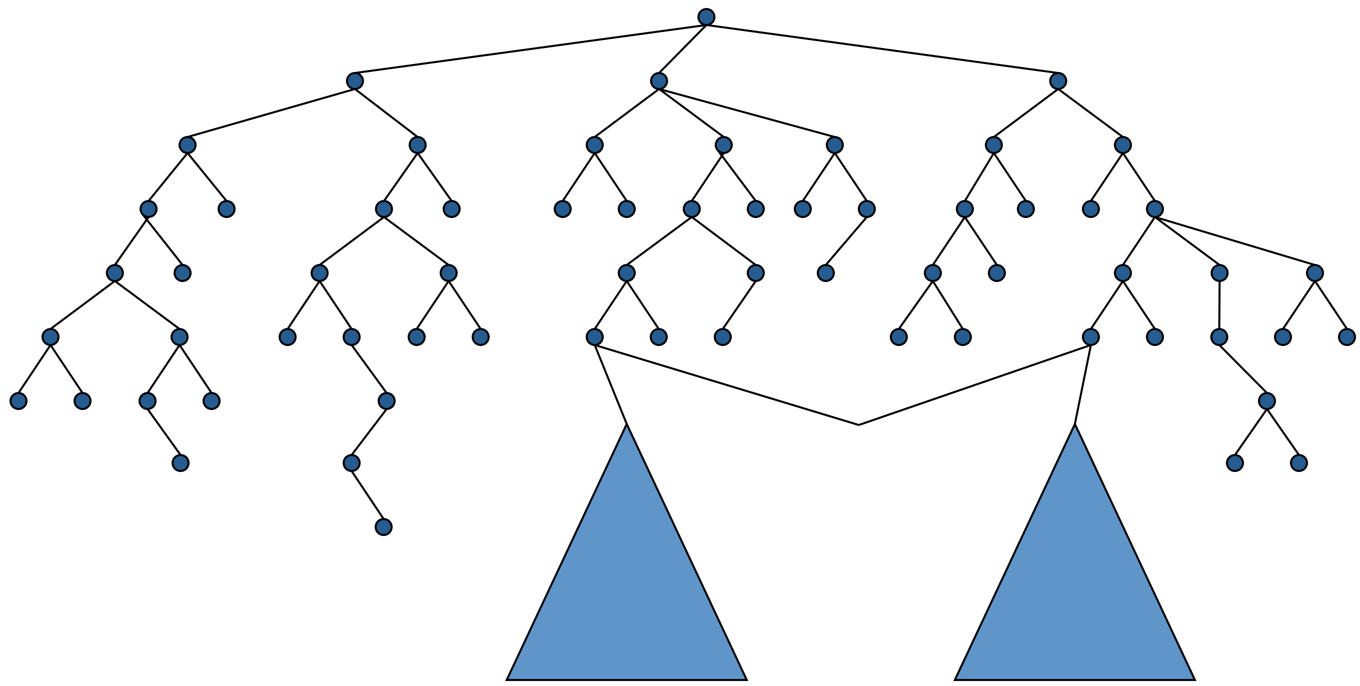
- Example: A=0, B=2, C=0 is not generated but still considered in the AND/OR space
- 8 virtual samples in AND/OR space versus 4 samples.
- In general, we get exponentially more samples.

Properties of AO-Tree Mean (AOTM)

- AOTM is an unbiased estimator of Z
- On a chain it reduces to IS estimator
- Theorem: Variance of AOTM is always smaller or equal regular IS
- Time complexity: O(nN), N= |S|
- Space complexity: O(h), h= height of tree

From search trees to search graphs

- Any two nodes that root identical subtrees (subgraphs) can be merged



AND/OR search tree

(Belief Updating)

$P(E \mid A,B)$

A	B	E=0	E=1
0	0	.4	.6
0	1	.5	.5
1	0	.7	.3
1	1	.2	.8

Evidence: E=0

$P(B \mid A)$

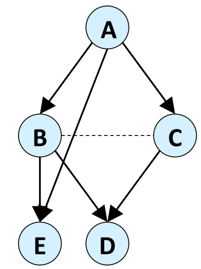
A	B=0	B=1
0	.4	.6
1	.1	.9

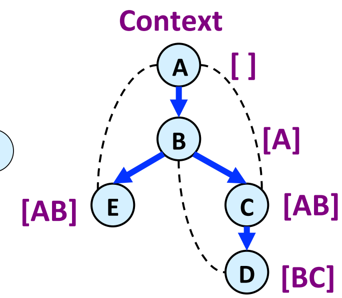
$P(C \mid A)$

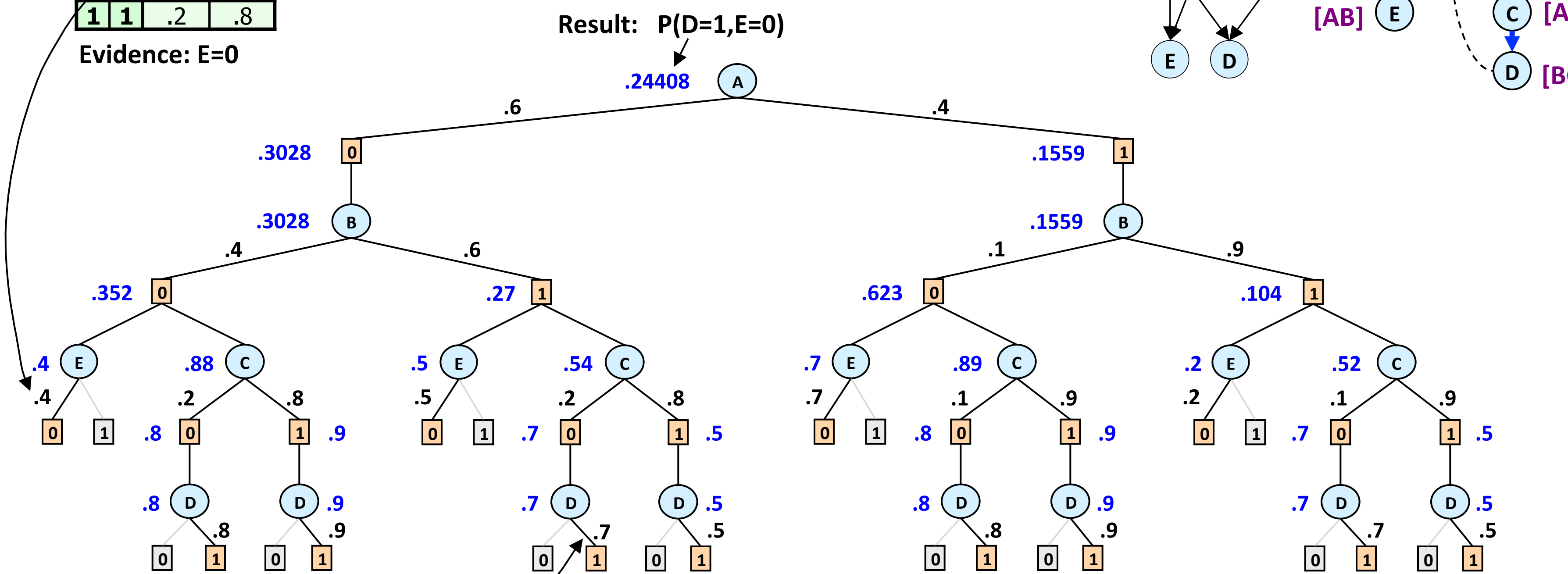
A	C=0	C=1
0	.2	.8
1	.7	.3

$P(A)$

A	P(A)
0	.6
1	.4







$P(D \mid B,C)$

B	C	D=0	D=1
0	0	.2	.8
0	1	.1	.9
1	0	.3	.7
1	1	.5	.5

Evidence: D=1

OR node: Marginalization operator (summation)

AND node: Combination operator (product)

Value of node = updated belief for sub-problem below

AND/OR search **graph**

(Belief Updating)

