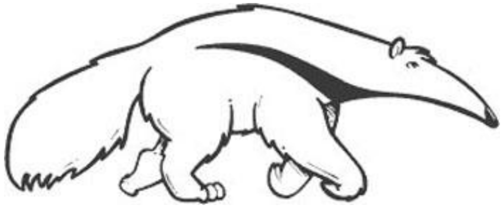
Tractable Islands Revisited

Rina Dechter
Information and Computer Sciences
UC-Irvine



In Search of Tractable Islands*

Rina Dechter

University of California

Irvine, California

*Supported in part by NSF grant IRI-9157936, by Air Force Office of Scientific Research, AFOSR 900136, Toshiba of America and Xerox Palo Alto Research center

Tractable Islands: Tasks and Methods

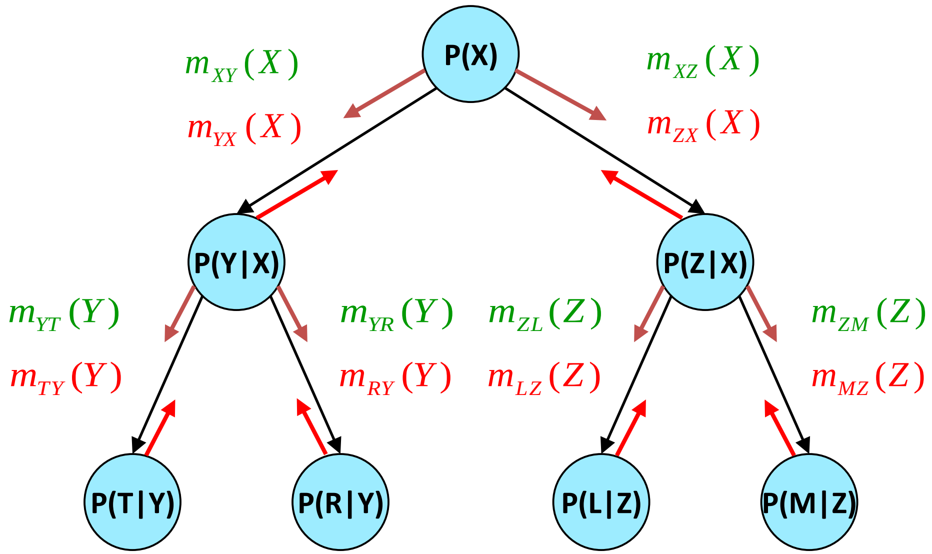
Bucket-elimination: Time and space O(n k^{w}) (dechter 1999)
Cutset-conditioning: is O(n $k^{\#c}$) time, linear memory, leading to anytime schemes

	Tasks / Tractable Islands	CSP/SAT	Optimization (MAX)	Weighted Counting (Sum)	MMAP (Max-Sum)	MEU (MAX-SUM)
Graph based	Trees (w=1)	☺	☺	☺	☺	☺
Graph based	W-trees	☺	☺	☺	☺	☺
Graph based	Cutset-trees	☺	☺	☺	☺	☺
Language-based	2-SAT	☺				
Language-based	Horn	☺				
Language-based	Tight domains-scopes	☺				
Mixed L+G	Row-convexity	☺				
Mixed L+G	Sub-modular	?	☺			

Tree-solving is easy

Belief updating (sum-prod)

CSP – consistency (projection-join)



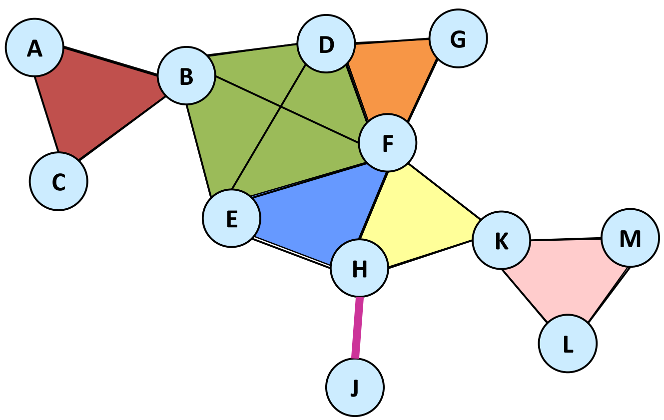
MPE (max-prod)

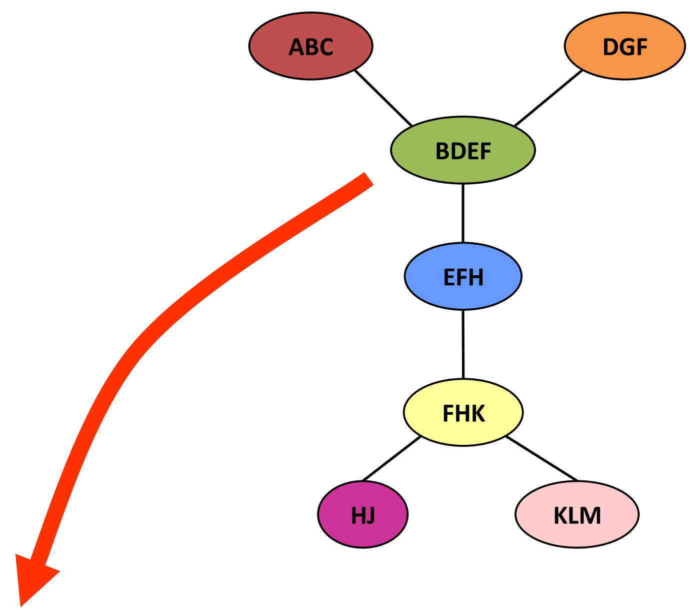
#CSP (sum-prod)

Trees are processed in linear time and memory

Traveling to trees via Inference

Traveling to trees by clustering
Distance: n exp(w)



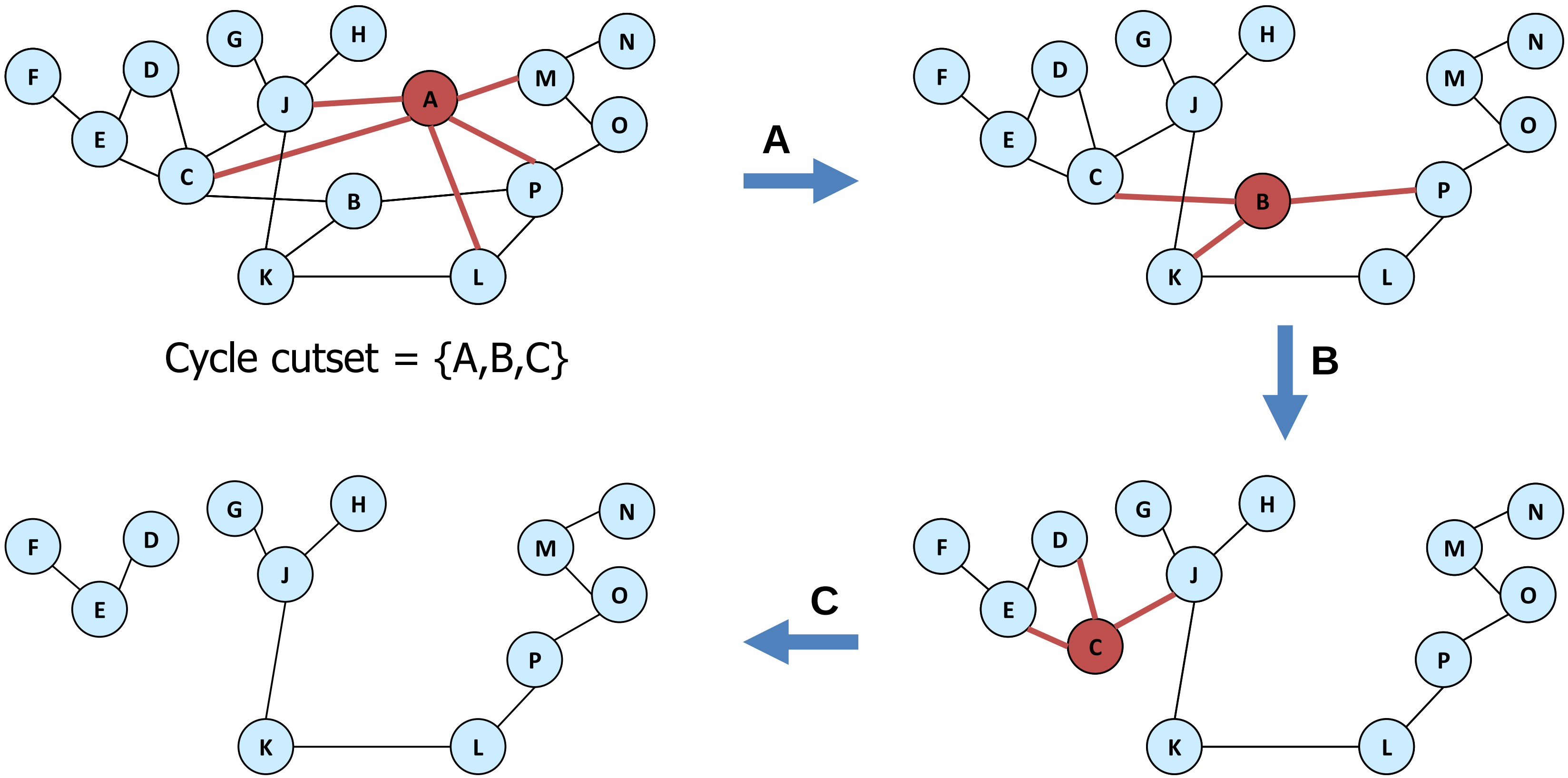


Inference algorithm:
Time: exp(tree-width)
Space: exp(tree-width)

treewidth = 4 - 1 = 3
treewidth = (maximum cluster size) - 1

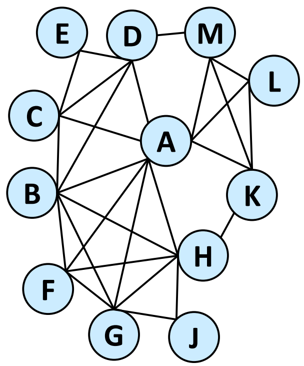
Traveling to trees via Conditioning

Traveling to trees by conditioning
Distance: exp(cycle-cutset)

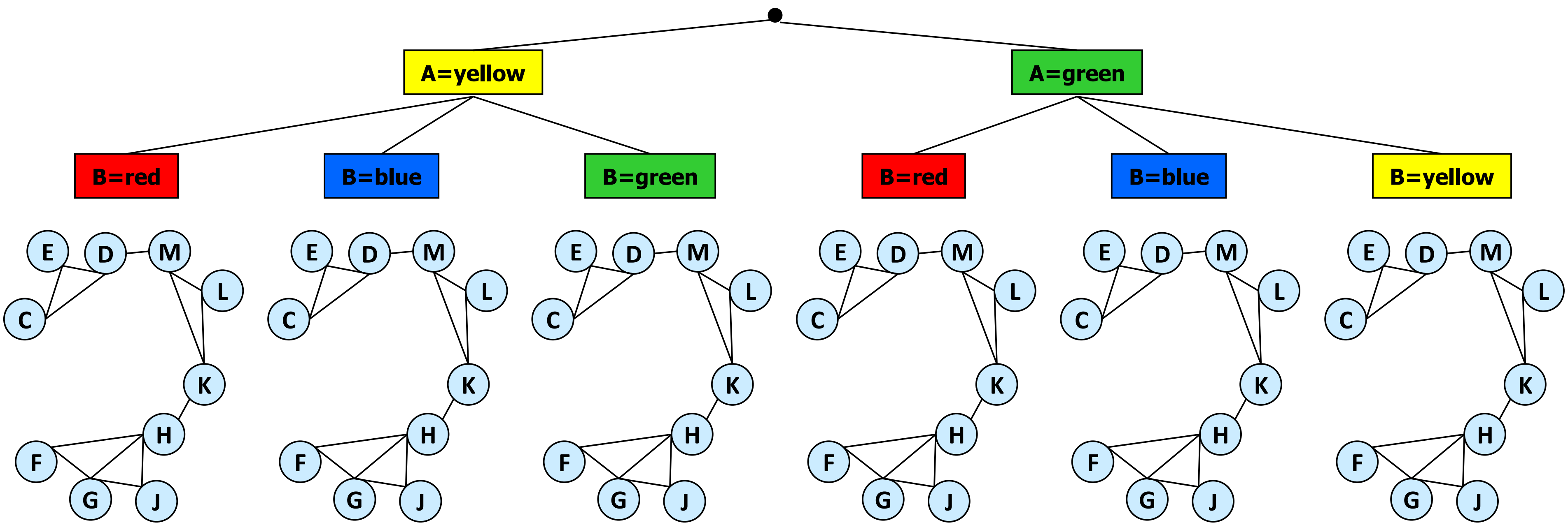


Traveling by Inference and Conditioning
w-Cutset: Balancing Memory and Time

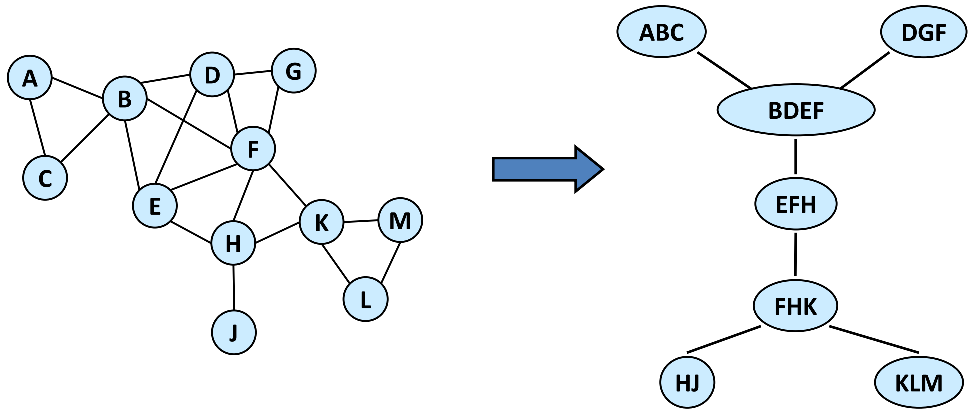
Graph Coloring problem



- Inference may require too much memory
- **Condition** on some of the variables
- W-cutset. Time exp(w+c_w), memory exp(w)

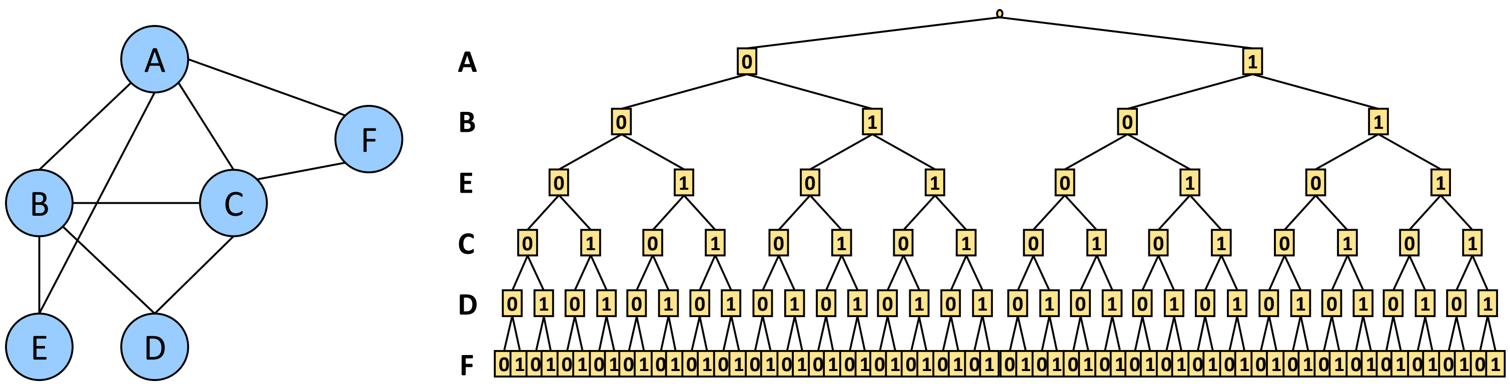


Bird's-eye View of Exact Algorithms



Inference

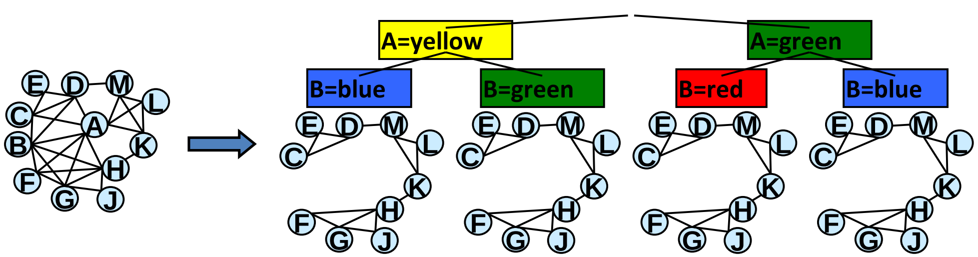
exp(w*) time/space



Search

Exp(w*) time

O(w*) space



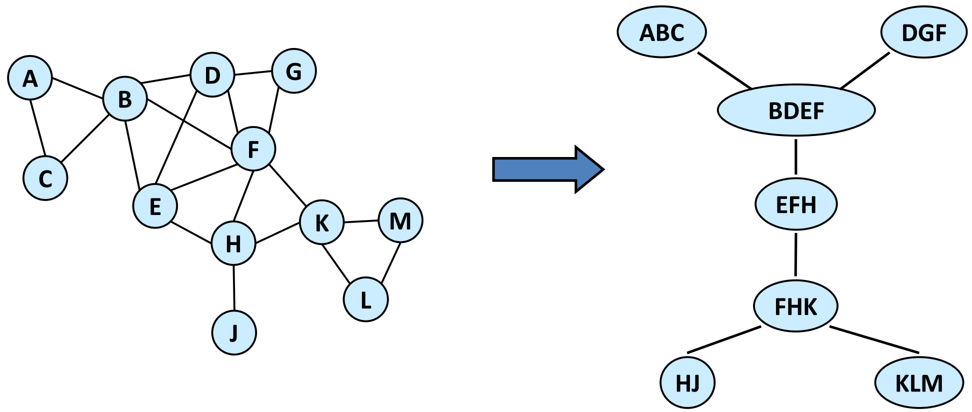
Search+inference:

Space: exp(q)

Time: exp(q+c(q))

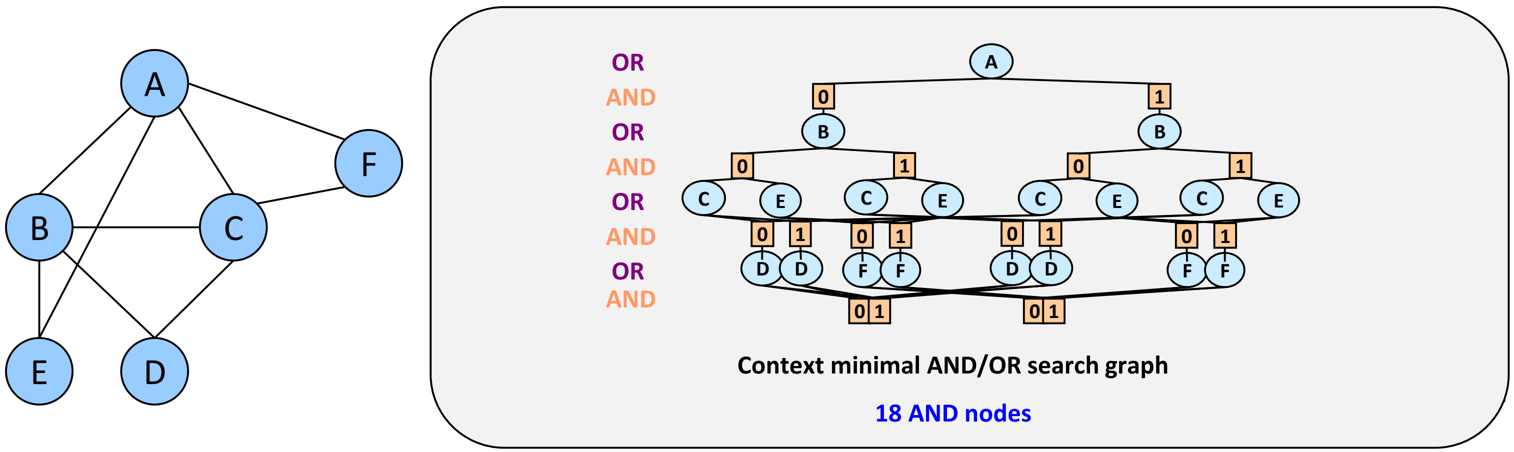
q: user controlled

Bird's-eye View of Exact Algorithms



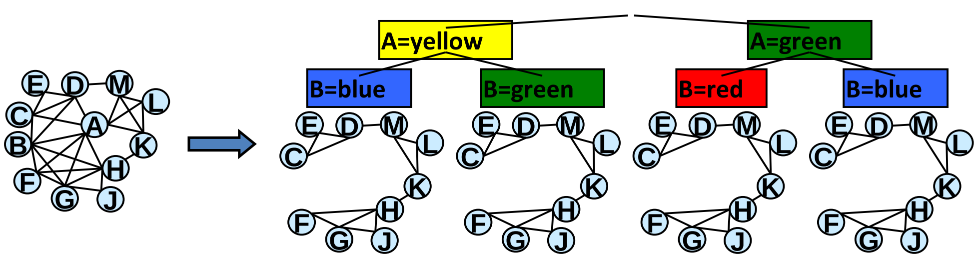
Inference

exp(w*) time/space



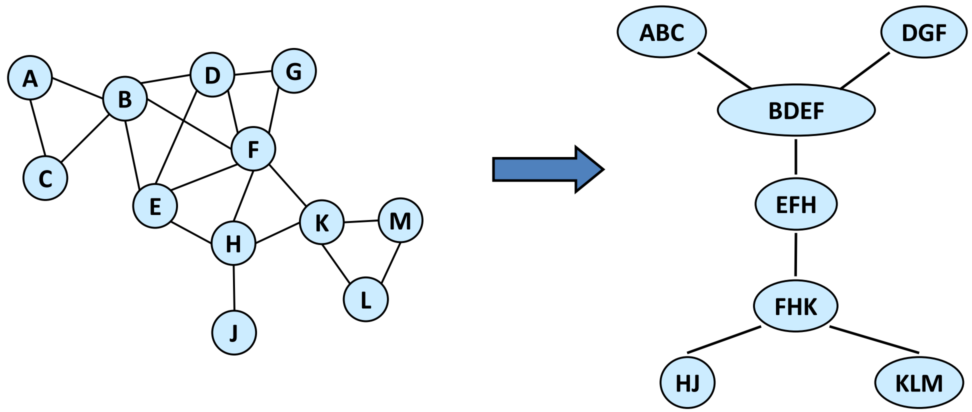
Search

Exp(w*) time
O(w*) space



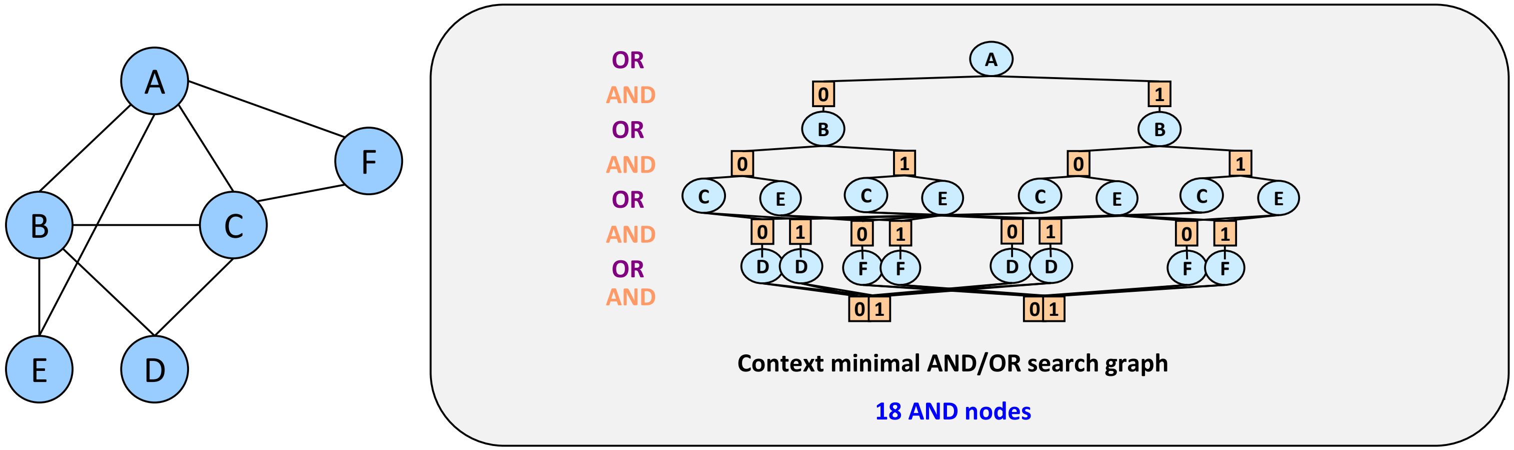
Search+inference:
Space: exp(q) q: user
Time: exp(q+c(q)) controlled

Bird's-eye View of Approximate Algorithms



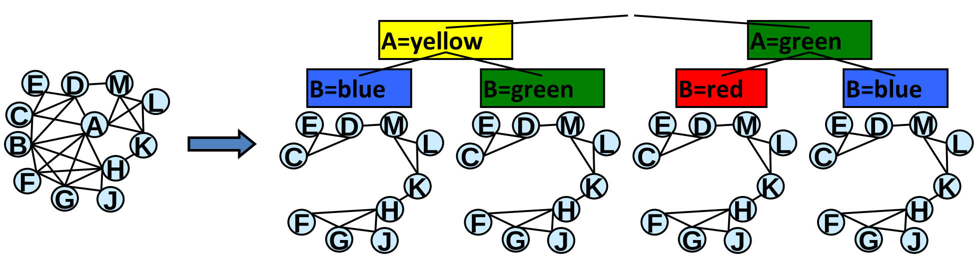
Inference

Bounded Inference



Search

Sampling



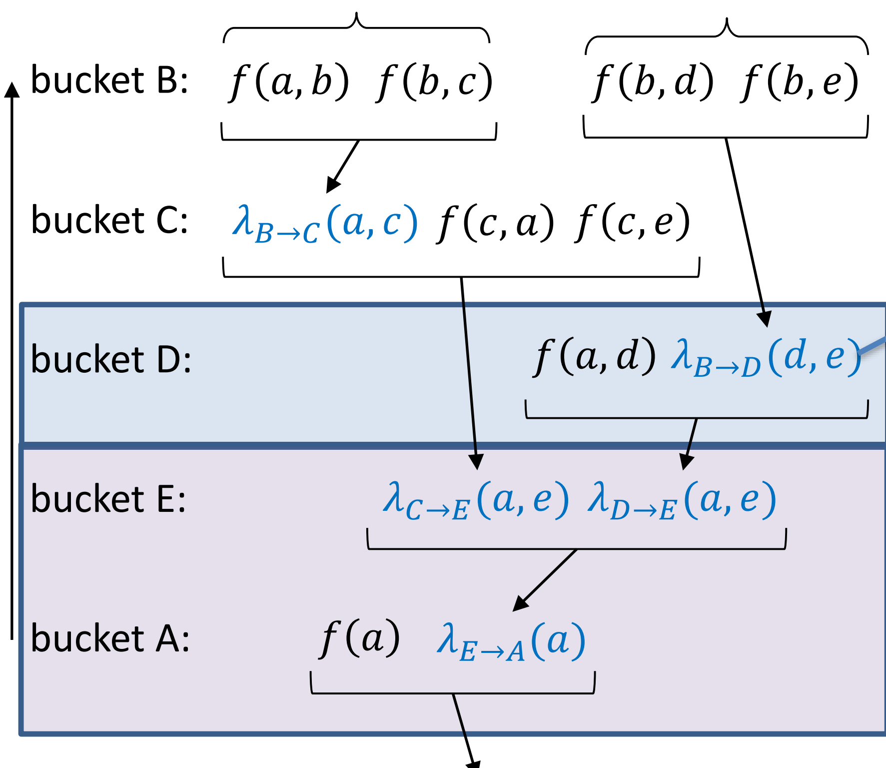
Search + inference:

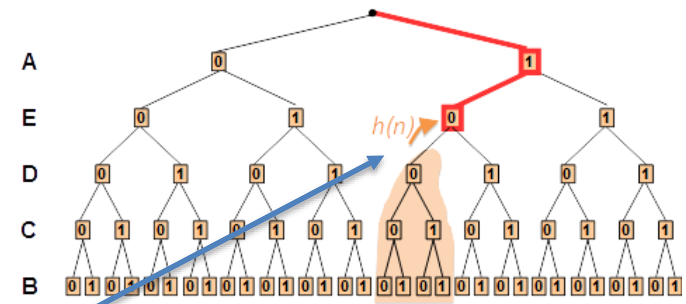
Sampling + bounded inference

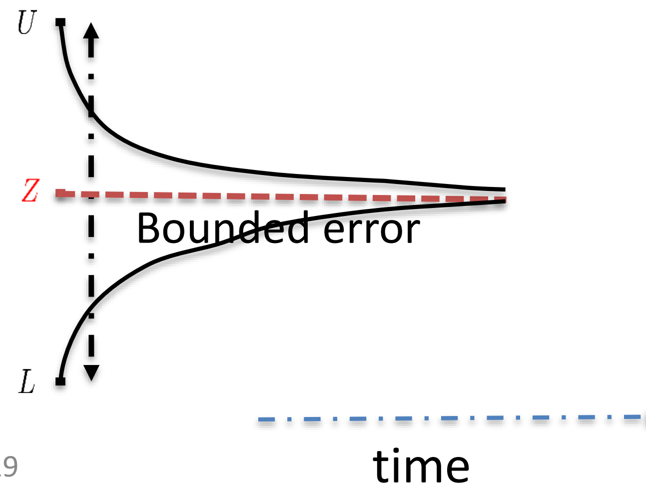
Our approach:
Use tractable islands in Reasoning, rather than in Modeling

Anytime Bounds via Tree Search

- Desiderata, meaningful confidence interval, responsive, complete
- Winning frameworks: search, or sampling guided by heuristics generated via tractable islands.