$P(E \mid A,B)$

A	B	E=0	E=1
0	0	**.4**	.6
0	1	.5	.5
1	0	.7	.3
1	1	.2	.8

Evidence: E=0

$P(B \mid A)$

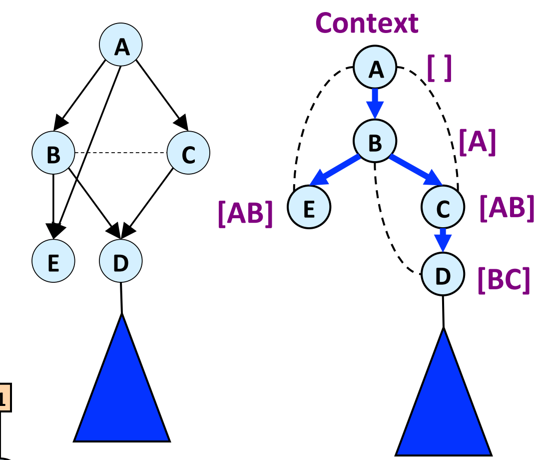
A	B=0	B=1
0	.4	.6
1	.1	.9

$P(C \mid A)$

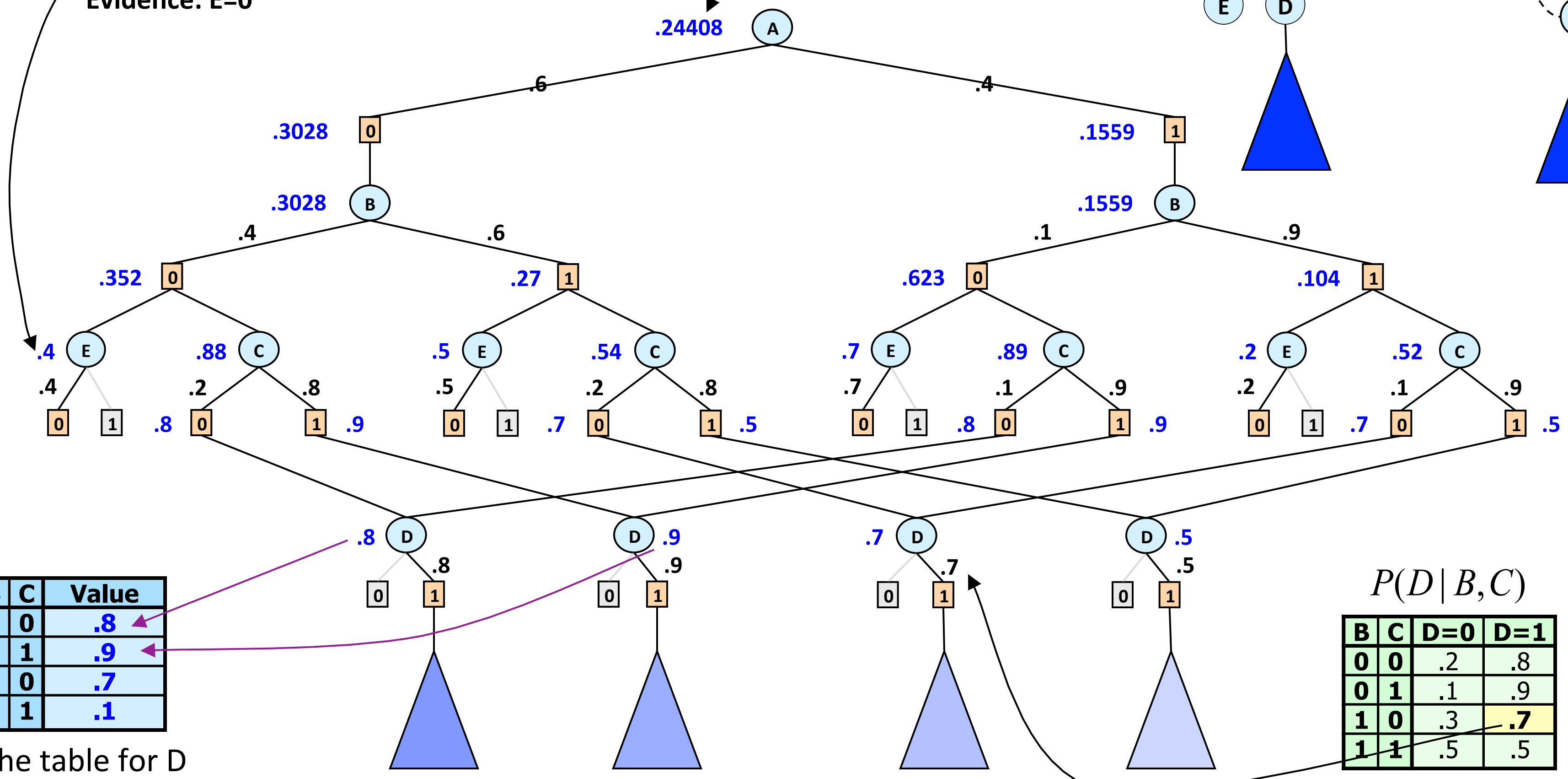
A	C=0	C=1
0	.2	.8
1	.7	.3

$P(A)$

A	P(A)
0	.6
1	.4



Result: P(D=1,E=0)



B	C	Value
0	0	.8
0	1	.9
1	0	.7
1	1	.1

Cache table for D

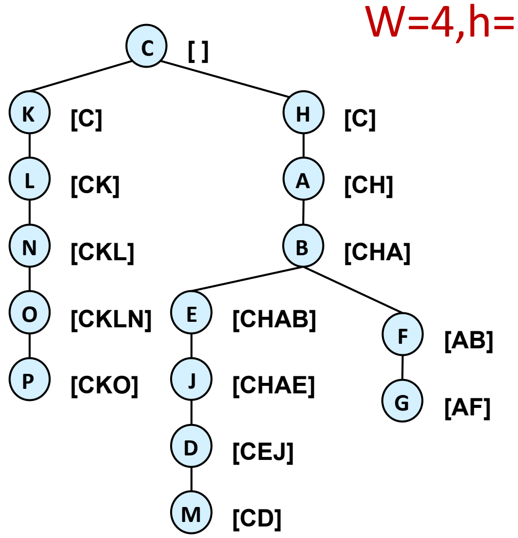
$P(D \mid B,C)$

B	C	D=0	D=1
0	0	.2	.8
0	1	.1	.9
1	0	.3	**.7**
1	1	.5	.5

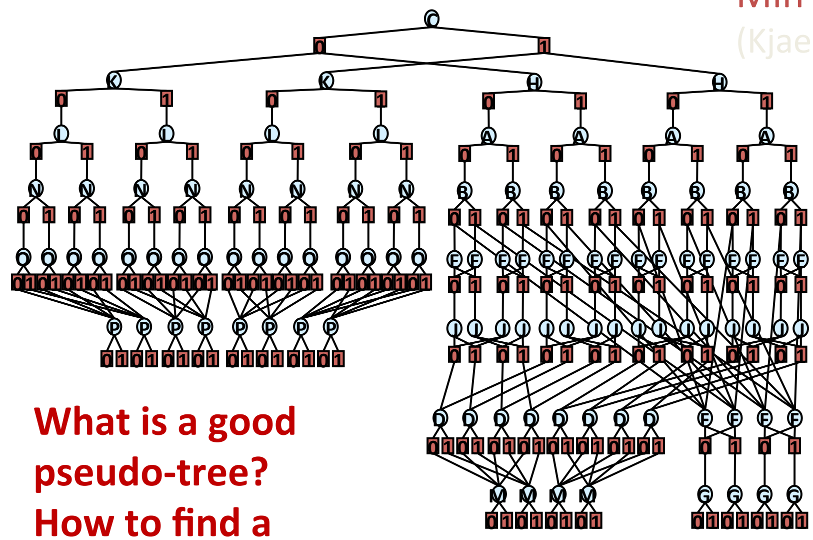
Evidence: D=1

The impact of the pseudo-tree

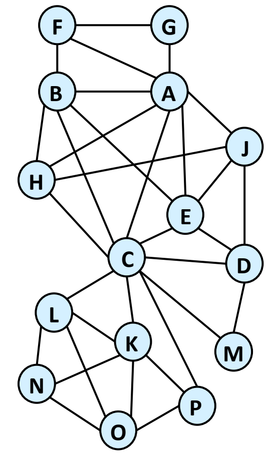
W=4,h=8



(C K H A B E J L N O D P M F G)



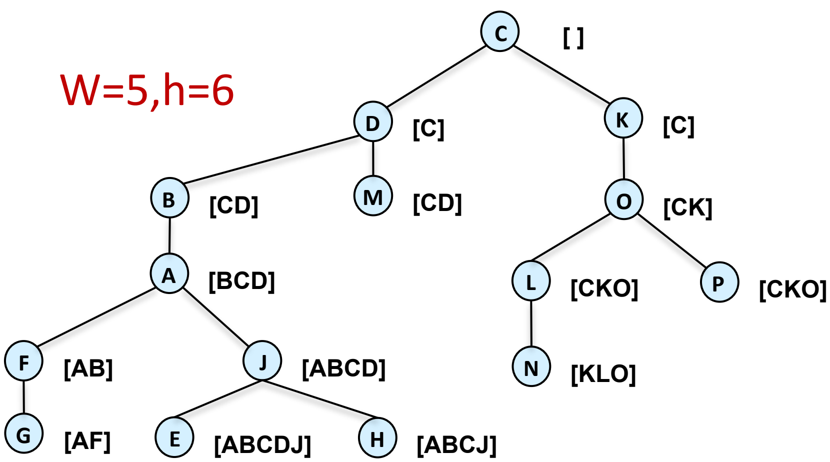
Min-Fill
(Kjaerulff90)

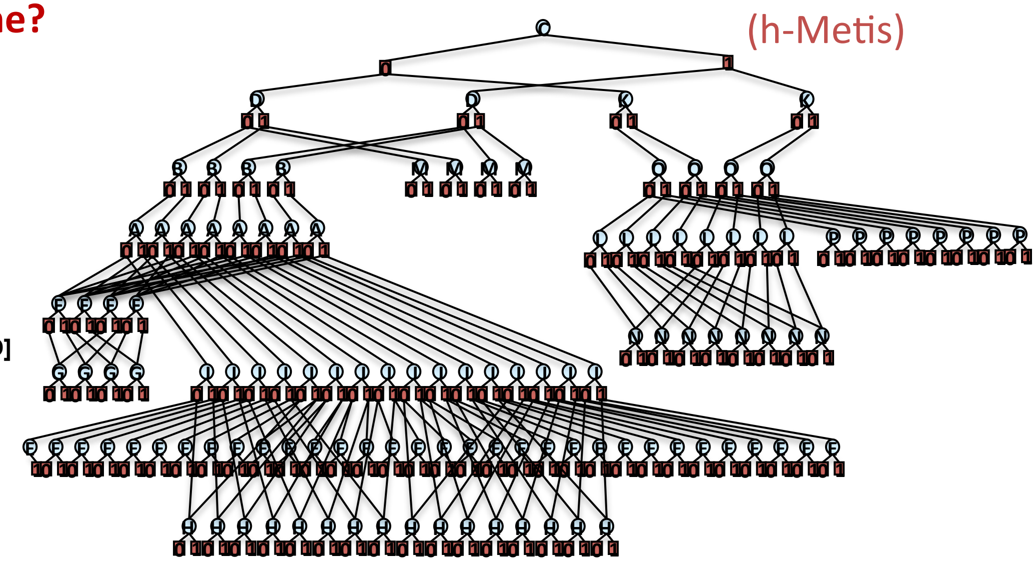


What is a good pseudo-tree? How to find a good one?

Hypergraph Partitioning
(h-Metis)