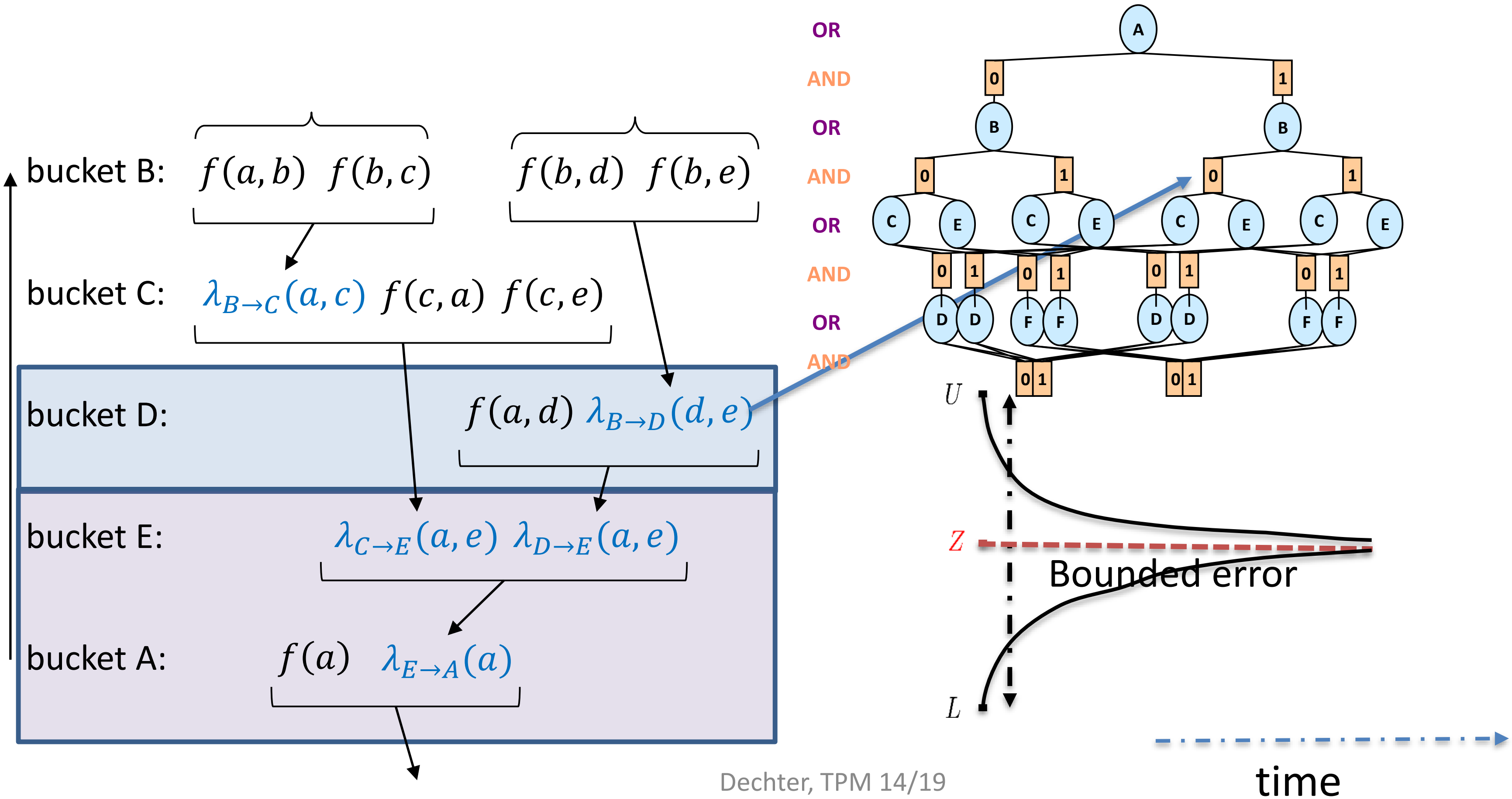






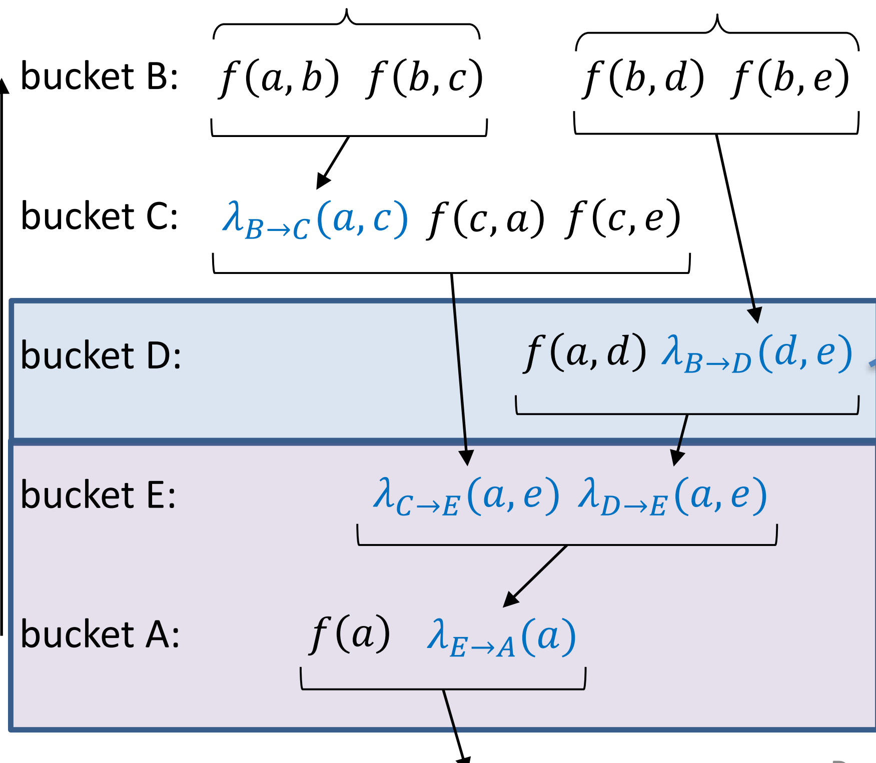
Anytime Bounds via Graph Search

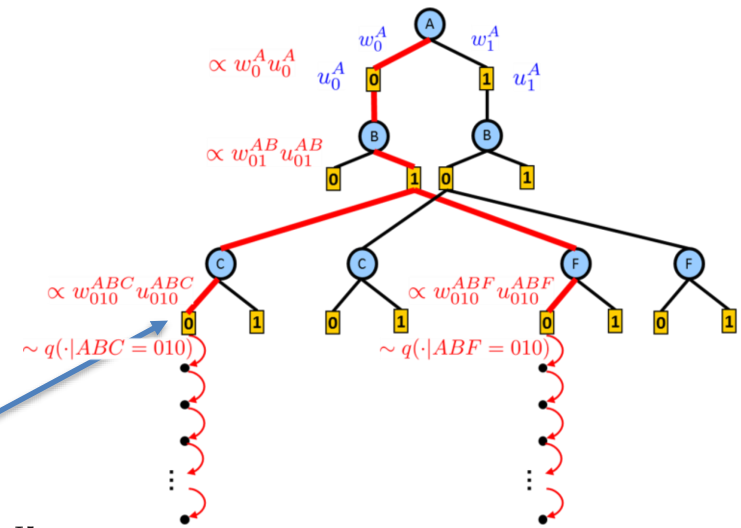
- Desiderata, meaningful confidence interval, responsive, complete
- Winning frameworks: search, or sampling guided by heuristics generated via tractable islands.

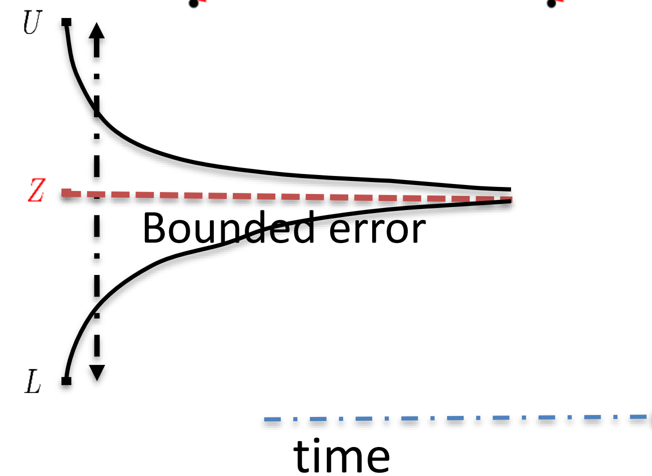


Anytime Bounds via Sampling

- Desiderata, meaningful confidence interval, responsive, complete
- Winning frameworks: search, or sampling guided by heuristics generated via tractable islands.

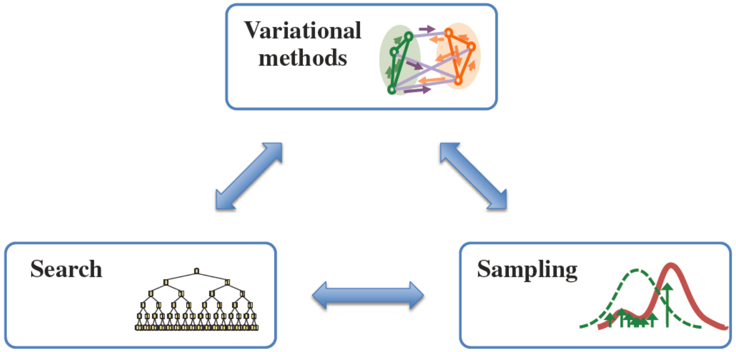


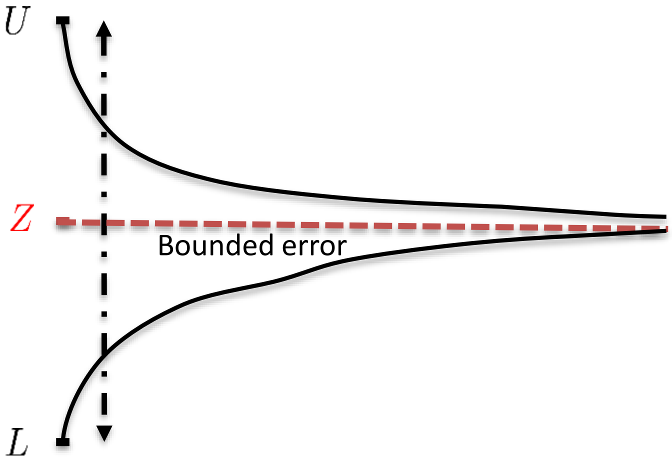




Outline

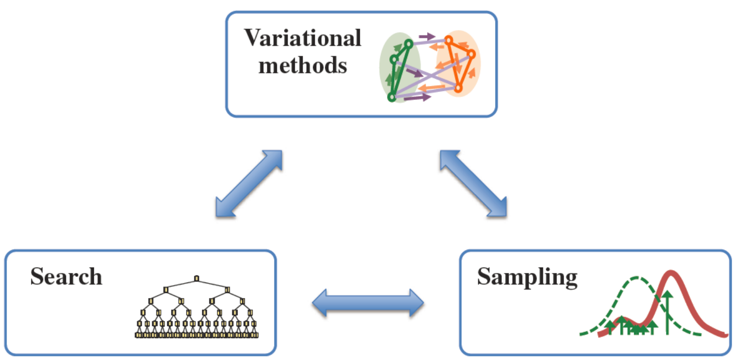
- Graphical models, The Marginal Map task
- AND/OR search spaces
- Variational bounds as search heuristics
- Combining methods: Heuristic Search for Marginal Map
- Combining methods: Sampling
- Conclusion

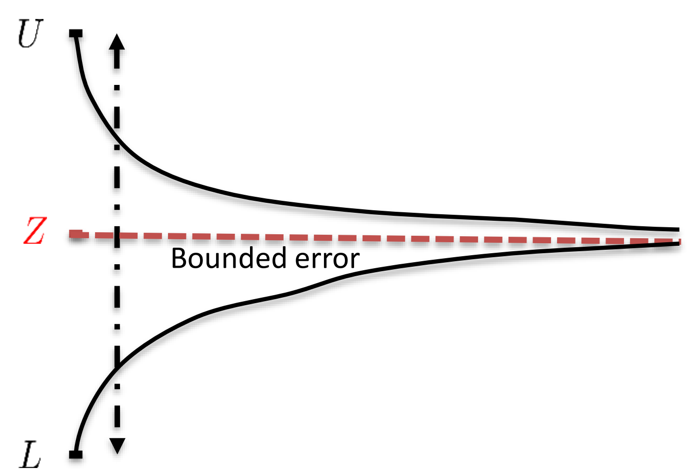




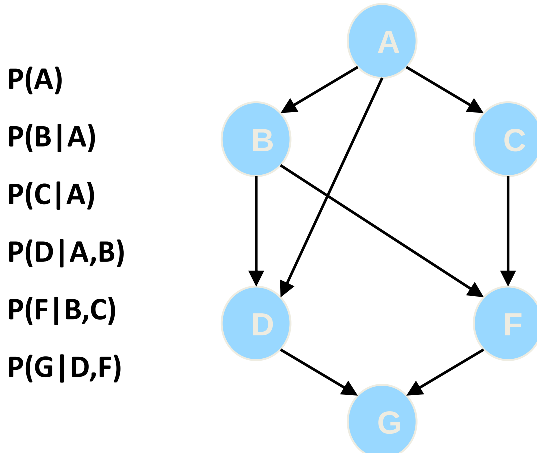
Outline

- Graphical models, The Marginal Map task
- AND/OR search spaces
- Variational bounds as search heuristics
- Combining methods: Heuristic Search for Marginal Map
- Combining methods: Sampling
- Conclusion

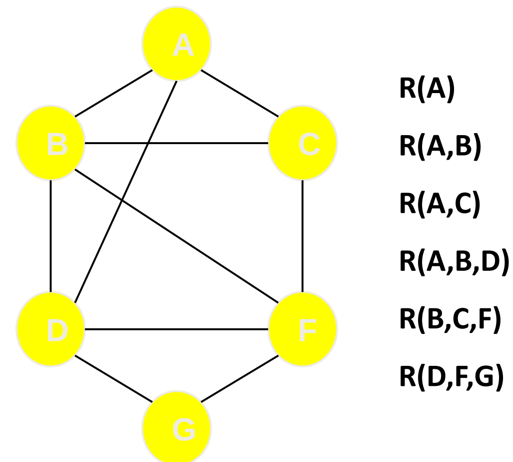




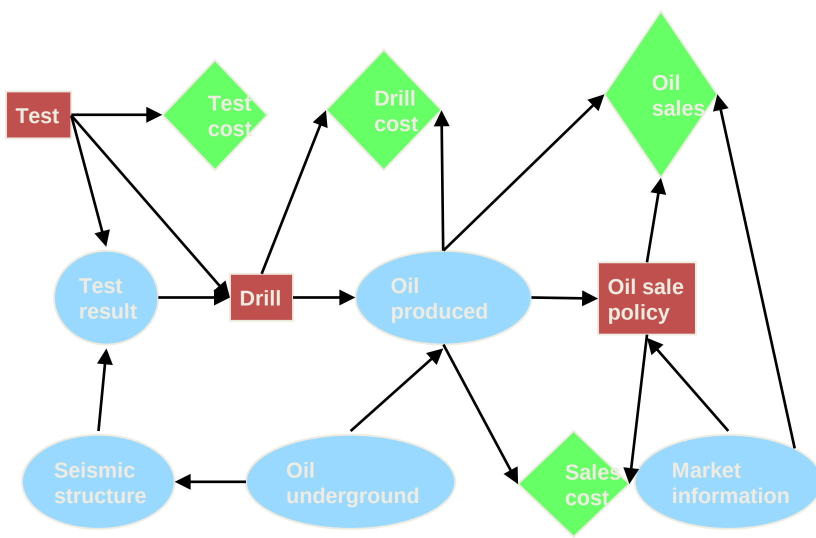
Graphical Models



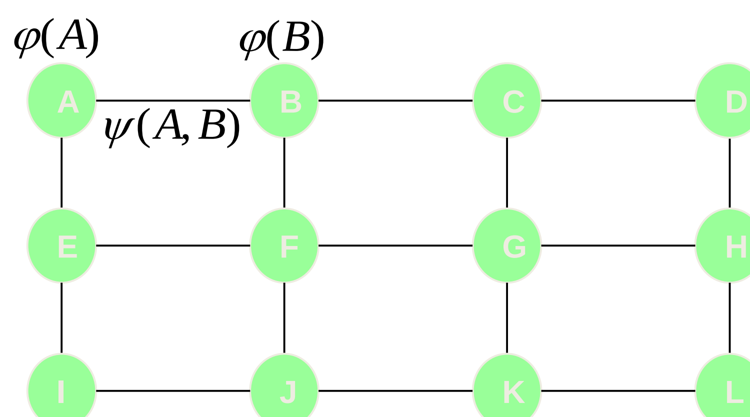
a) Belief network (directed)



b) Constraint network (undirected)



c) Influence diagram

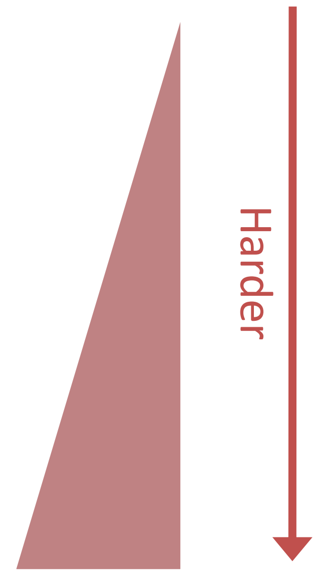


d) Markov network

Probabilistic Reasoning Problems

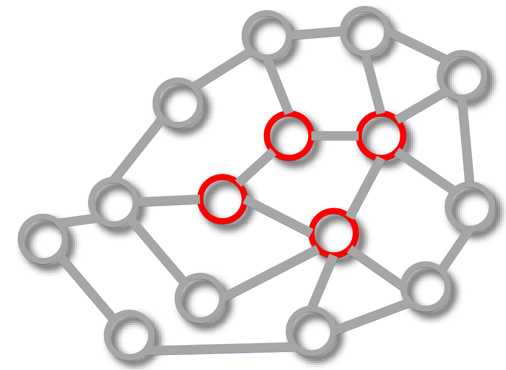
- Tasks:

Max-Inference (most likely config.)	$f(\mathbf{x}^*) = \max_{\mathbf{x}} \prod_{\alpha} f_\alpha(\mathbf{x}_\alpha)$
Sum-Inference (data likelihood)	$Z = \sum_{\mathbf{x}} \prod_{\alpha} f_\alpha(\mathbf{x}_\alpha)$
Mixed-Inference (optimal prediction)	$f(\mathbf{x}_M^*) = \max_{\mathbf{x}_M} \sum_{\mathbf{x}_S} \prod_{\alpha} f_\alpha(\mathbf{x}_\alpha)$



  - Combinatorial search / counting queries
  - Exact reasoning NP-complete (or worse)

Why Marginal MAP?



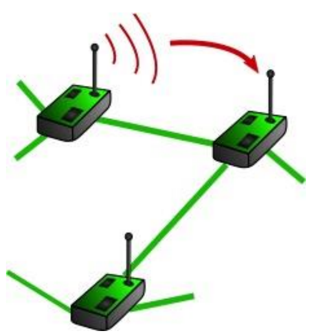
- Often, Marginal MAP is the "right" task:
  - We have a model describing a large system
  - We care about predicting the state of some part

- Example: decision making
  - Sum over random variables
  - Max over decision variables (specify action policies)

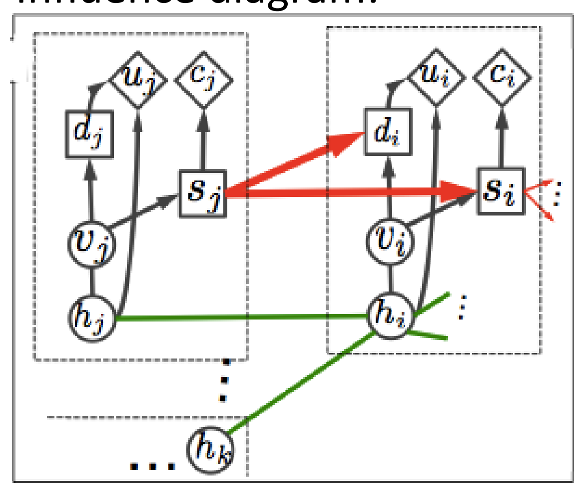
– Complexity: NP^{pp} complete

– Not necessarily easy on trees

Sensor network

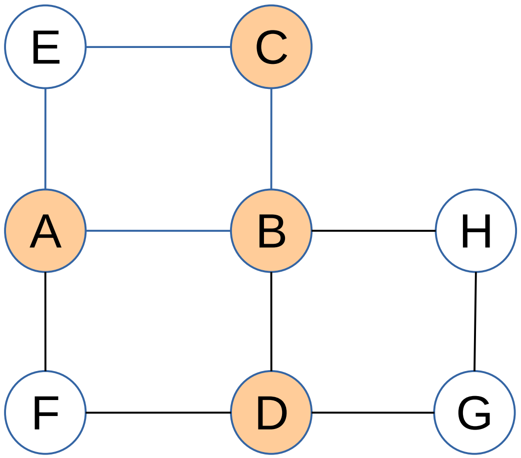


Influence diagram:



Marginal Map

primal graph



- Graphical Model: $\mathcal{M} = \langle \mathbf{X}, \mathbf{D}, \mathbf{F} \rangle$

  – variables $\mathbf{X} = \{X_1, \ldots, X_n\}$
  – domains $\mathbf{D} = \{D_1, \ldots, D_n\}$
  – functions $\mathbf{F} = \{f_1, \ldots, f_r\}$

$$P(\mathbf{X}) = \frac{1}{Z} \prod_j f_j$$

- Marginal MAP task:

$$\mathbf{X} = \mathbf{X}_M \cup \mathbf{X}_S$$

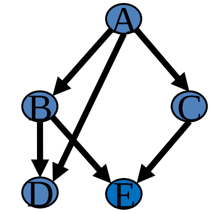
$$x_M^* = argmax_{X_M} \sum_{X_S} \prod_j f_j$$

$$\mathbf{X}_M = \{A, B, C, D\}$$

$$\mathbf{X}_S = \{E, F, G, H\}$$

Why is it harder? intuitively

Finding Marginals by Bucket Elimination



Algorithm *BE-bel* (Dechter 1996)

$$P(A \mid E=0) = \alpha \sum_{E=0,D,C,B} P(A) \cdot P(B \mid A) \cdot P(C \mid A) \cdot P(D \mid A,B) \cdot P(E \mid B,C)$$

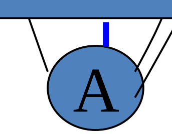
$\sum_b \prod$ ← Elimination operator

Time and space exponential in the induced-width / treewidth

$$\mathbf{O(n}k^{w^*+1}\mathbf{)}$$

bucket A: P(a) $\lambda_{E \to A}(a)$

(max clique size)

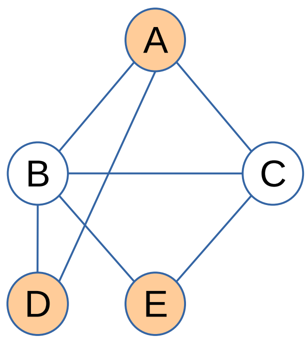


P(e=0)

P(a|e=0)

Why is MMAP Harder for Inference (BE)?

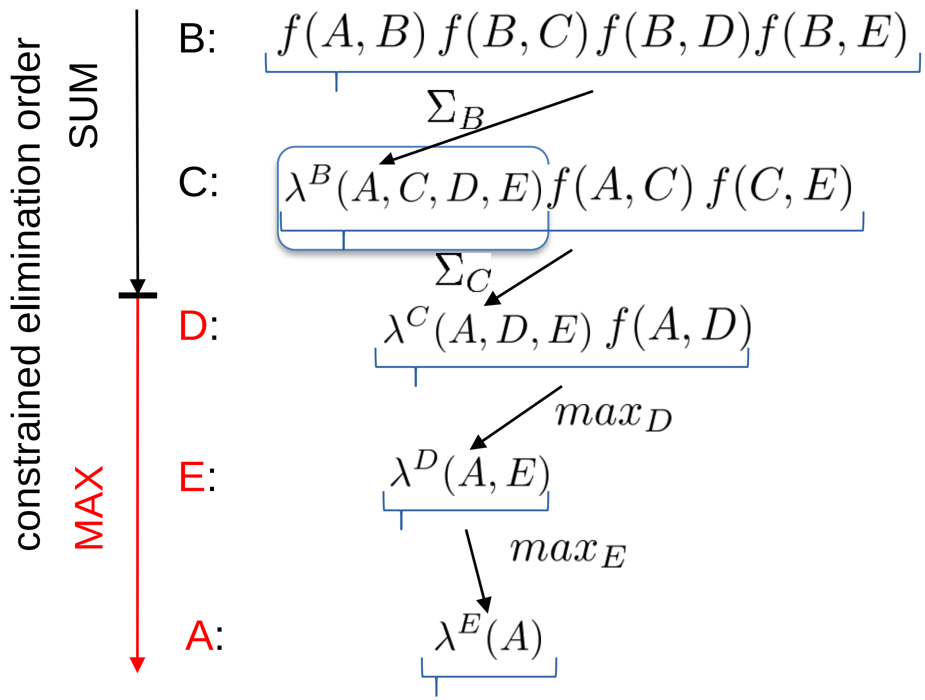
Let’s apply Bucket-elimination: Complexity is exponential in the *constrained* induced-width



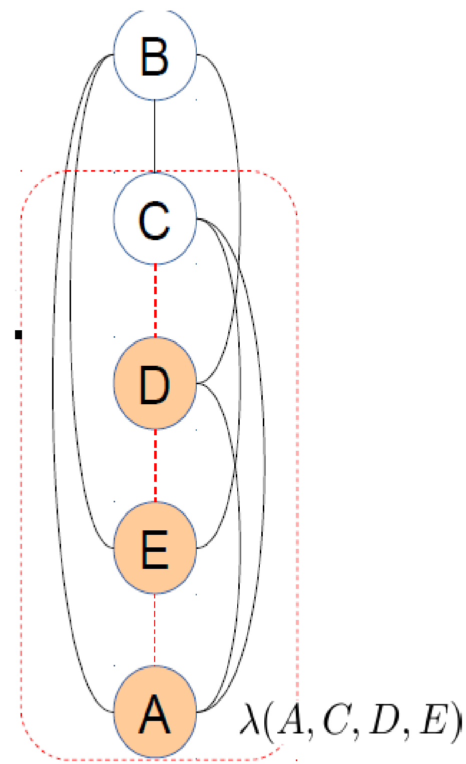
$$\mathbf{X}_M = \{A, D, E\}$$
$$\mathbf{X}_S = \{B, C\}$$

$$\max_{\mathbf{X}_M} \sum_{\mathbf{X}_S} P(\mathbf{X})$$

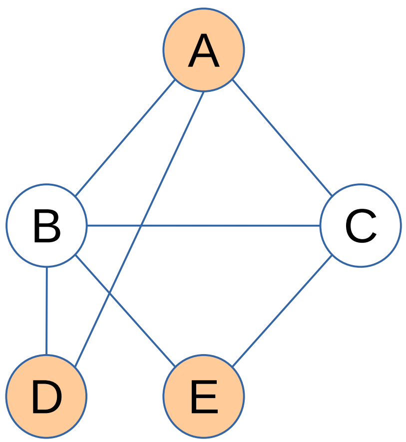
$$P(X) = \prod_j f_j$$



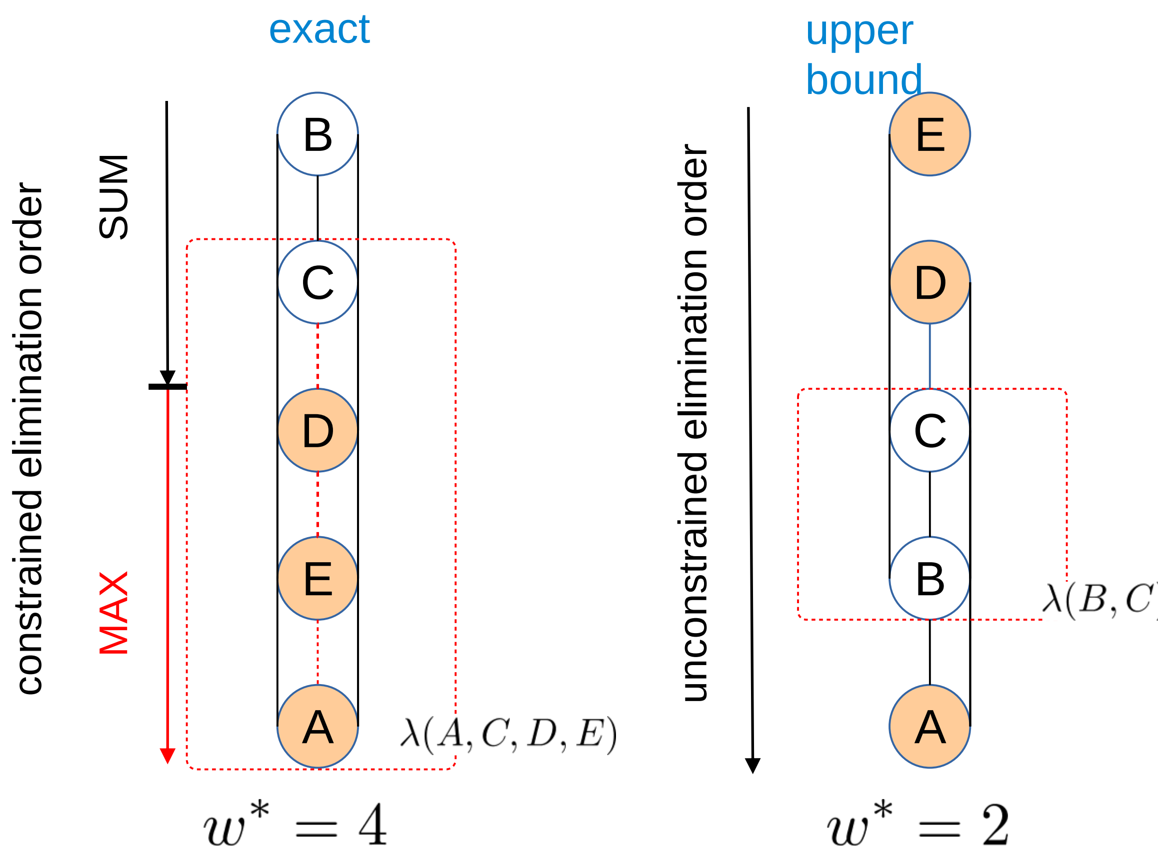
MAP* is the marginal MAP value



Why is MMAP Harder for Inference (BE)?



$$\mathbf{X}_M = \{A, D, E\}$$
$$\mathbf{X}_S = \{B, C\}$$



(Park & Darwiche, 2003)
(Yuan & Hansen, 2009)

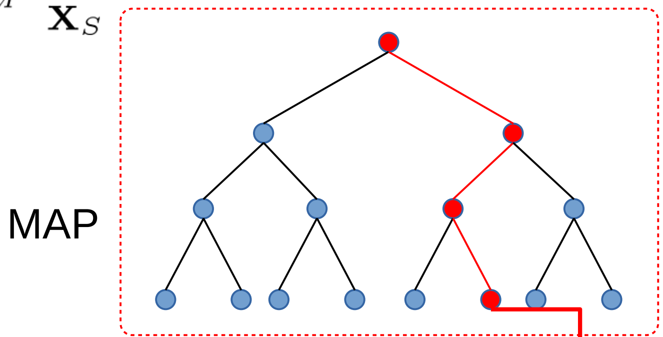
In practice, constrained induced is much larger!

$$\max_X \sum_Y \phi \leq \sum_Y \max_X \phi$$

Why is MMAP Harder for Search?

Brute-Force Search

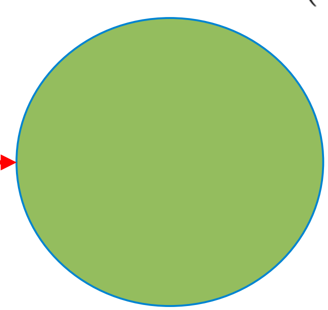
$$\max_{\mathbf{X}_M} \sum_{\mathbf{X}_S} P(\mathbf{X})$$



- Enumerate all full MAP assignments
- Evaluate each full MAP assignment
- Return the one with maximum cost

$$cost(\bar{x}_M) = \sum_{\mathbf{X}_S} \phi$$

$$\#P - complete$$



Evaluating a MAP assignment is hard!

Harder relative to optimization because induced-width is higher and evaluation of a configuration is higher

Harder relative to summation: higher induced-width

For anytime behavior we need conditioning
→ Search

AND/OR Search Spaces for Graphical Models

And, if possible, lets exploit structure in the search space as well.

Potential search spaces

A	B	f_1
0	0	2
0	1	0
1	0	1
1	1	4

A	C	f_2
0	0	3
0	1	0
1	0	0
1	1	1

A	E	f_3
0	0	0
0	1	3
1	0	2
1	1	0

A	F	f_4
0	0	2
0	1	0
1	0	0
1	1	2

B	C	f_5
0	0	0
0	1	1
1	0	2
1	1	4

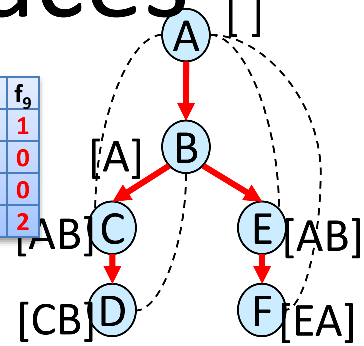
B	D	f_6
0	0	4
0	1	2
1	0	1
1	1	0

B	E	f_7
0	0	3
0	1	2
1	0	1
1	1	0

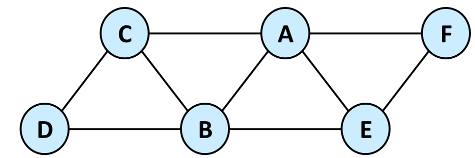
C	D	f_8
0	0	1
0	1	4
1	0	0
1	1	0

E	F	f_9
0	0	1
0	1	0
1	0	0
1	1	2

Pseudo-tree



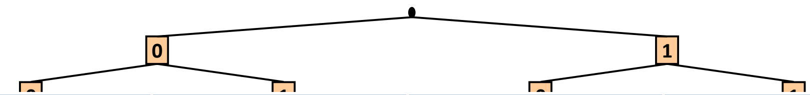
pseudo tree



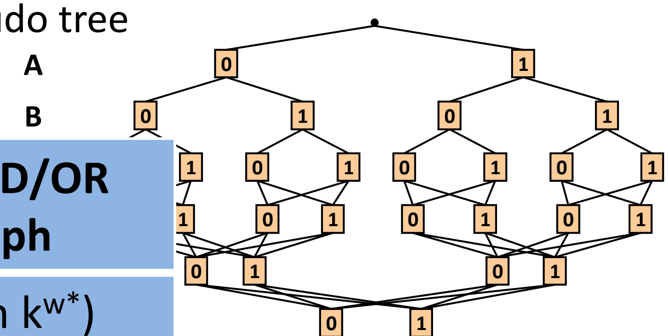
$$f(\mathbf{x}^*) = \max_{\mathbf{x}} \prod_{\alpha} f_\alpha(\mathbf{x}_\alpha) \qquad Z = \sum_{\mathbf{x}} \prod_{\alpha} f_\alpha(\mathbf{x}_\alpha)$$

$$f(\mathbf{x}_M^*) = \max_{\mathbf{x}_M} \sum_{\mathbf{x}_S} \prod_{\alpha} f_\alpha(\mathbf{x}_\alpha)$$

A
B
C
D
E
F

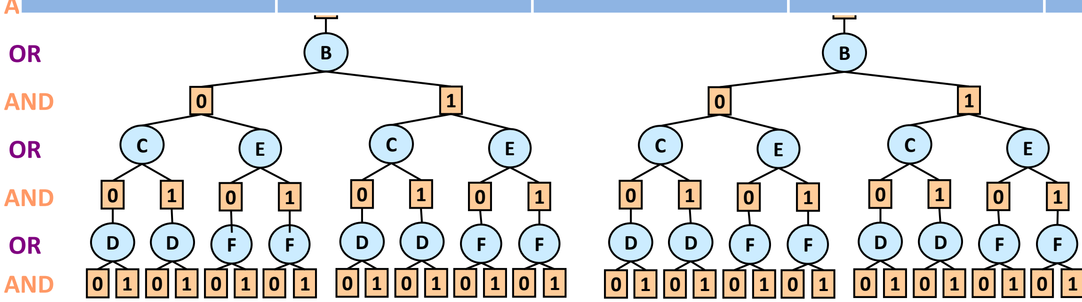


	OR tree	AND/OR tree	OR graph	AND/OR graph
time	$O(k^n)$	$O(nk^h)$	$O(n\, k^{pw^*})$	$O(n\, k^{w^*})$
memory	O(n)	O(n)	$O(n\, k^{pw^*})$	$O(n\, k^{w^*})$