W=5,h=6

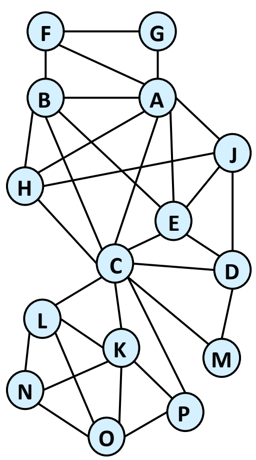


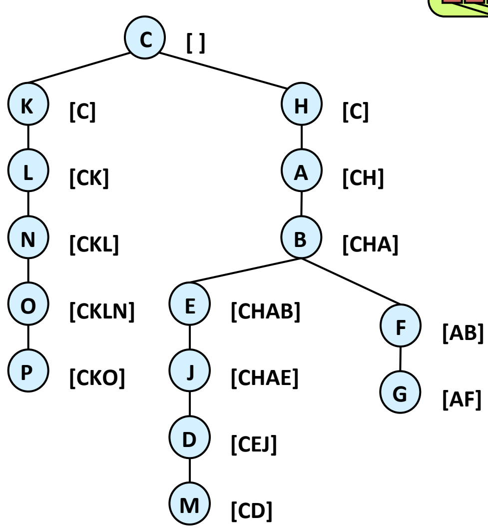


BGU 2011

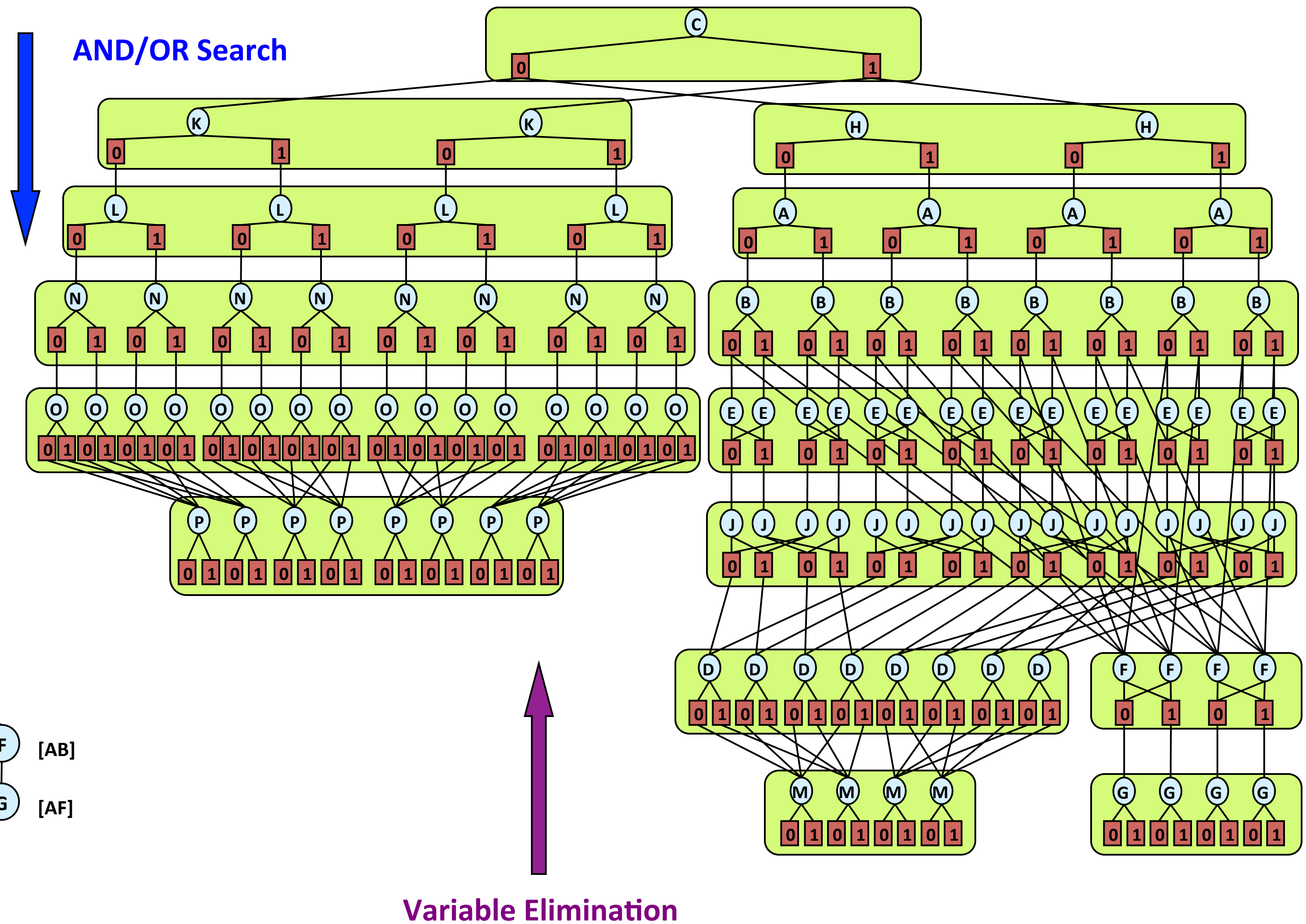
(C D K B A O M L N P J H E F G)

AND/OR context-minimal graph





(C K H A B E J L N O D P M F G)



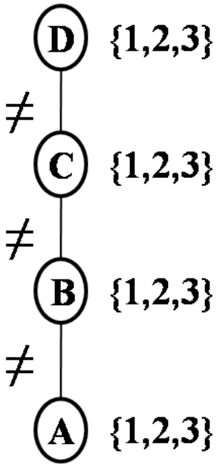
Complexity of AND/OR graph search

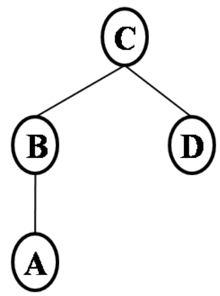
	AND/OR graph	OR graph
Space	$O(n\ d^{w^*})$	$O(n\ d^{pw^*})$
Time	$O(n\ d^{w^*})$	$O(n\ d^{pw^*})$

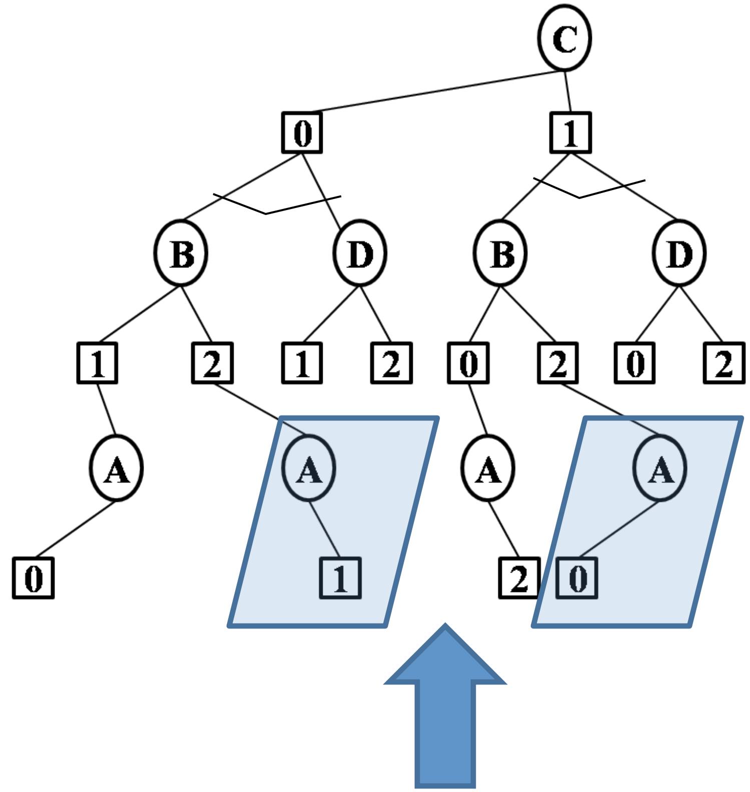
d = domain size
n = number of variables
w*= treewidth
pw*= pathwidth