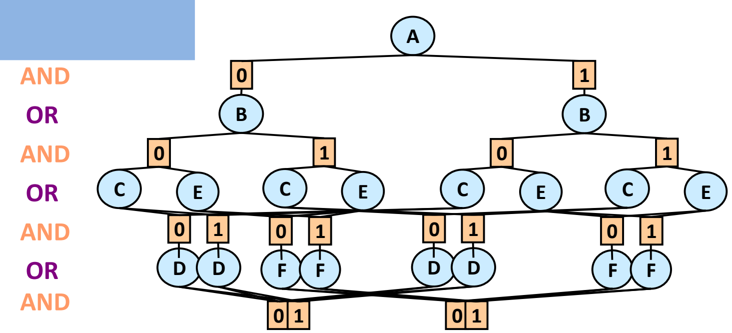
Context minimal OR search graph

28 nodes



Full AND/OR search tree

54 AND nodes



Context minimal AND/OR search graph

18 AND nodes

Any query can be computed over any of the search spaces

Cost of a Solution Tree

$P(E|A,B)$

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

$P(B|A)$

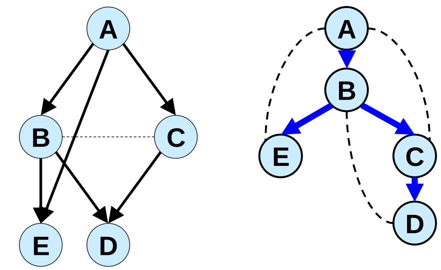
A	B=0	B=1
0	.4	.6
1	.1	.9

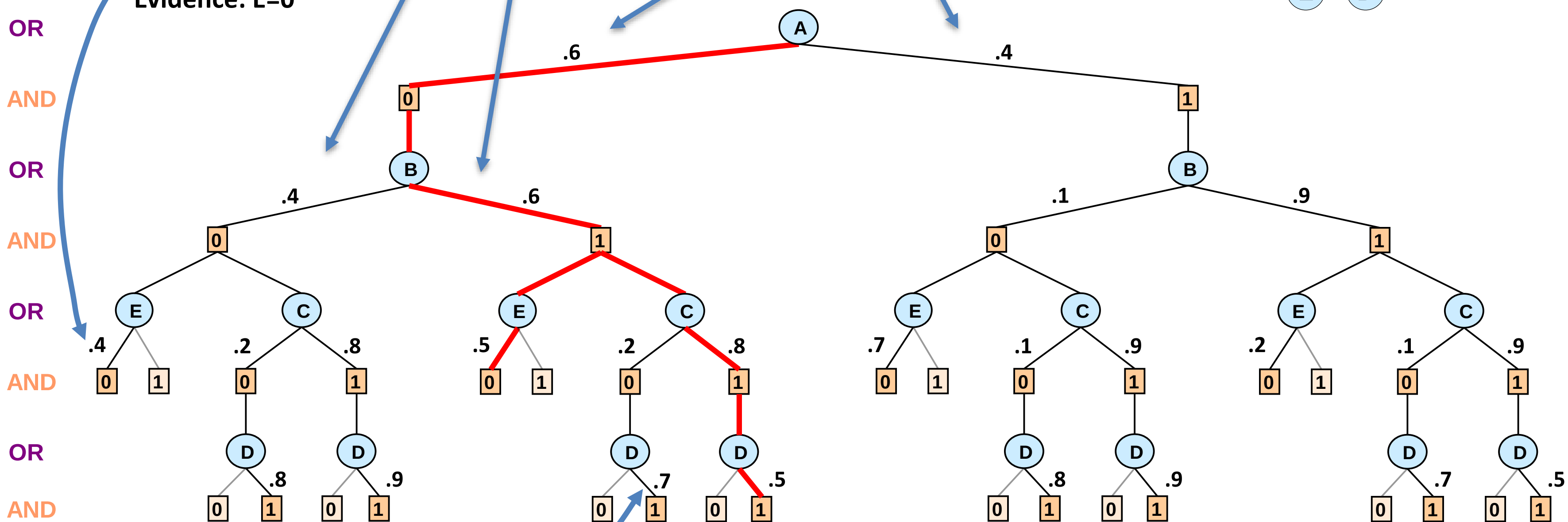
$P(C|A)$

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

$P(A)$

A	P(A)
0	.6
1	.4





$P(D|B,C)$

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

Cost of the solution tree: the product of weights on its arcs

Cost of (A=0,B=1,C=1,D=1,E=0) = $0.6 \cdot 0.6 \cdot 0.5 \cdot 0.8 \cdot 0.5 = 0.0720$

Value of a Node (e.g., Probability of Evidence)

$P(E \mid A, B)$

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

$P(B \mid A)$

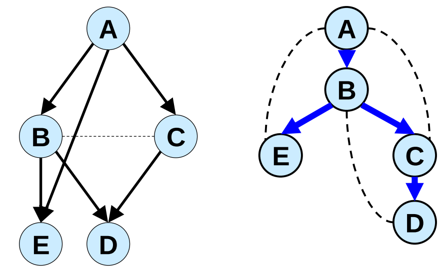
A	B=0	B=1
0	.4	.6
1	.1	.9

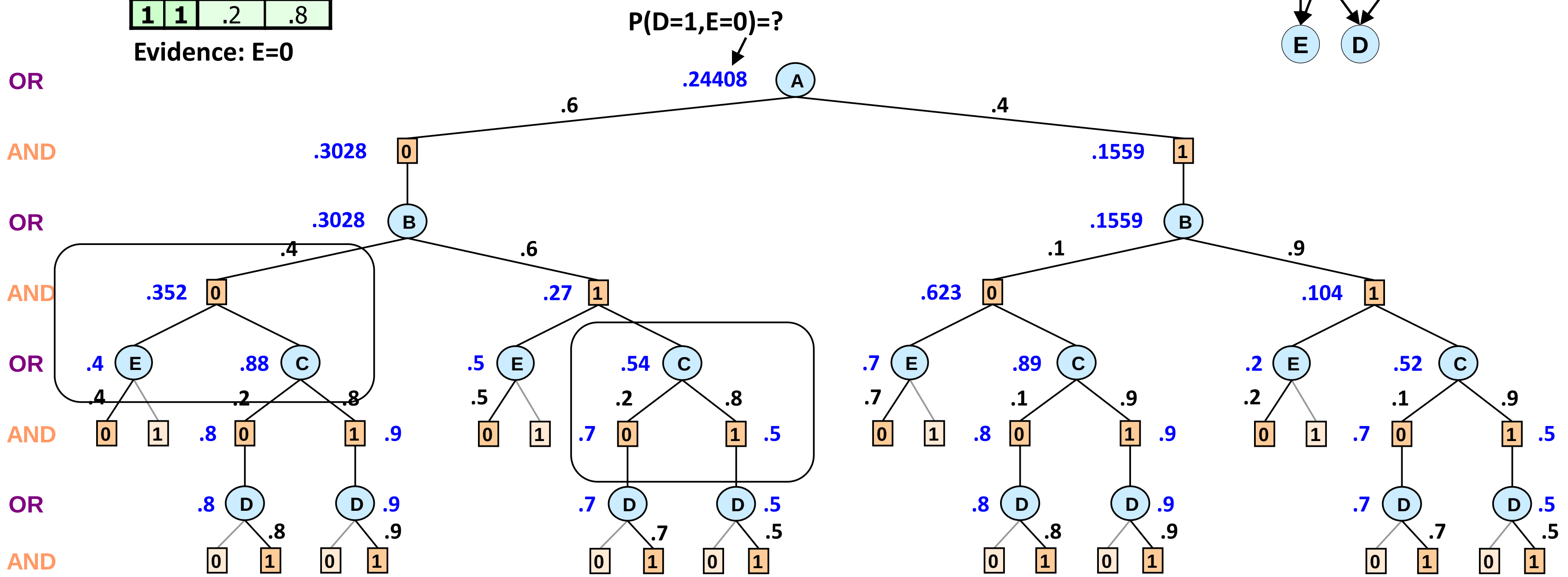
$P(C \mid A)$

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

Value of node = updated belief for sub-problem below

AND node: product $\prod_{n' \in children(n)} v(n')$

OR node: Marginalization by summation $\sum_{n' \in children(n)} w(n, n')\, v(n')$

Answering Queries: Sum-Product (Belief Updating)