$w^* \leq pw^* \leq w^* \log n$

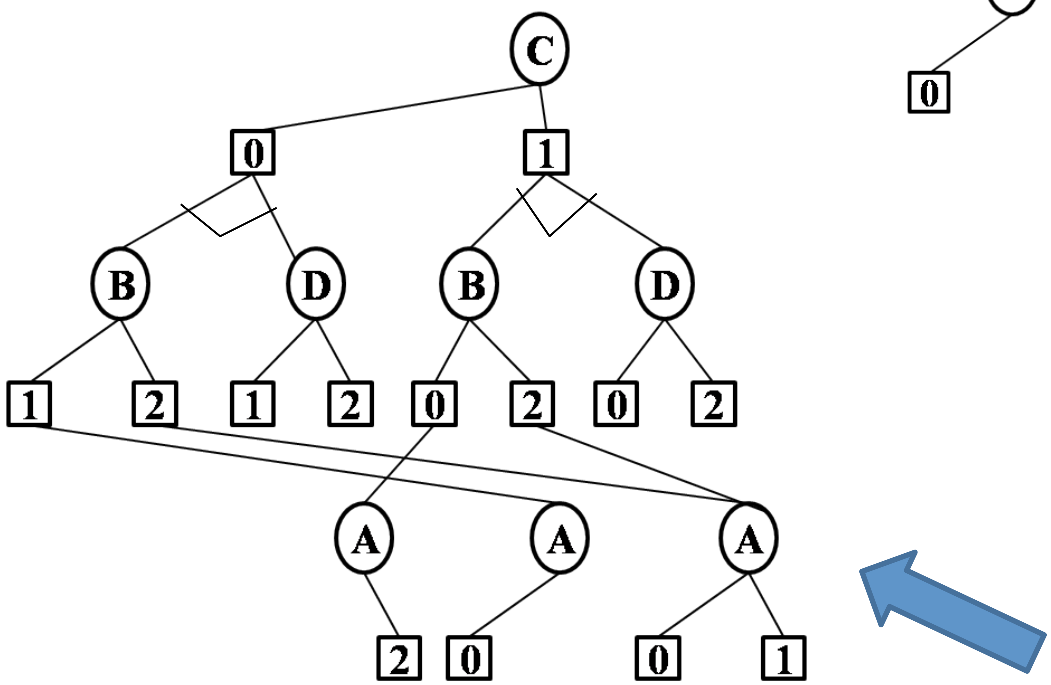
AND/OR tree vs graph sampling





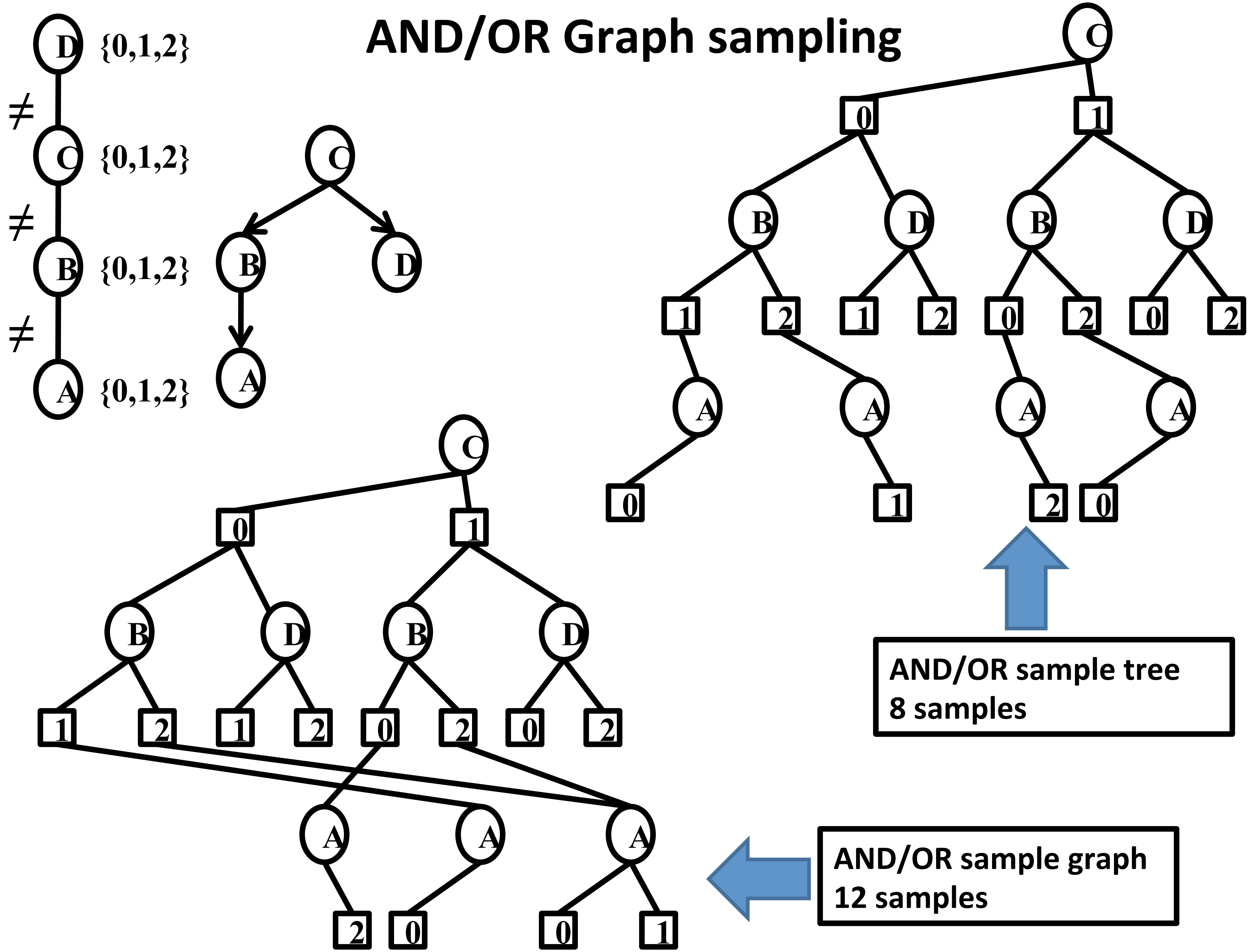


AND/OR sample tree
Complexity: O(nN)



AND/OR sample graph
Complexity : O(nwN)
where w: treewidth

AND/OR Graph sampling



OR tree vs. A/O tree vs A/O-graph

- AND/OR tree
  - Time Complexity: O(nN)
  - Space Complexity: O(h)
  - **AOTM lower Variance than OR IS**
- AND/OR Graph
  - Time Complexity: O(nwN)
  - Space Complexity: O(nN)
  - **AOGM has Lower Variance than AOT**
- Something in between Cache on "i" variables
  - Time Complexity: O(niN)
  - Space Complexity: O(nN)
  - Variance in between.

Experiments

Hypergraph decomposition

Algorithms	Benchmarks
or-tree-IJGP-IS-hmetis	
ao-tree-IJGP-IS-hmetis	Alarm
ao-graph-IJGP-IS-hmetis	Grids
or-tree-IJGP-IS-minfill	
ao-tree-IJGP-IS-minfill	
ao-graph-IJGP-IS-minfill	

(a) **Algorithms evaluated on benchmarks without Strong Deterministic Relationships**

Algorithms	Benchmarks
or-tree-IJGP-SS-hmetis	
ao-tree-IJGP-SS-hmetis	Coding
ao-graph-IJGP-SS-hmetis	Linkage
or-tree-IJGP-SS-minfill	Graph Coloring
ao-tree-IJGP-SS-minfill	
ao-graph-IJGP-SS-minfill	

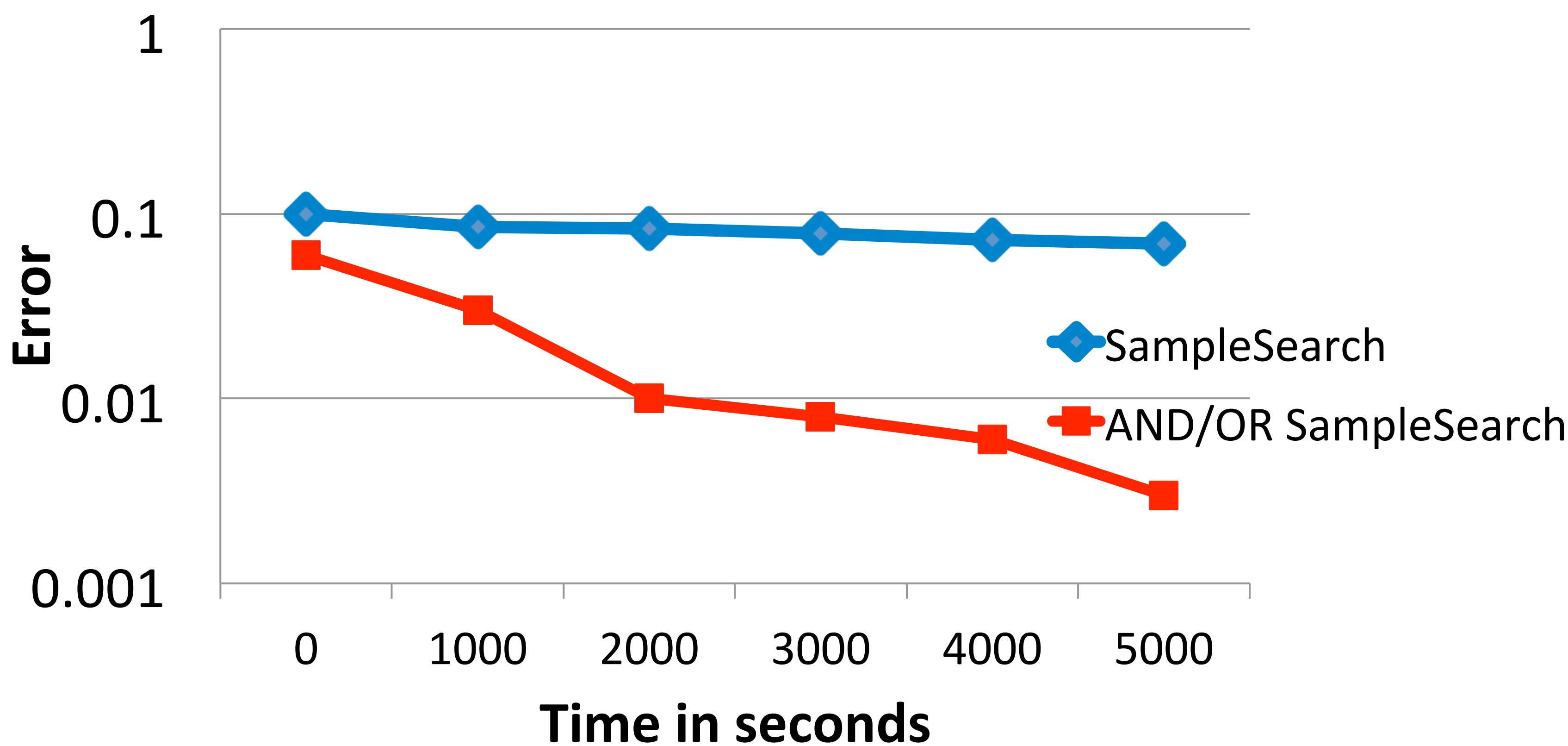
(b) **Algorithms evaluated on benchmarks with Strong Deterministic Relationships**

minfill

AND/OR sampling: Some experimental results

(UAI, 2008; AIJ, 2011)

Linkage analysis instance having ~500K variables



Grid instances

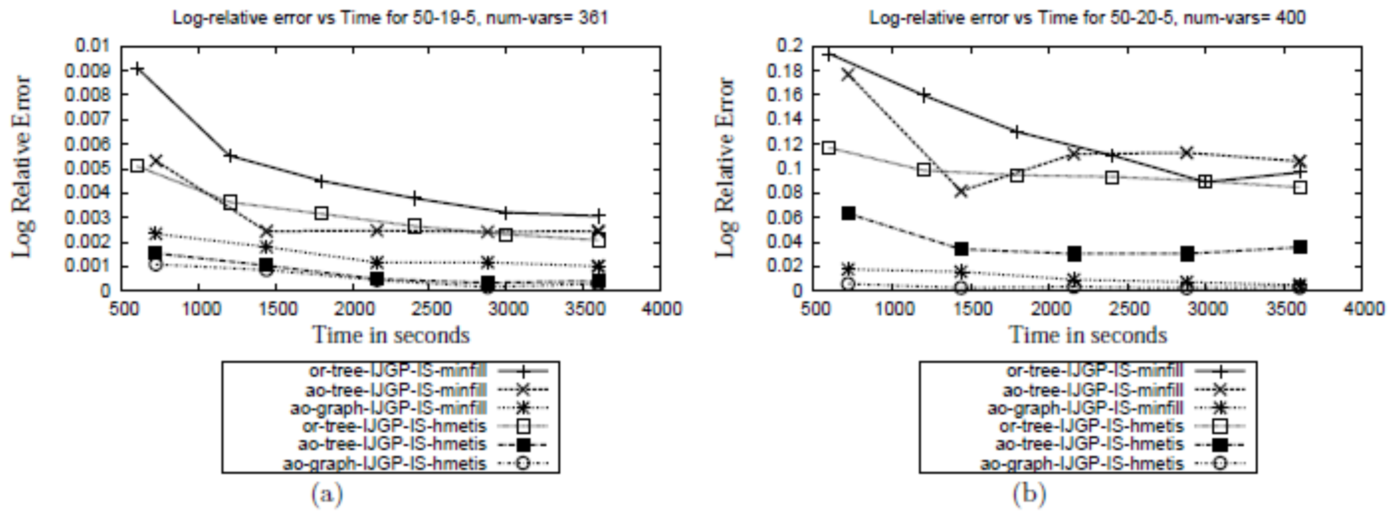


Figure 12: Log-relative error versus time plots for the two largest Grid instances with Deterministic ratio = 50%.

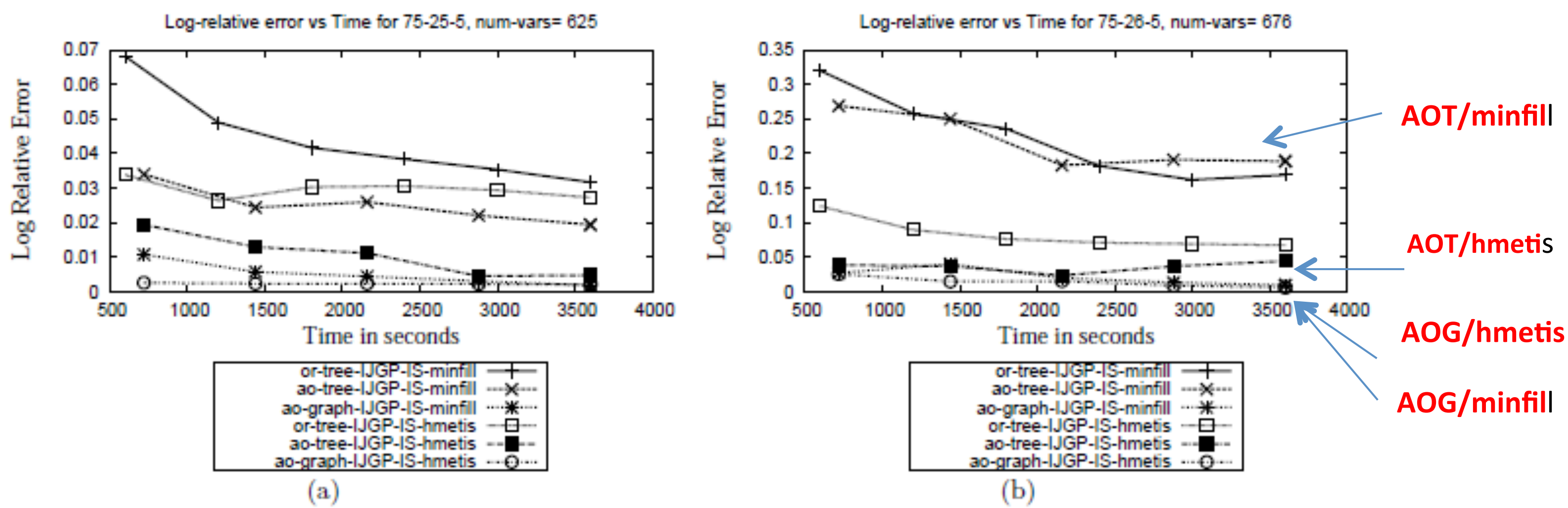


Figure 13: Log-relative error versus time plots for the two largest Grid instances with Deterministic ratio = 75%.

Linkage instances

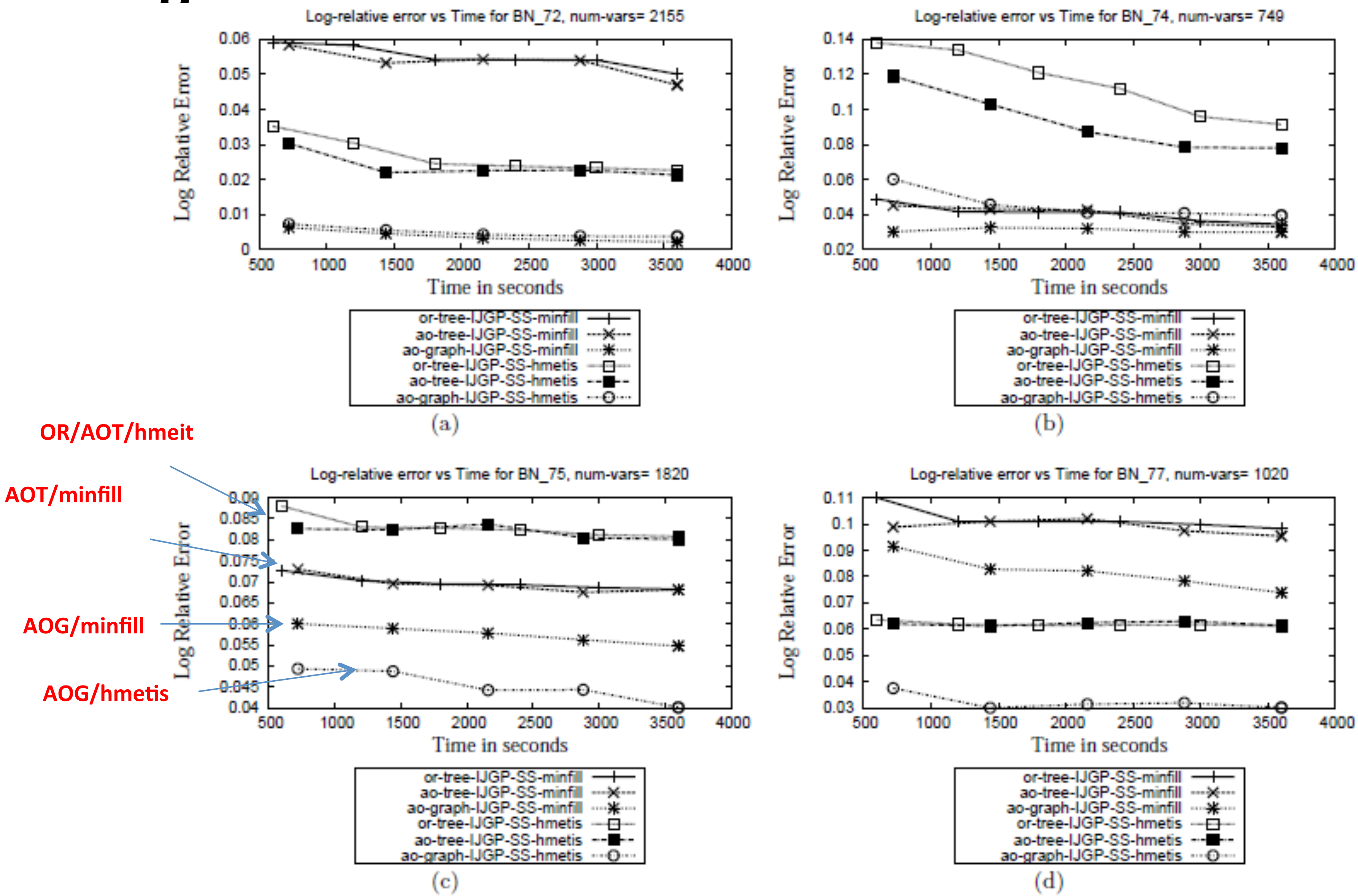
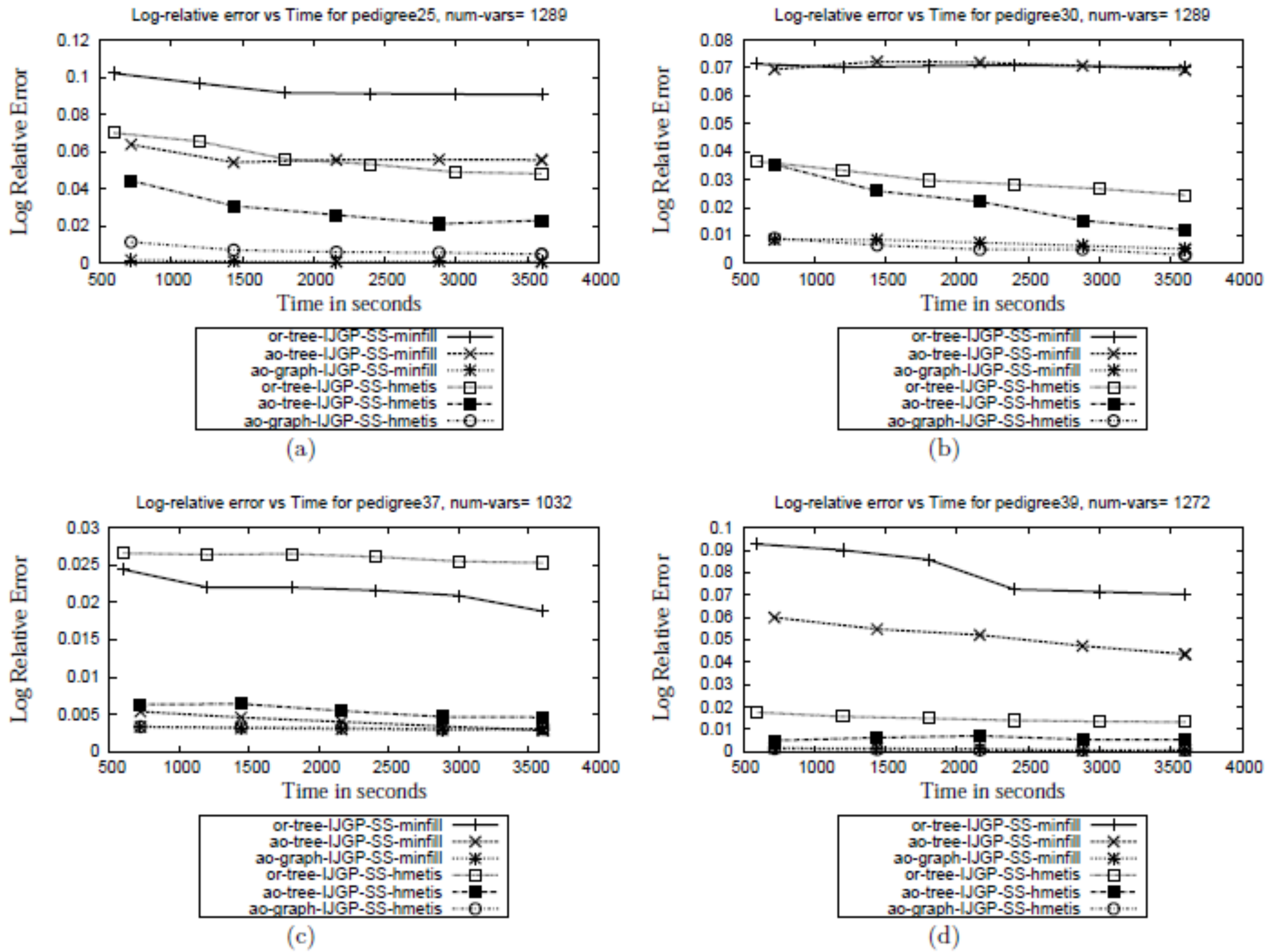


Figure 16: Log-relative error versus time plots for four sample linkage instances from the UAI 2006 evaluation.

Linkage instances



'igure 17: Log-relative error versus time plots for four sample linkage instances from the UAI 2008 evaluation.