$P(E|A,B)$

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

$P(B|A)$

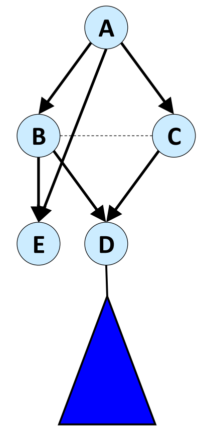
A	B=0	B=1
0	.4	.6
1	.1	.9

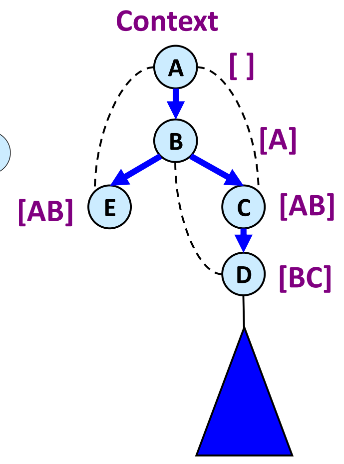
$P(C|A)$

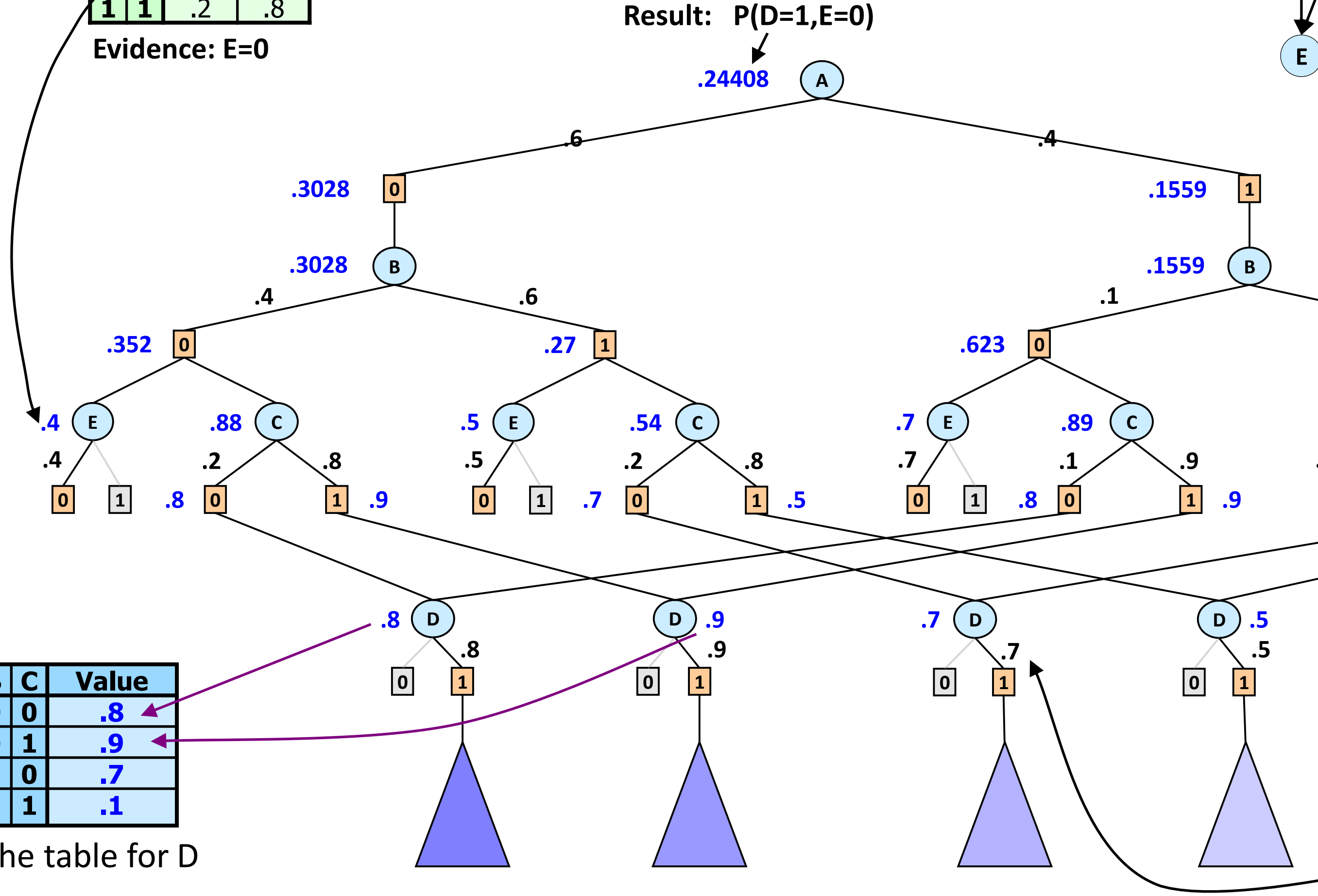
A	C=0	C=1
0	.2	.8
1	.7	.3

$P(A)$

A	P(A)
0	.6
1	.4







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

Cache table for D

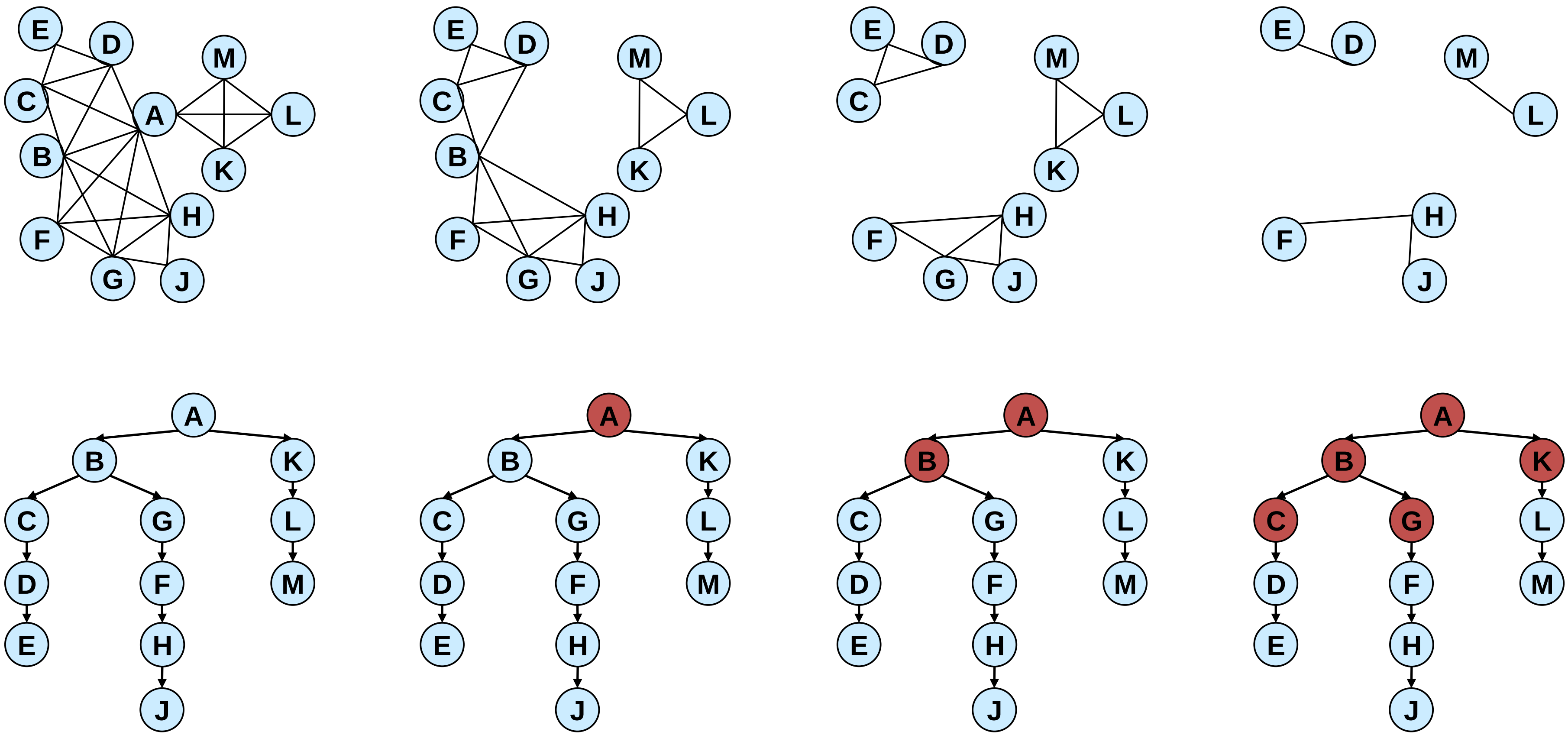
$P(D|B,C)$

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

AND/OR w-Cutset

Start from a pseudo-tree, terminate when reaching a cutset, and apply inference



A pseudo-tree

3-cutset
start pseudo tree

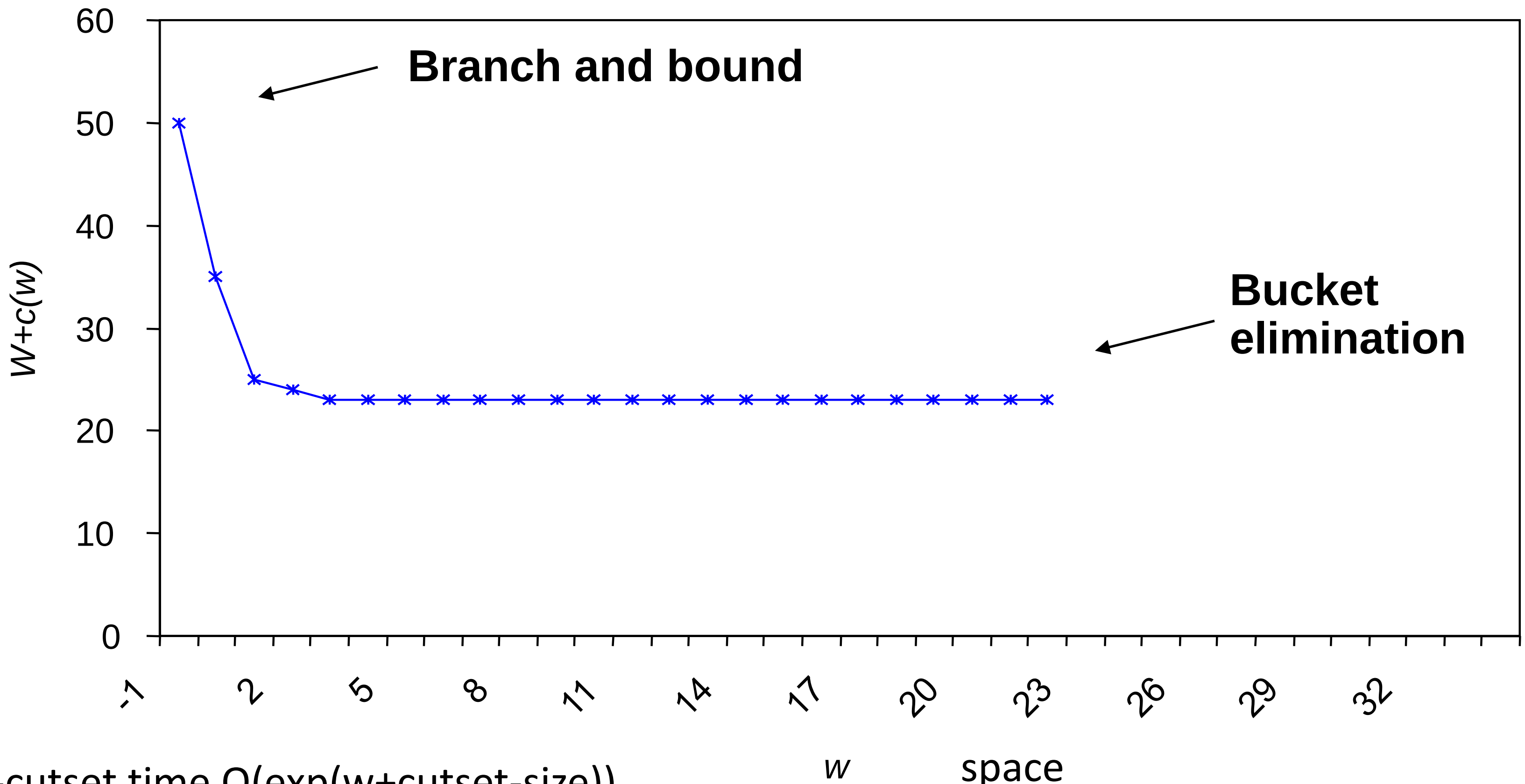
2-cutset
start pseudo tree

1-cutset
start pseudo tree

Time vs Space for w-cutset

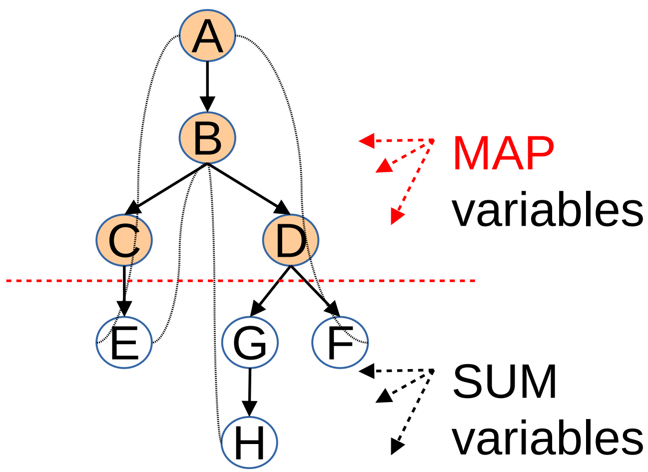
(Dechter and El-Fatah, 2000)
(Larrosa and Dechter, 2001)
(Rish and Dechter 2000)

- **Random Graphs (50 nodes, 200 edges, average degree 8, *w**≈23)**

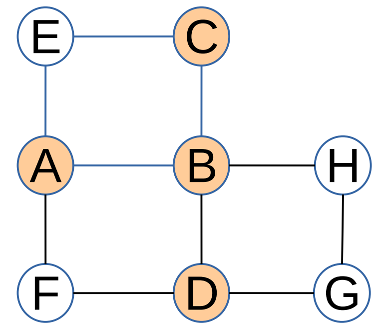


w-cutset time O(exp(w+cutset-size))
Space O(exp(w))

AND/OR search for Marginal MAP



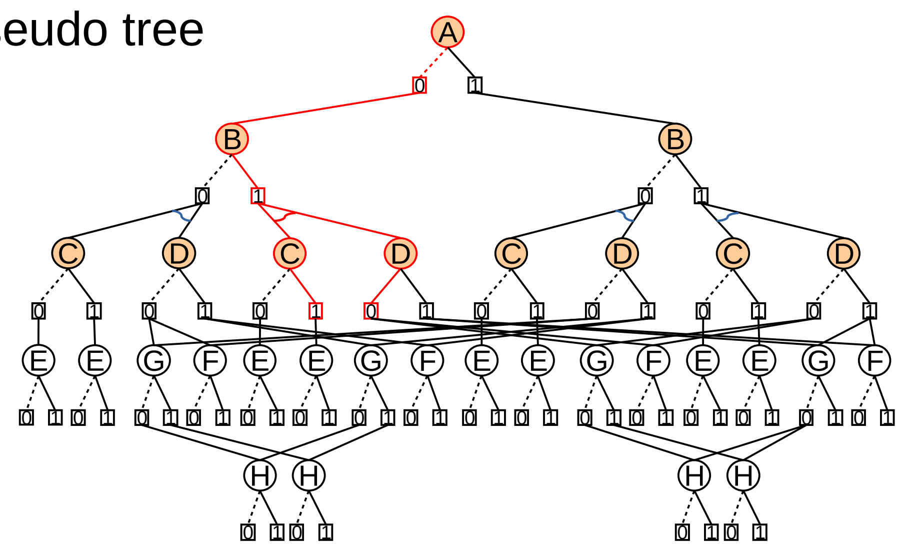
constrained
pseudo tree



primal graph

$X_M = \{A, B, C, D\}$

$X_S = \{E, F, G, H\}$

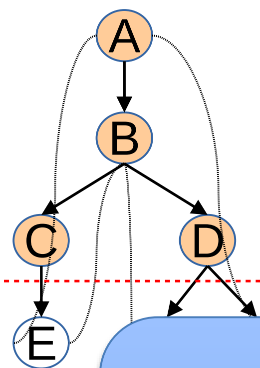


Node types

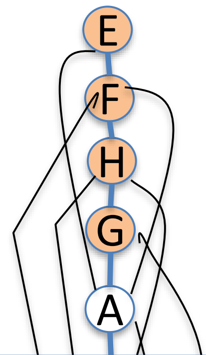
- OR (MAP): max
- OR (SUM): sum
- AND: multiplication

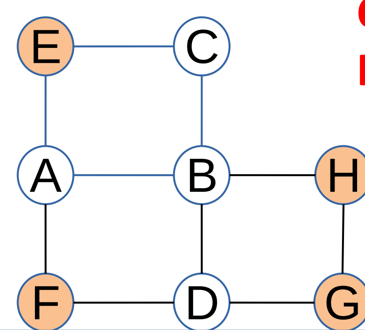
[Marinescu, Dechter and Ihler, 2014]

AND/OR Search for Marginal MAP



MAP variables

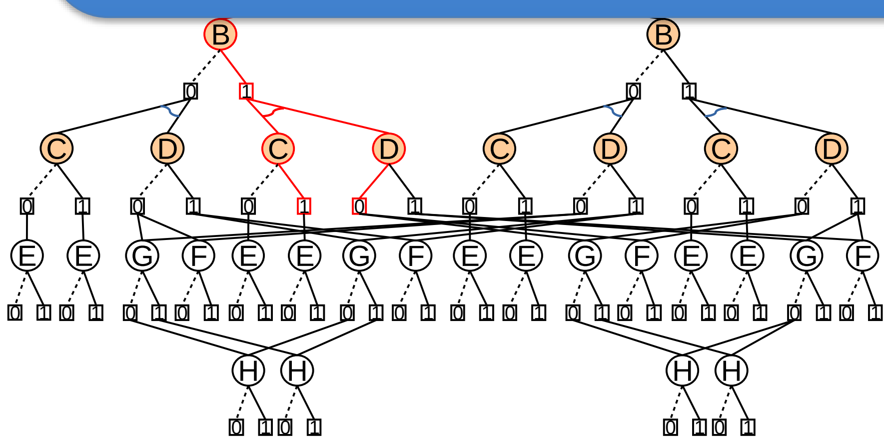


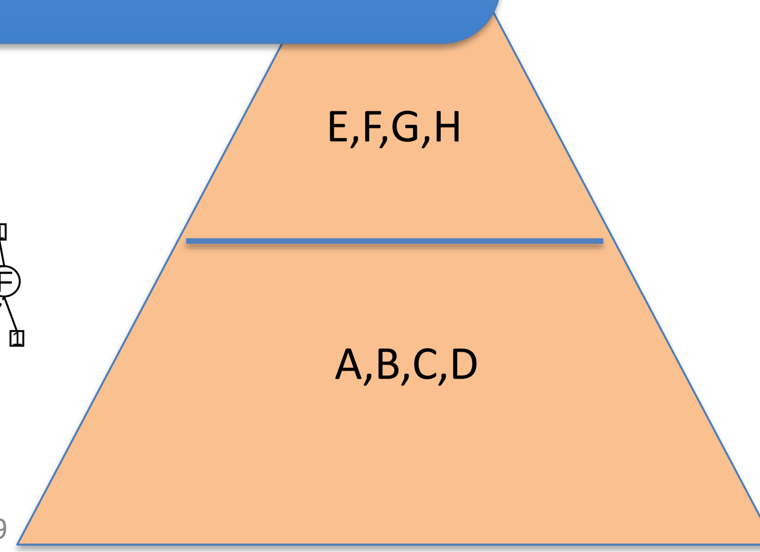


C-induced-width = 6
Induced-width = 2

const
pseu

For MMAP search space is:
k^{h_c} on a AND/OR tree
k^{w_c} on AND/OR graph





[Marinescu, Dechter and Ihler, 2014]

For anytime behavior we need conditioning
And we need heuristics to guide search

Generating Heuristic Using Relaxed Tractable Models

Mini-Bucket Approximation

For optimization

Split a bucket into mini-buckets —> bound complexity

bucket (X) =

$$\left\{ \underbrace{f_1,\ f_2,\ \ldots\ f_r,\ f_{r+1},\ \ldots\ f_n}\right\}$$

$$\lambda_X(\cdot) = \max_x \prod_{i=1}^{n} f_i(x,\ldots)$$

$$\left\{ f_1,\ \ldots\ f_r \right\} \qquad \left\{ f_{r+1},\ \ldots\ f_n \right\}$$

$$\lambda_{X,1}(\cdot) = \max_x \prod_{i=1}^{r} f_i(x,\ldots) \qquad \lambda_{X,2}(\cdot) = \max_x \prod_{i=r+1}^{n} f_i(x,\ldots)$$

$$\lambda_X(\cdot) \ \leq\ \lambda_{X,1}(\cdot)\ \lambda_{X,2}(\cdot)$$

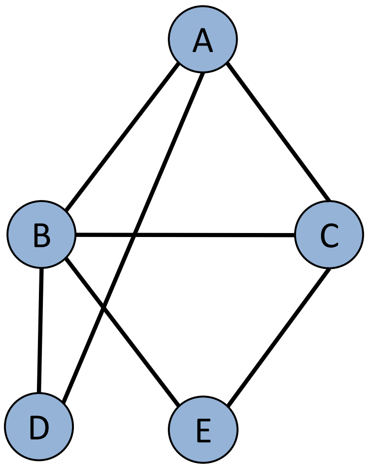
Exponential complexity decrease: $O(e^n) \longrightarrow O(e^r) + O(e^{n-r})$

Mini-Bucket Elimination

[Dechter & Rish 2003]

For optimization

mini-buckets



bucket B: $f(a,b)\ f(b,c)$ $\quad$ $f(b,d)\ f(b,e)$

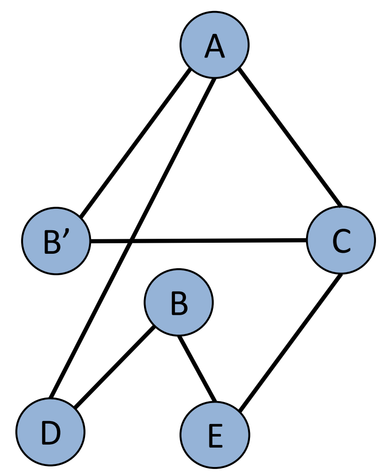
bucket C: $f(c,a)\ f(c,e)\ \lambda_{B\to C}(a,c)$

bucket D: $f(a,d)\ \ \lambda_{B\to D}(d,e)$

bucket E: $\lambda_{C\to E}(a,e)\ \lambda_{E\to E}(a,e)$

bucket A: $f(a)\ \lambda_{E\to A}(a)$

U = upper bound



$$\lambda_{B\to C}(a,c) = \max_b f(a,b)\ f(b,c)$$

$$\lambda_{B\to D}(d,e) = \max_b f(b,d)\ f(b,e)$$

$$\lambda_{C\to E}(a,e) = \max_c \ldots$$

Mini-Bucket Decoding

For optimization

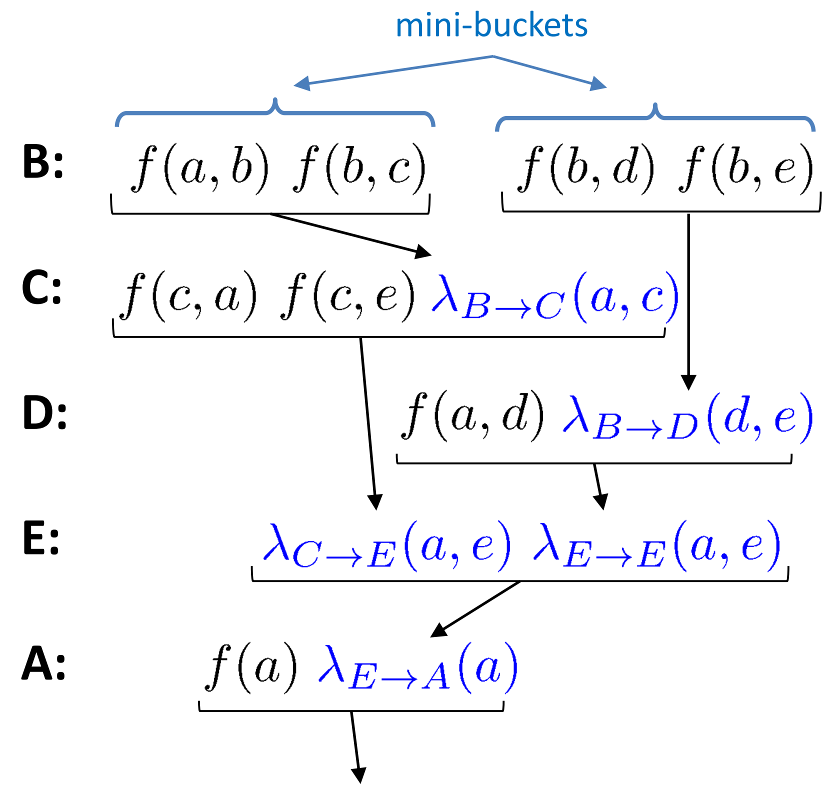
$$\mathbf{b}^* = \arg\max_b f(a^*, b) \cdot f(b, c^*) \cdot f(b, d^*) \cdot f(b, e^*)$$

$$\mathbf{c}^* = \arg\max_c f(c, a^*) \cdot f(c, e^*) \cdot \lambda_{B\to C}(a^*, c)$$

$$\mathbf{d}^* = \arg\max_d f(a^*, d) \cdot \lambda_{B\to D}(d, e^*)$$

$$\mathbf{e}^* = \arg\max_e \lambda_{C\to E}(a^*, e) \cdot \lambda_{D\to E}(a^*, e)$$

$$\mathbf{a}^* = \arg\max_a f(a) \cdot \lambda_{E\to A}(a)$$



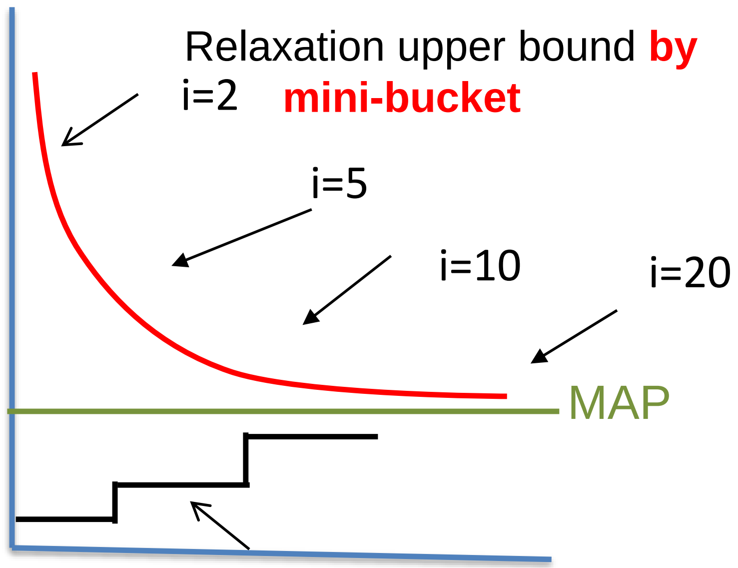
U = upper bound

Greedy configuration = lower bound

Properties of Mini-Bucket Eliminaton

(For optimization)