Outline

- Background: Bayesian networks
- Importance Sampling
- AND/OR search space for grpahical Models
- AND/OR Sampling: exploiting structure
- AND/OR+w-cutset sampling
- UAI 2010 experience

Rao-Blackwellised w-cutset sampling

- Rao-Blackwellisation
  - Partition X into two sets K and R, such that we can compute Z(R|k) efficiently.

$$\tilde{Z} = \frac{1}{N} \sum_{i=1}^{N} \frac{\sum_{R} Z(R \mid k_i)}{Q(k_i)}$$

partition function given evidence $K=k_i$

- **Smaller variance**
- **w-cutset sampling**: Select K such that the treewidth of R is bounded by “w” (Bidyuk and Dechter, 2007)

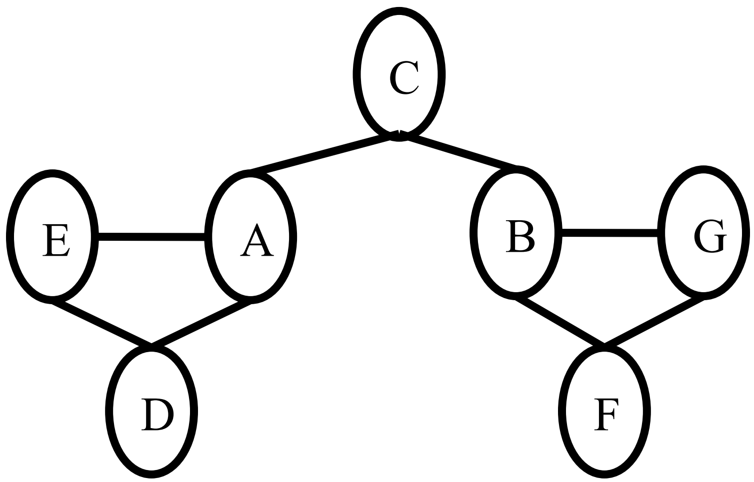
Algorithm AND/OR w-cutset sampling

Given an integer constant w

1. Partition variables into K and R, such that treewidth of R is <= w.
2. AND/OR sampling on K
   1. Construct a pseudo-tree of K and compute Q(K) consistent with K
   2. Generate samples from Q(K) and store them on an AND/OR tree
3. Rao-Blackwellisation (Exact inference) at each leaf
   1. For each leaf node of the tree compute Z(R|g) where g is the assignment from the leaf to the root.
4. Value computation: Recursively from the leaves to the root
   1. At each AND node compute product of values at children
   2. At each OR node compute a weighted average over the values at children
5. Return the value of the root node

AND/OR w-cutset sampling:

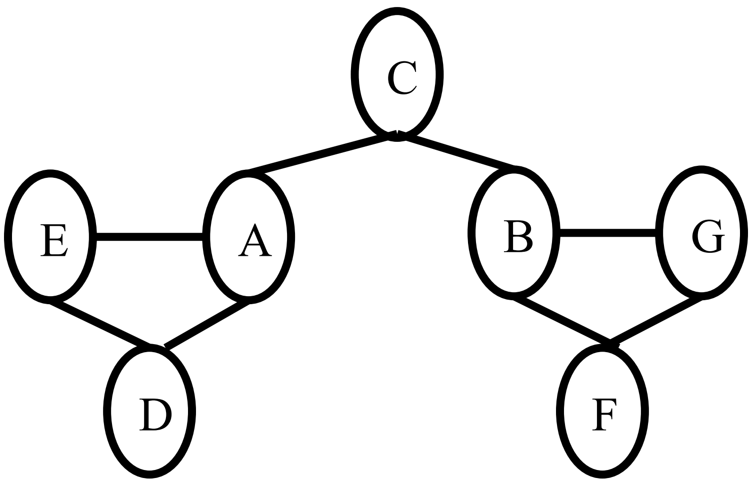
Step 1: Partition the set of variables



Graphical model

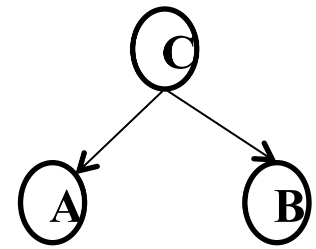
W=1

AND/OR w-cutset sampling: Step 2: AND/OR sampling over {A,B,C}



Graphical model

Ch sta pseudo-tree over {A,B,C} Sample A,B and C without taking into account conditional independence

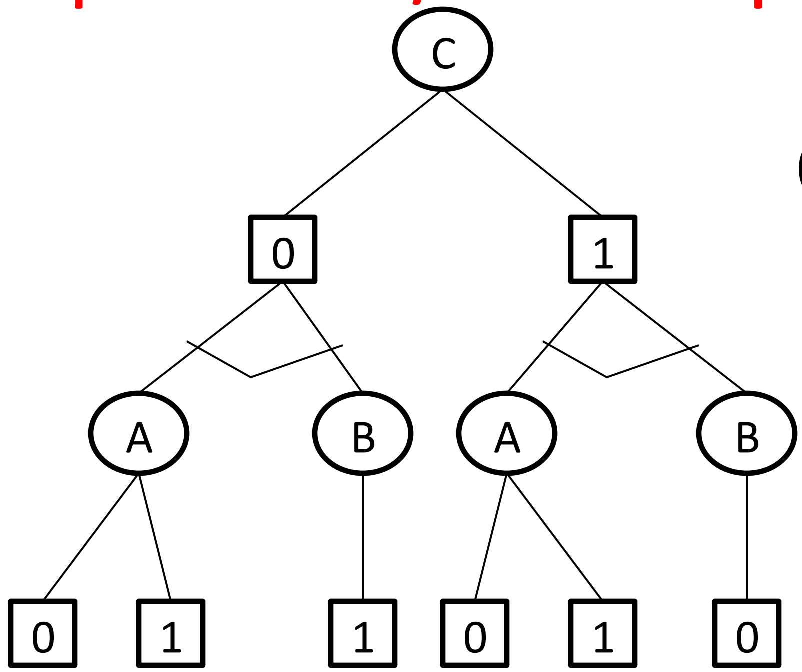


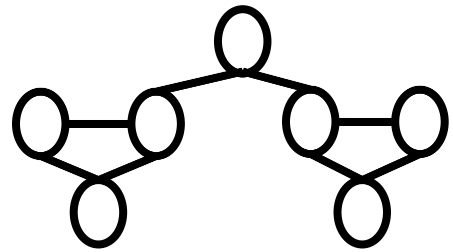
Pseudo-tree

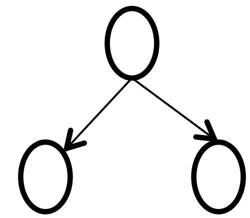
A start pseudo-tree over {A,B,C} that takes into account conditional independence properties

AND/OR w-cutset sampling:

Step 2: AND/OR sampling over {A,B,C}



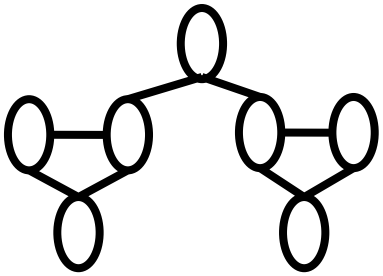


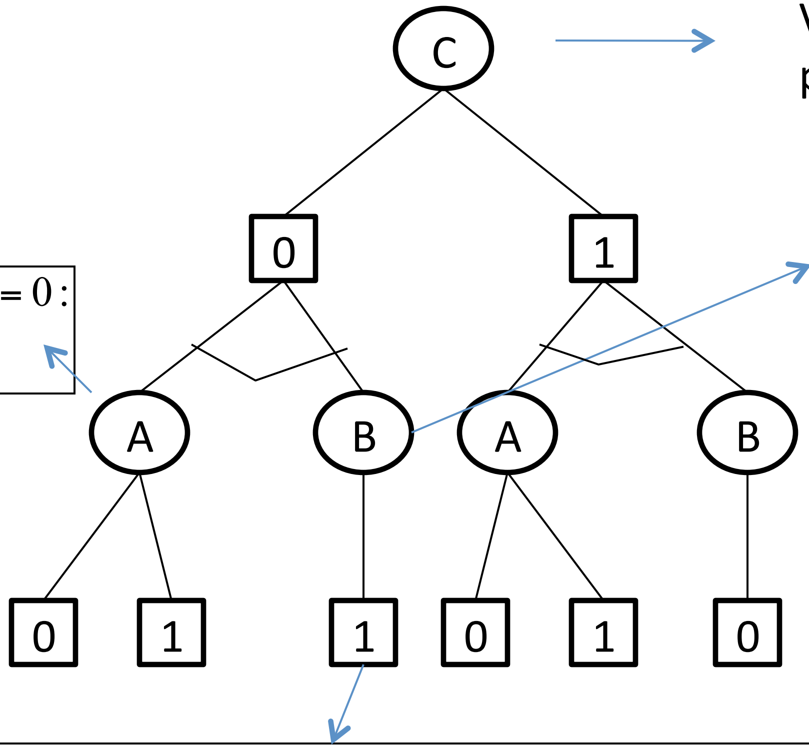

Pseudo-tree

Samples: (C=0,A=0,B=1), (C=0,A=1,B=1), (C=1,A=0,B=0), (C=1,A=1,B=0)

AND/OR w-cutset sampling:

Step 4: Value computation





Value of C: Estimate of the partition function

Value of B given C = 0:
$= \hat{E}[B, F, G \mid C = 0]$

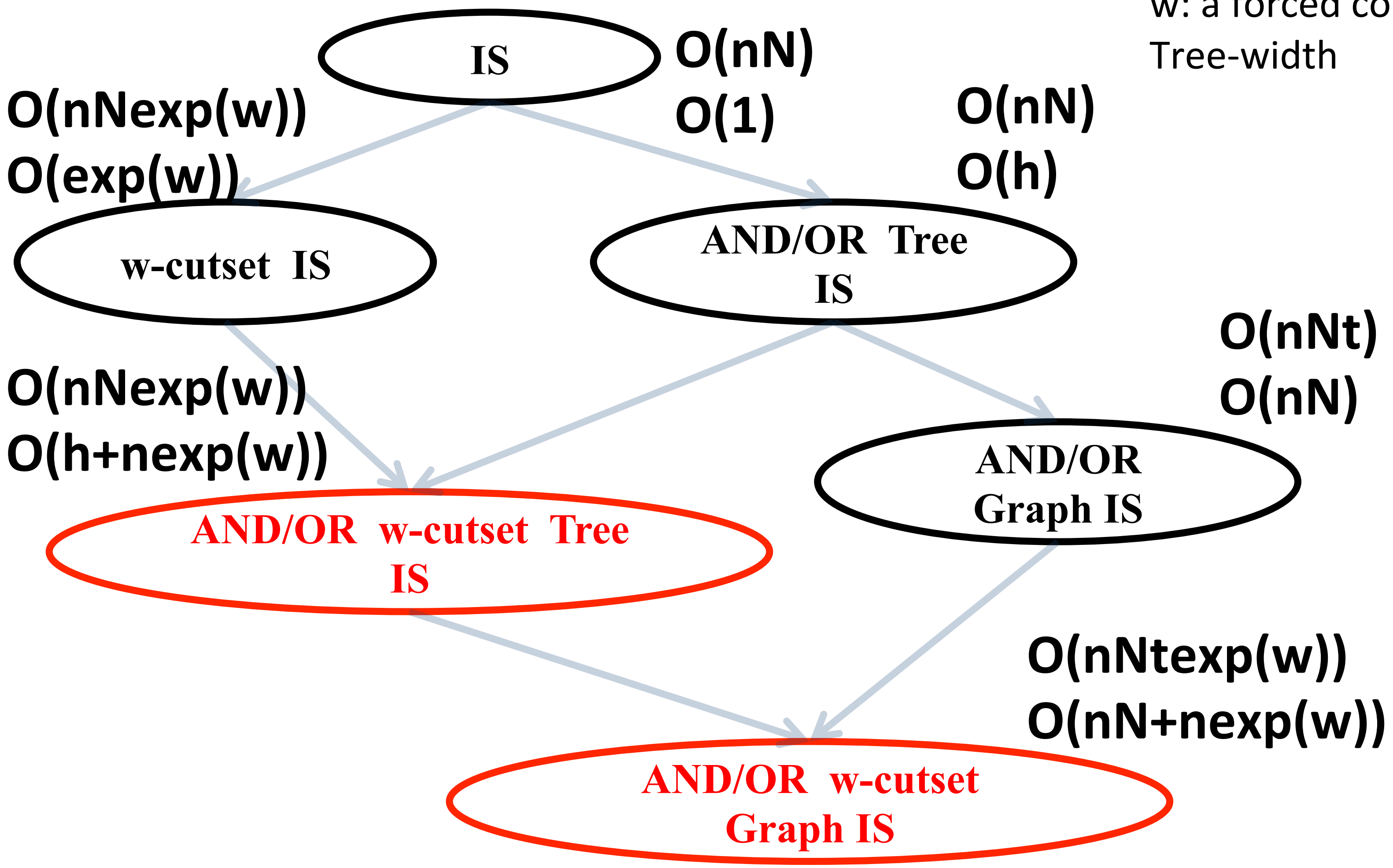
Value of A given C = 0:
$= \hat{E}[A, D, E \mid C = 0]$

Value of B = 1 given C = 0:

$$v(B = 1 \mid C = 0) = \sum_{f \in F} \sum_{g \in G} H(C = 0, g) H(B = 1, g) H(g, f) H(B = 1, f)$$

Variance Hierarchy and Complexity

t: the treewidt
w: a forced conditioned
Tree-width



Experiments

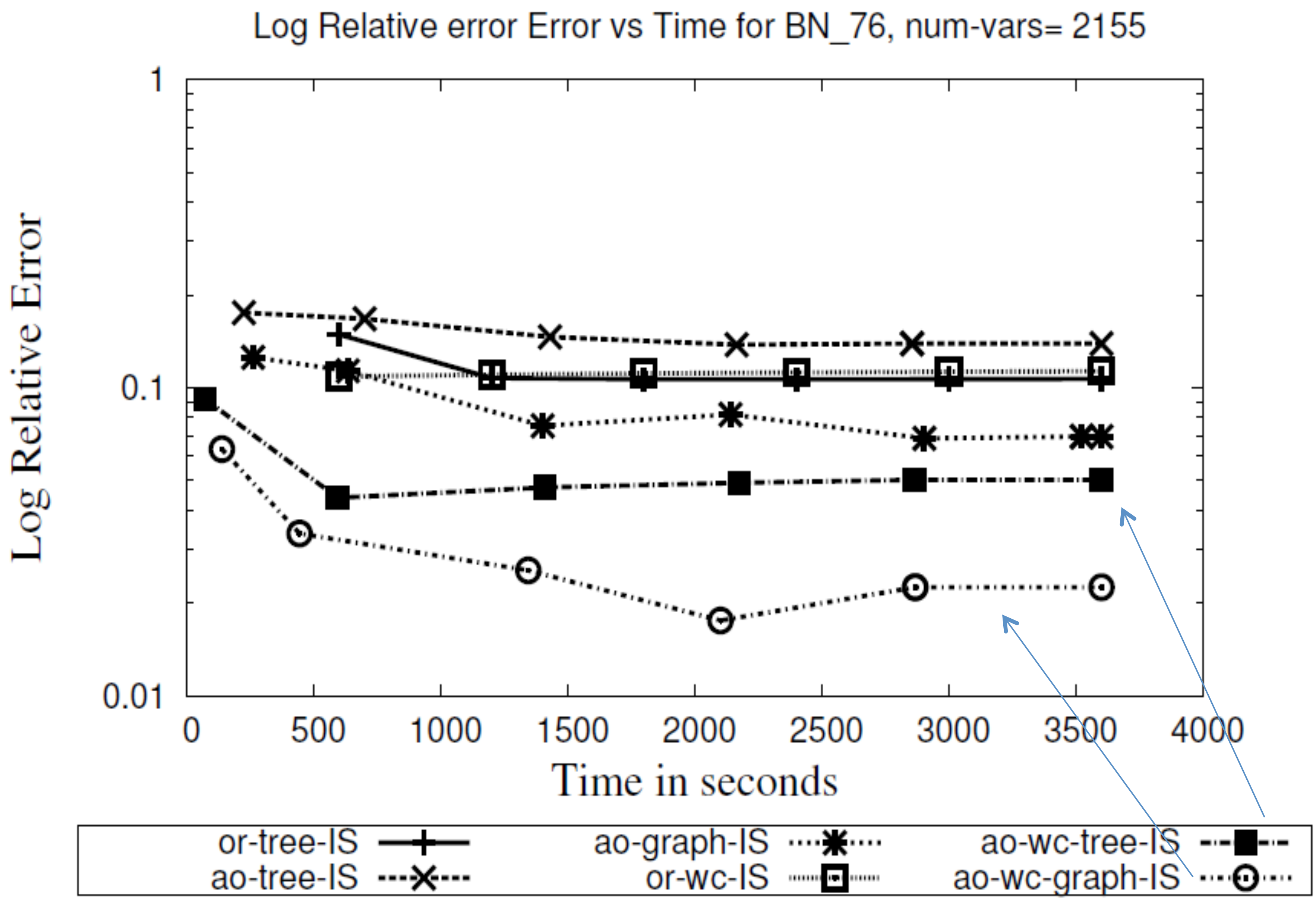
- Benchmarks
  - Linkage analysis
  - Graph coloring
- Algorithms
  - OR tree sampling
  - AND/OR tree sampling
  - AND/OR graph sampling
  - w-cutset versions of the three schemes above

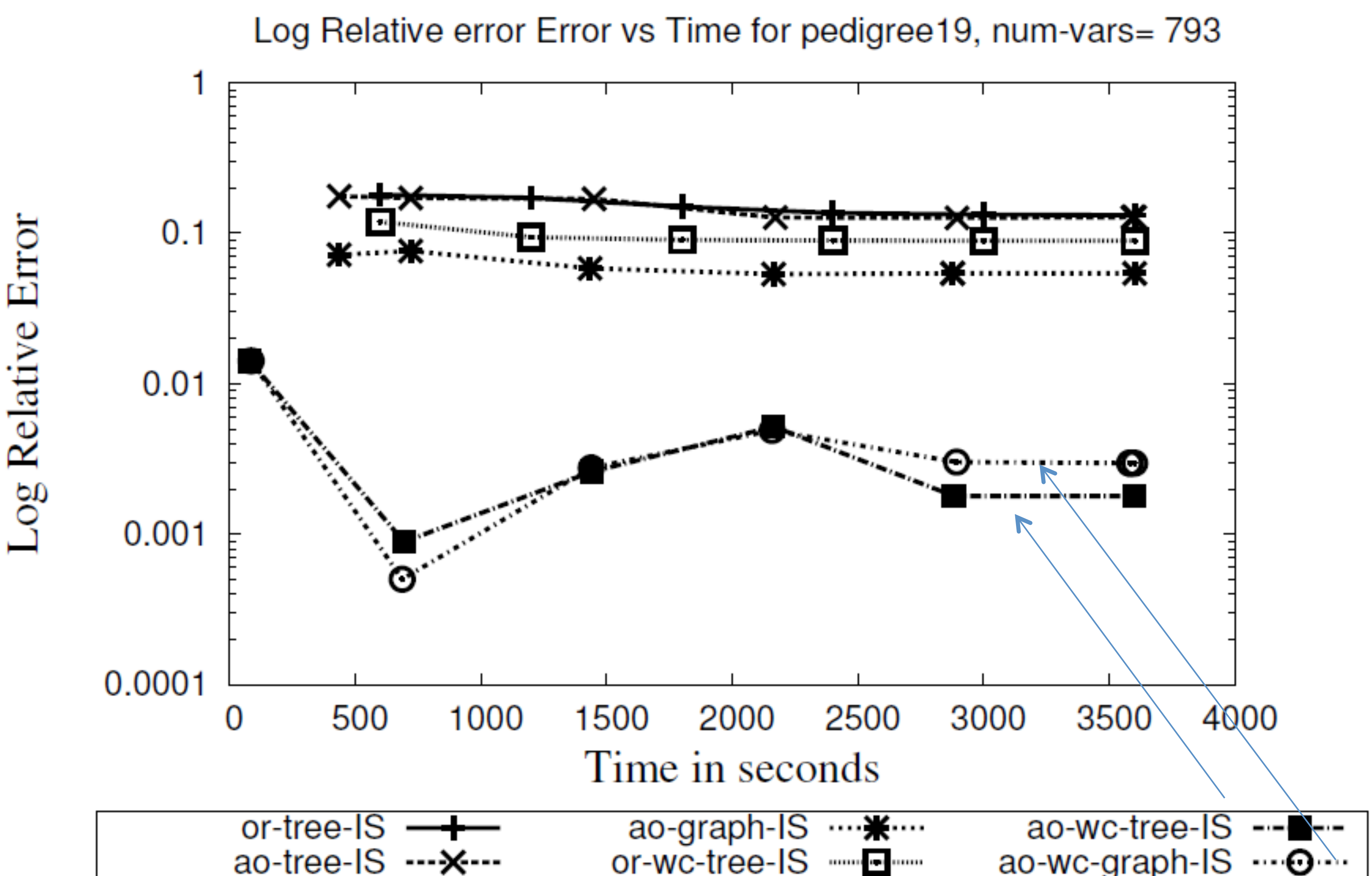
Results: Probability of Evidence

Linkage instances (UAI 2006 evaluation)

Problem	$\langle n, k, E, t^*, c \rangle$	Exact	or-tree-IS Δ	ao-tree-IS Δ	ao-graph-IS Δ	or-wc-tree-IS Δ	ao-wc-tree-IS Δ	ao-wc-graph-IS Δ
BN_69.uai	$\langle 777, 7, 78, 47, 59 \rangle$	5.28E-54	2.26E-02	2.46E-02	2.43E-02	2.42E-02	2.34E-02	**4.22E-03**
BN_70.uai	$\langle 2315, 5, 159, 87, 98 \rangle$	2.00E-71	6.32E-02	7.25E-02	5.12E-02	8.18E-02	5.36E-02	**2.62E-02**
BN_71.uai	$\langle 1740, 6, 202, 70, 139 \rangle$	5.12E-111	6.74E-02	5.51E-02	2.35E-02	8.58E-02	**9.46E-03**	1.21E-02
BN_72.uai	$\langle 2155, 6, 252, 86, 88 \rangle$	4.21E-150	3.19E-02	4.61E-02	2.46E-03	6.12E-02	**1.41E-03**	2.63E-03
BN_73.uai	$\langle 2140, 5, 216, 101, 149 \rangle$	2.26E-113	1.18E-01	1.12E-01	4.55E-02	1.58E-01	**3.54E-02**	3.95E-02
BN_74.uai	$\langle 749, 6, 66, 45, 72 \rangle$	3.75E-45	5.34E-02	4.31E-02	2.87E-02	8.08E-02	2.83E-02	**2.76E-02**
BN_75.uai	$\langle 1820, 5, 155, 92, 131 \rangle$	5.88E-91	4.47E-02	8.15E-02	4.73E-02	7.28E-02	4.20E-02	**7.60E-03**
BN_76.uai	$\langle 2155, 7, 169, 64, 239 \rangle$	4.93E-110	1.07E-01	1.39E-01	6.95E-02	1.13E-01	5.03E-02	**2.26E-02**
BN_77.uai	$\langle 1020, 9, 135, 22, 97 \rangle$	6.88E-79	1.06E-01	9.38E-02	8.26E-02	1.24E-01	6.75E-02	**3.27E-02**