- Bounding from above and below

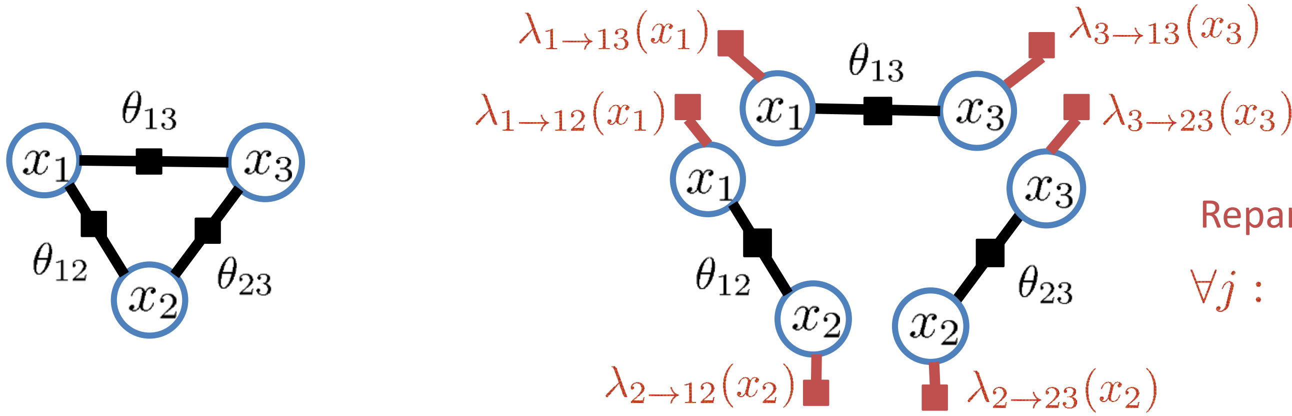


Consistent solutions (greedy search)

- Complexity: O(r exp(i)) time and O(exp(i)) space.
- Accuracy: determined by Upper/Lower bound.
- As i increases, both accuracy and complexity increase.
- Possible use of mini-bucket approximations:
  - As anytime algorithms
  - As heuristics in search
-

Tightening the Bound

Add factors that "adjust" each local term, but cancel out in total



Reparameterization:

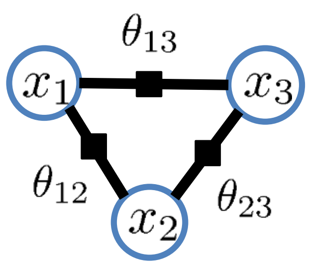
$$\forall j: \sum_{\alpha \ni j} \lambda_{j\to\alpha}(x_j) = 0$$

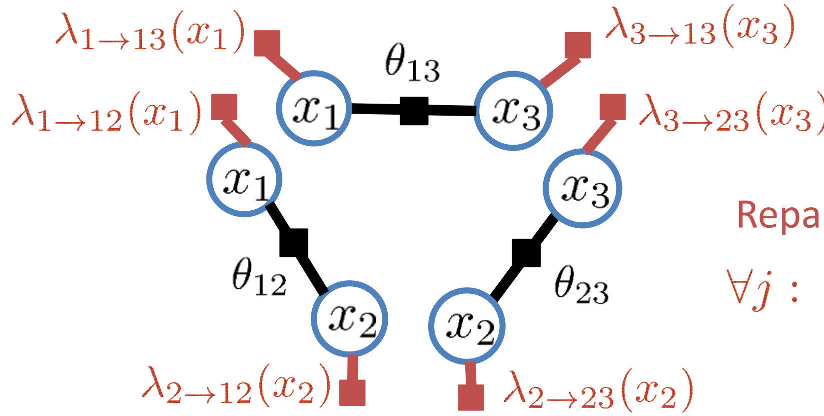
$$\log f(\mathbf{x}^*) = \max_{\mathbf{x}} \sum_{\alpha} \theta_\alpha(\mathbf{x}_\alpha) \leq \min_{\{\lambda_{i\to\alpha}\}} \sum_{\alpha} \max_{\mathbf{x}_\alpha} \left[\theta_\alpha(\mathbf{x}_\alpha) + \sum_{i\in\alpha} \lambda_{i\to\alpha}(x_i) \right]$$

- Bound solution using decomposed optimization
- Solve independently: optimistic bound
- Tighten the bound by re-parameterization
  - Enforces lost equality constraints using Lagrange multipliers

Tightening the bound

Add factors that "adjust" each local term, but cancel out in total





Reparameterization:

$$\forall j : \sum_{\alpha \ni j} \lambda_{j\to\alpha}(x_j) = 0$$

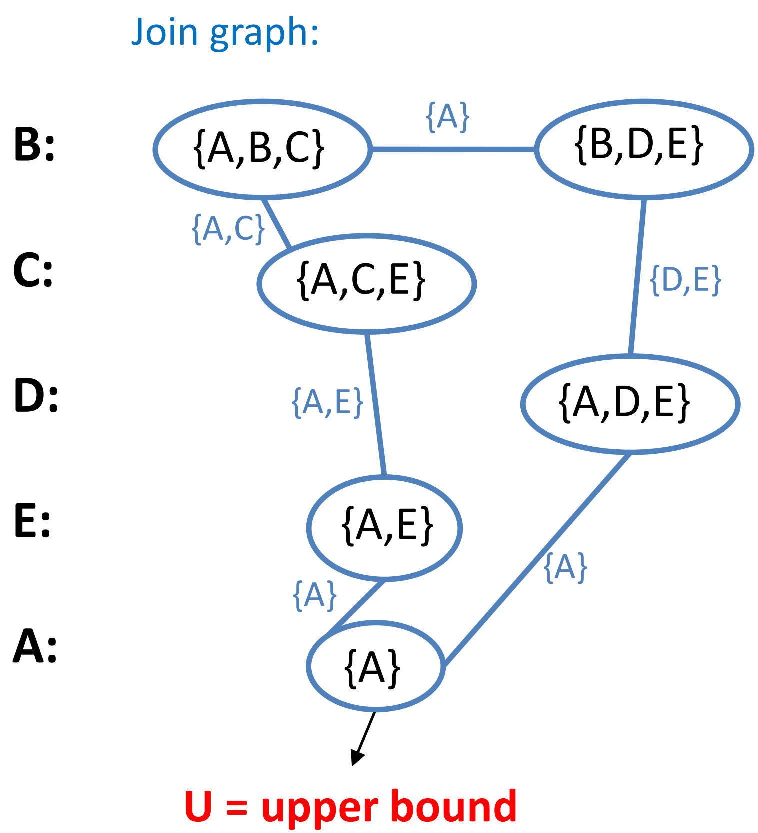
$$\log f(\mathbf{x}^*) = \max_{\mathbf{x}} \sum_{\alpha} \theta_\alpha(\mathbf{x}_\alpha) \quad \leq \min_{\{\lambda_{i\to\alpha}\}} \sum_{\alpha} \max_{\mathbf{x}_\alpha} \left[\theta_\alpha(\mathbf{x}_\alpha) + \sum_{i\in\alpha} \lambda_{i\to\alpha}(x_i) \right]$$

- Many names for the same class of bounds
  - Dual decomposition [Komodakis et al. 2007]
  - TRW, MPLP [Wainwright et al. 2005; Globerson & Jaakkola 2007]
  - Soft arc consistency [Cooper & Schiex 2004]
  - Max-sum diffusion [Warner 2007]

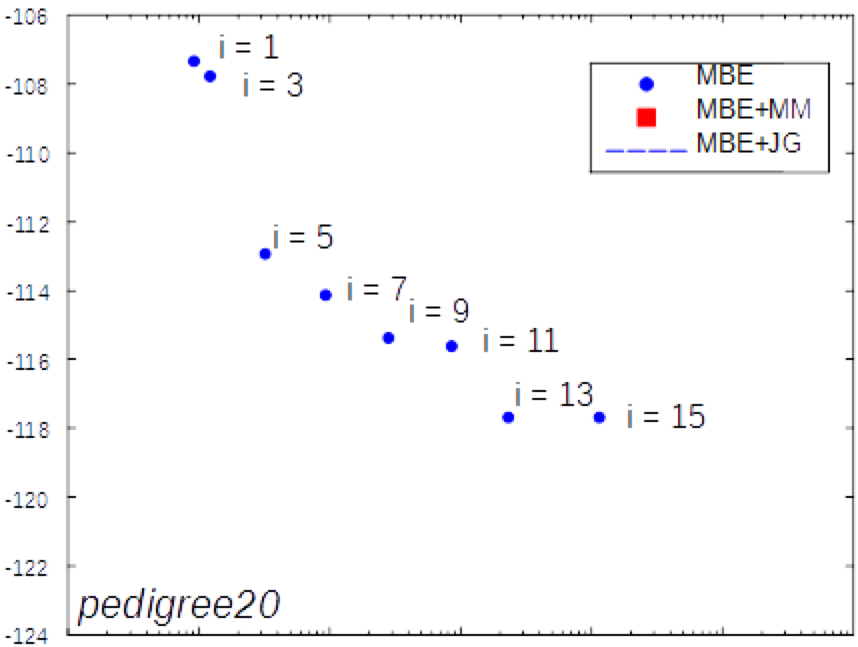
Mini-Bucket with Moment-Matching

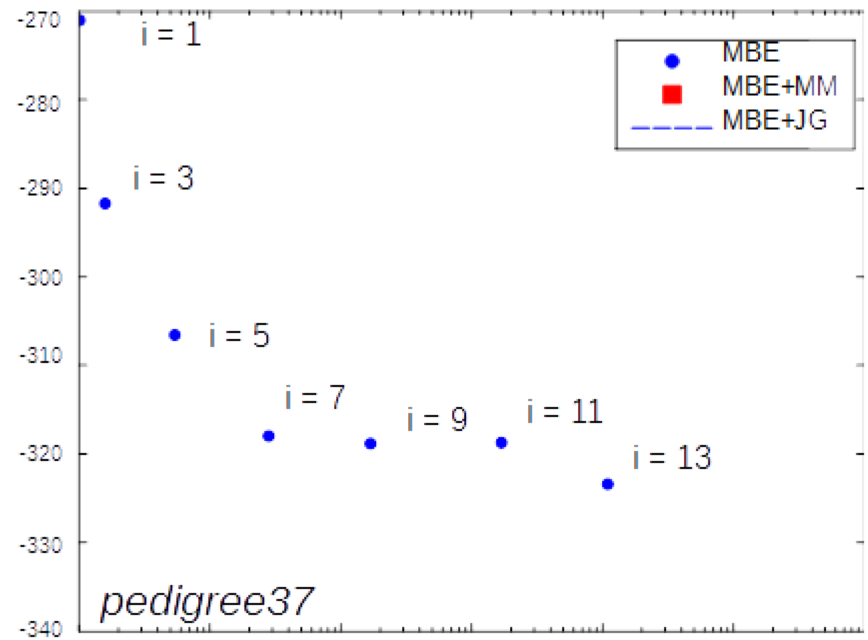
[Ihler et al. 2012]

- Downward pass as cost shifting
- Can also do cost shifting within mini-buckets:
  "Join graph" message passing
- "Moment-matching" version:
  One message exchange within each bucket, during downward sweep
- Optimal bound defined by cliques ("regions") and cost-shifting f'n scopes ("coordinates")



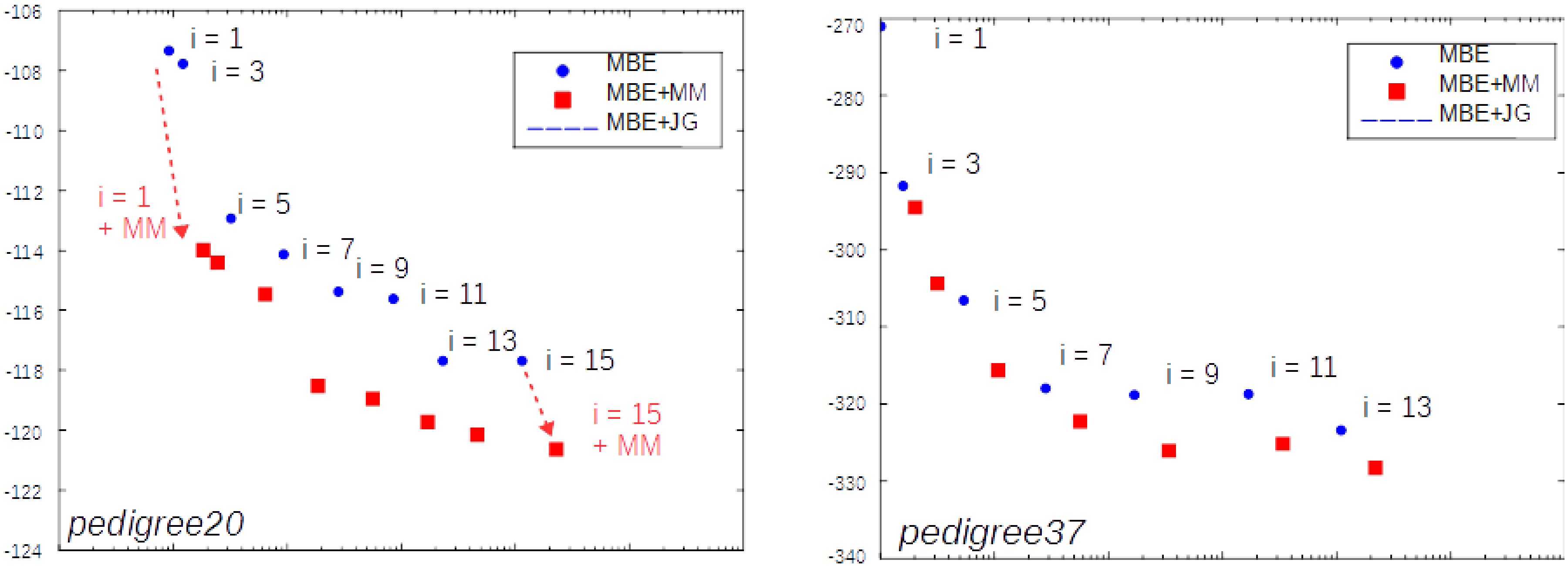
Anytime Approximation





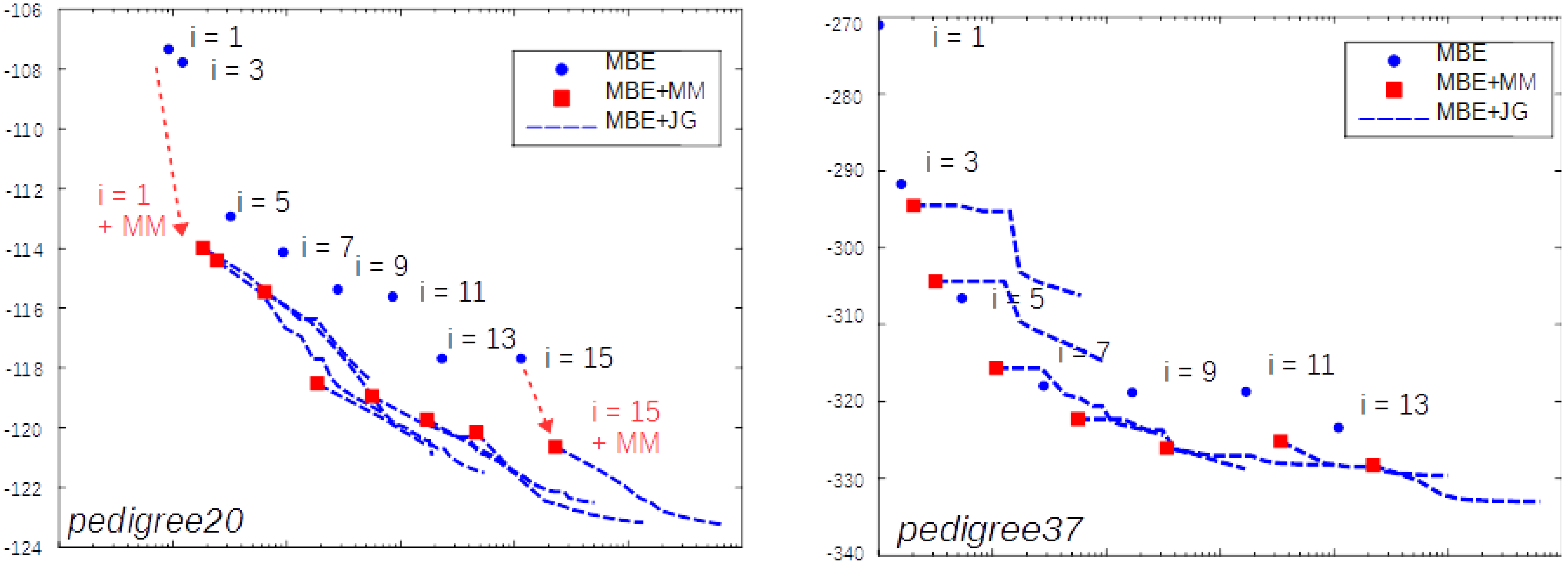
- Can tighten the bound in various ways
  - Cost-shifting (improve consistency between cliques)
  - Increase i-bound (higher order consistency)
- Simple moment-matching step improves bound significantly