Time Bound: 1hr


Log Relative error Error vs Time for BN_76, num-vars= 2155
Log Relative Error
1
0.1
0.01
0
500
1000
1500
2000
2500
3000
3500
4000
Time in seconds
or-tree-IS
ao-tree-IS
ao-graph-IS
or-wc-IS
ao-wc-tree-IS
ao-wc-graph-IS


Log Relative error Error vs Time for pedigree19, num-vars= 793
Log Relative Error
1
0.1
0.01
0.001
0.0001
0
500
1000
1500
2000
2500
3000
3500
4000
Time in seconds
or-tree-IS
ao-tree-IS
ao-graph-IS
or-wc-tree-IS
ao-wc-tree-IS
ao-wc-graph-IS

Outline

- Background: Bayesian networks
- Importance Sampling
- AND/OR search space for grpahical Models
- AND/OR Sampling: exploiting structure
- AND/OR+w-cutset sampling
- UAI 2010 experience

2010 Approximate Inference Evaluation (Gal Elidan, Amir Globerson)

Tasks:

- MAR: univariate marginals
- PR: partition function (probability of evidence)
- MPE: most probably explanation

Time frames:

- Lightning speed: 20 seconds
- Coffee break: 20 minutes
- Lunch break: 1 hour (reduced from 2 hours)

Anonymous solvers (on request)

Network: The Big Picture

	PR	MAR	MPE	Total
20 sec	204	204	461	869
1200 sec	204	204	**532**	940
3600 sec	204	204	287	695
Total	612	612	1280	2504

Networks – by domain (1 hour)

Network	PR	MAR	MPE
CSP	8	8	55
Grids	20	20	40
Image Alignment			10
Medical Diagnosis	26	26	
Object Detection	96	96	92
Pedigree	4	4	
Protein Folding			21
Protein-Protein Interaction			8
Segmentation	50	50	50

Many thanks to: Kristian Kersting, Stephen Gould, Menachem Fromer, Ben Packer, Dan Geiger, Ariel Jaimovich, Farshid Moussavi

Leader-board Summary

Seconds	PR	MAR	MPE
20	Arthur Choi (UCLA)	Arthur Choi (UCLA)	Joris Mooij (Max Planck)
1200	Vibhav Gogate (UW+UCI)	Vibhav Gogate (UCI)	Thomas Schiex (INRA)
3600	Vibhav Gogate (UW+UCI)	Vibhav Gogate (UCI)	Joris Mooij (Max Planck)

Solver for PR (Gogate 2010)

0. Choose a good variable ordering (30 sec or random minfill)

1. IJGP-based IS
2. AOT w-cutset
3. SampleSearch (handles zeros)
4. Formula-based sampling (handles context specific independence , helped on ising and promedas only)
5. Rejection control and epsilon-correction (Replace very small probabilities (e.g., 0.0001) in the proposal distribution by epsilon (0.01), and normalize. Note that the proposal distribution is a Bayesian network.)

2 Solvers for MAR

Sampling-based MAR algorithm:

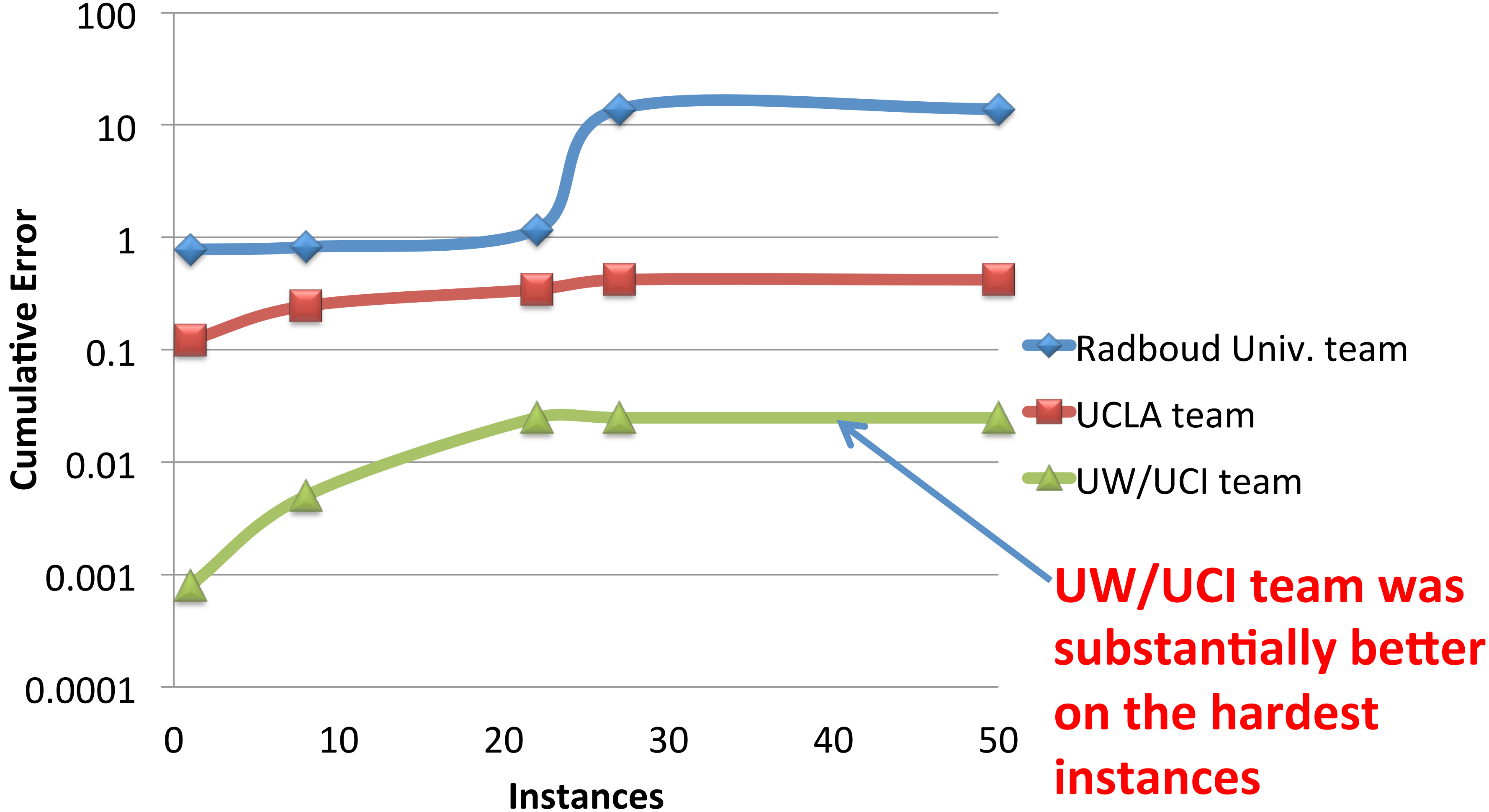
Step 1: Select values of w and i based on the problem and time-bound
Step 2: Select an ordering of variables by running min-fill 100 times or for 30 seconds whichever is earlier
Step 3: Select a w-cutset
Step 4: Run IJGP with i-bound=i
Step 5: If the problem has context specific independence and determinism
- Run formula-based SampleSearch with proposal constructed from the output of IJGP

Else:
- Run IJGP-sampling

Anytime IJGP algorithm:

Step 1: Select an initial value of i based on the problem and time-bound
Step 2: Select an ordering of variables by running min-fill 100 times or for 30 seconds whichever is earlier
Step 3: For j = i to treewidth
Run IJGP with i-bound=j until 20 iterations or convergence
Output the result

UAI 2010 approximate inference challenge: Comparison on the hardest instances (20 mins)



Winning Teams

- (MAR) **IJGP** by Vibahv Gogate, Andrew Gefland, Natasha Flerova and Rina Dechter (UCI):
  Anytime iterative GBP based algorithm
- (PR) **Vgogate** by Vibahv Gogate, Pedro Domingos (UW),, Andrew Gefland and Rina Dechter (UCI):
  Formula based importance sampling
- (PR+MAR) **EDBP** by Arthur Choi, Adnan Darwiche, with support from Glen Lenker and Knot Pipatsrisawat (UCLA):
  Anytime BP based anytime thickening of structure
- (MAP) **libDAI** by Joris Mooij (Max Planck):
  junction tree, LBP/MP, double-loop GBP, Gibbs, decimation
- (MAP) **toulbar2** by Thomas Schiex et al (INRA)
  Anytime branch and bound weighted CSP solver

Software and UAI-2008-10 results

- AND/OR search algorithms
- Bucket-tree elimination
- Generalized belief propsagation
- Samplesearch sampling

are available at:

  - http://graphmod.ics.uci.edu/group
  - http://graphmod.ics.uci.edu/uai08/Evaluation/Report

Conclusions

- Exploit structure
  - R. Marinescu and R. Dechter: AND/OR Branch-and-Bound search for combinatorial optimization in graphical models. Artifficial Intelligence. 173(16- 17): 1457-1491 (2009)
  - R. Marinescu and R. Dechter: Memory intensive AND/OR search for combinatorial optimization in graphical models. Artifficial Intelligence. 173(16- 17): 1492-1524 (2009)
  - V. Gogate and R. Dechter. "SampleSearch: Importance Sampling in presence of Determinism." Artificial Intelligence, 2011
  - R. Mateescu, K. Kask, V. Gogate, and R. Dechter. "Join-Graph Propagation Algorithms." Journal of Artificial Intelligence Research (JAIR) 37 (2010) 279-328
  - Bozhena Bidyuk, Rina Dechter and Emma Rollon. "Active Tuples-based Scheme for Bounding Posterior Beliefs" Journal of Artificial Intelligence Research (JAIR), (2010) 1-38
  - V. Gogate and R. Dechter "Importance Sampling based Estimation over AND/OR Search Spaces for Graphical Models" accepted to Artificial Intelligence, Forthcoming.

Thank you