Anytime Approximation



- Can tighten the bound in various ways
  - Cost-shifting (improve consistency between cliques)
  - Increase i-bound (higher order consistency)
- Simple moment-matching step improves bound significantly

Anytime Approximation



- Can tighten the bound in various ways
  - Cost-shifting (improve consistency between cliques)
  - Increase i-bound (higher order consistency)
- Simple moment-matching step improves bound significantly

Mini-Bucket for Summation

(Liu &Ihler, 2011)

$$F(x) = f_1(x) \cdot f_2(x)$$

- Generalize technique to sum via Holder's inequality:

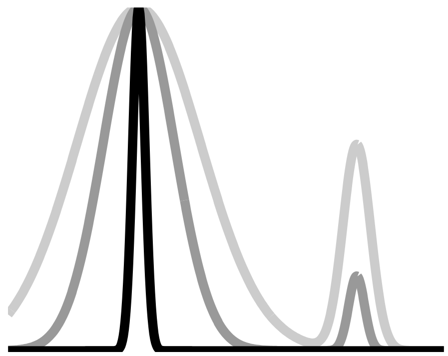
$$\sum_x f_1(x) \cdot f_2(x) \quad \leq \quad \Big[\sum_x f_1(x)^{\frac{1}{w_1}}\Big]^{w_1} \cdot \Big[\sum_x f_2(x)^{\frac{1}{w_2}}\Big]^{w_2}$$

$$w_1 + w_2 = 1$$

- Define the weighted (or powered) sum:

$$\sum_{x_1}^{w_1} f(x_1) = \Big[\sum_{x_1} f(x_1)^{\frac{1}{w_1}}\Big]^{w_1}$$

  – "Temperature" interpolates between sum & max:



$$\lim_{w \to 0^+} \sum_x^{w} f(x) = \max_x f(x)$$

  – Different weights do not commute:

$$\sum_{x_1}^{w_1} \sum_{x_2}^{w_2} f(x_1, x_2) \neq \sum_{x_2}^{w_2} \sum_{x_1}^{w_1} f(x_1, x_2)$$

WMB for Marginal MAP

$$\lambda_{B\to C}(a,c) = \sum_b^{w_1} f(a,b)f(b,c)$$

$$\lambda_{B\to D}(d,e) = \sum_b^{w_2} f(b,d)f(b,e)$$

$$(w_1 + w_2 = 1)$$

⋮

$$\lambda_{E\to A}(a) = \max_e \lambda_{C\to E}(a,e)\lambda_{D\to E}(a,e)$$

$$U = \max_a f(a)\lambda_{E\to A}(a)$$

Marginal MAP

mini-buckets

Σ_B	bucket B:	$f(a,b)\ f(b,c)$ $f(b,d)\ f(b,e)$
Σ_C	bucket C:	$\lambda_{B\to C}(a,c)\ f(a,c)\ f(c,e)$
$\max_D$	bucket D:	$f(a,d)\ \lambda_{B\to D}(d,e)$
$\max_E$	bucket E:	$\lambda_{C\to E}(a,e)\ \lambda_{D\to E}(a,e)$
$\max_A$	bucket A:	$f(a)\ \lambda_{E\to A}(a)$

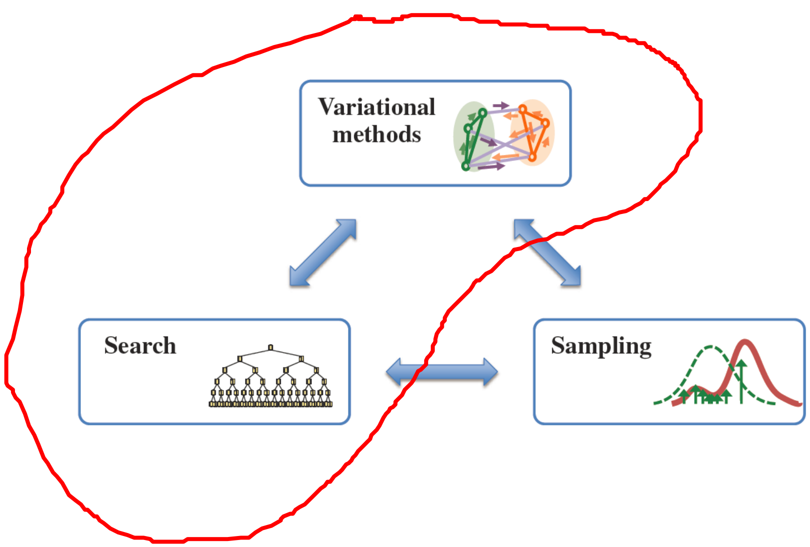
U = upper bound

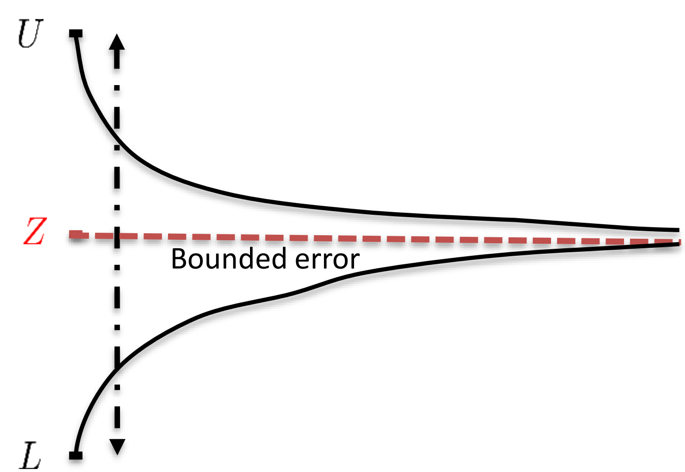
Can optimize over cost-shifting and weights
(single pass "MM" or iterative message passing)

[Liu and Ihler, 2011; 2013]
[Dechter and Rish, 2003]

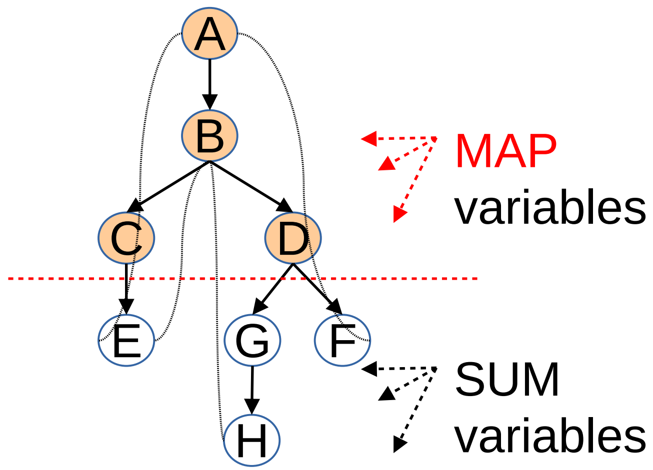
Outline

- Graphical models, The Marginal Map task
- AND/OR search spaces
- Variational bounds as search heuristics
- Combining methods: Heuristic Search for Marginal Map
- Combining methods: Sampling
- Conclusion

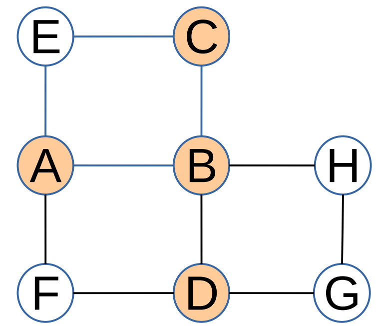




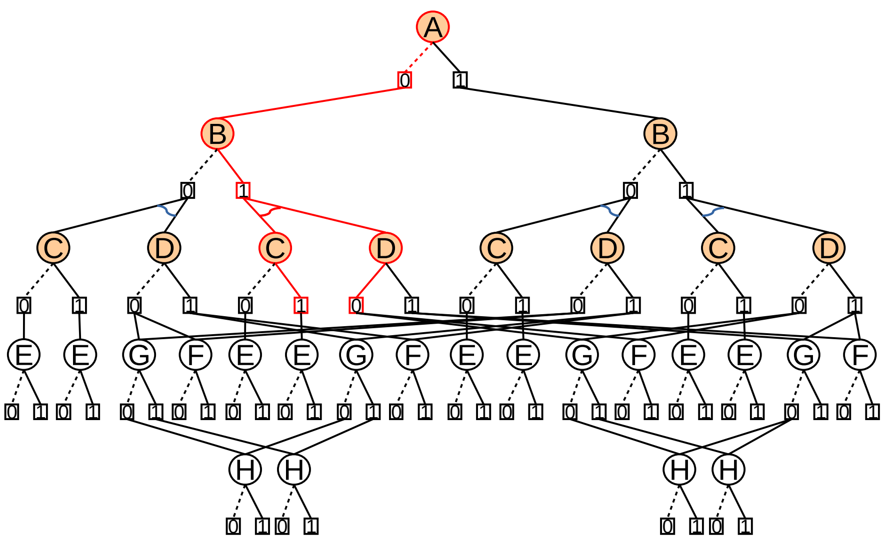
AND/OR Search for Marginal MAP



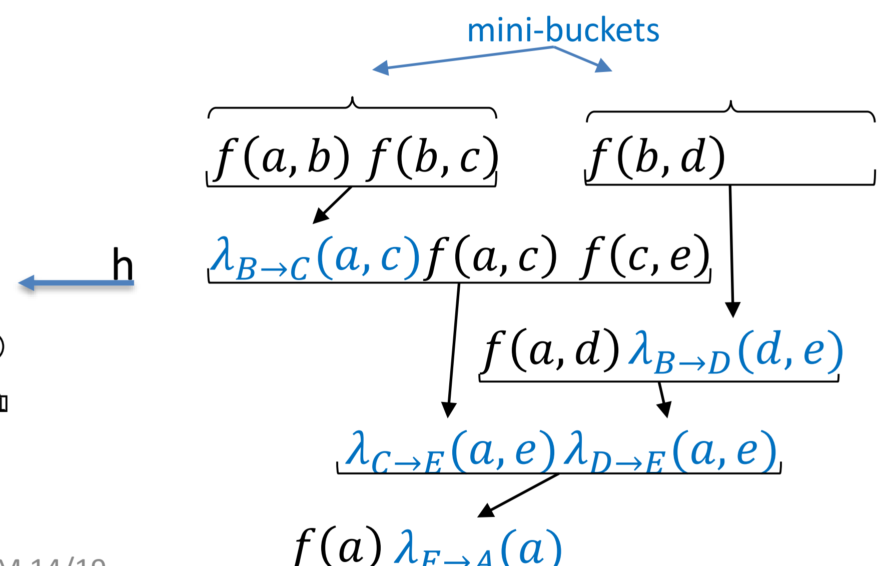
constrained
pseudo tree



primal graph

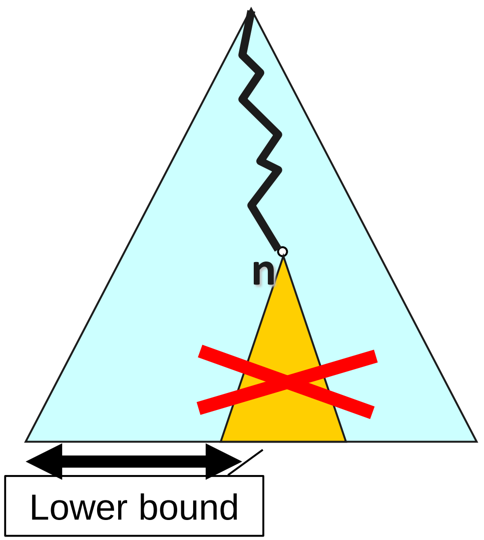


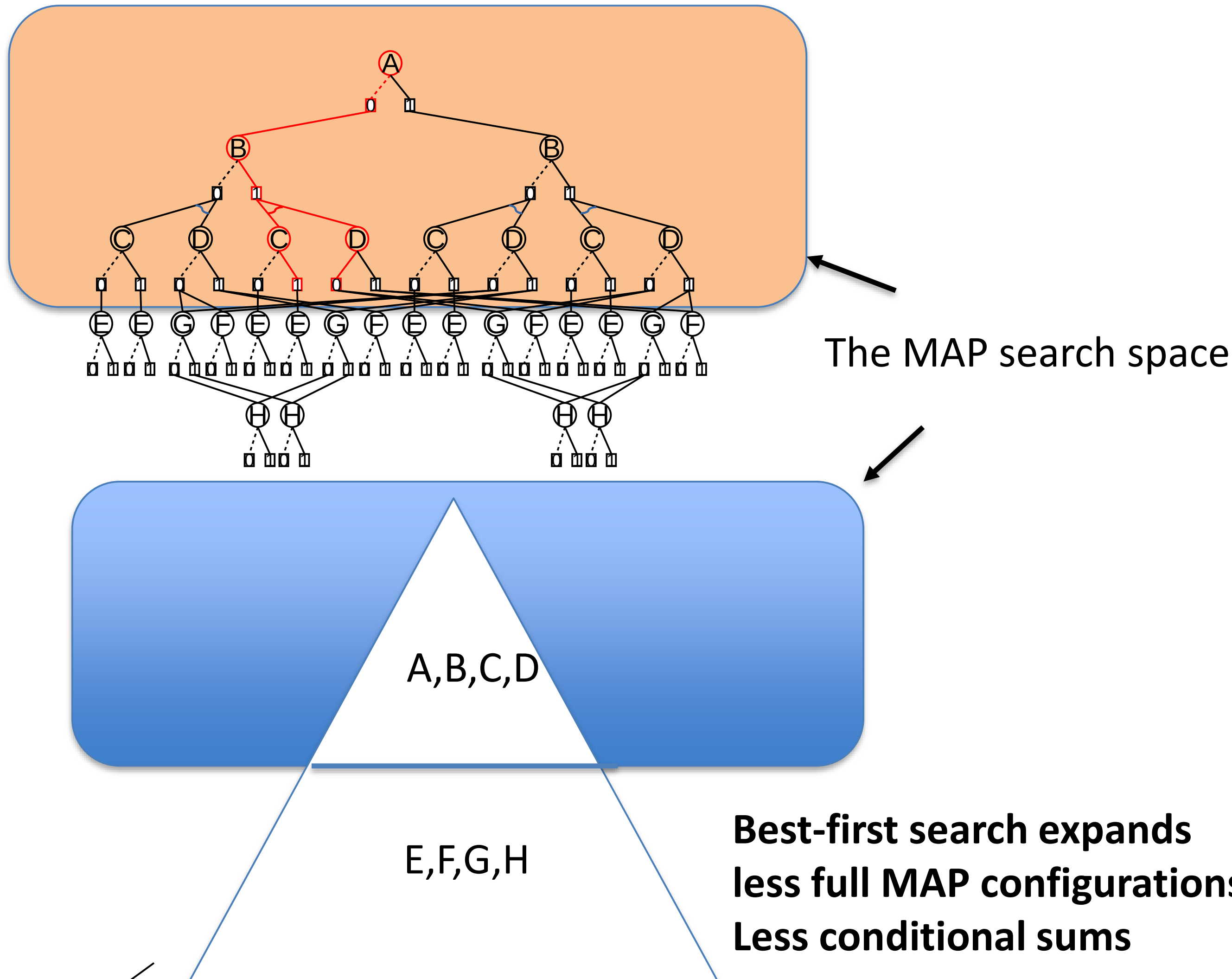
[Marinescu, Dechter and Ihler, 2014]



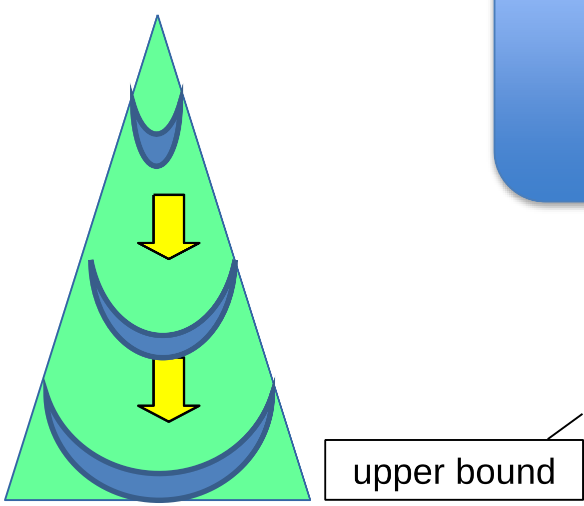
Exact MMAP Solvers: Best or Depth-First Search?

Depth-First search





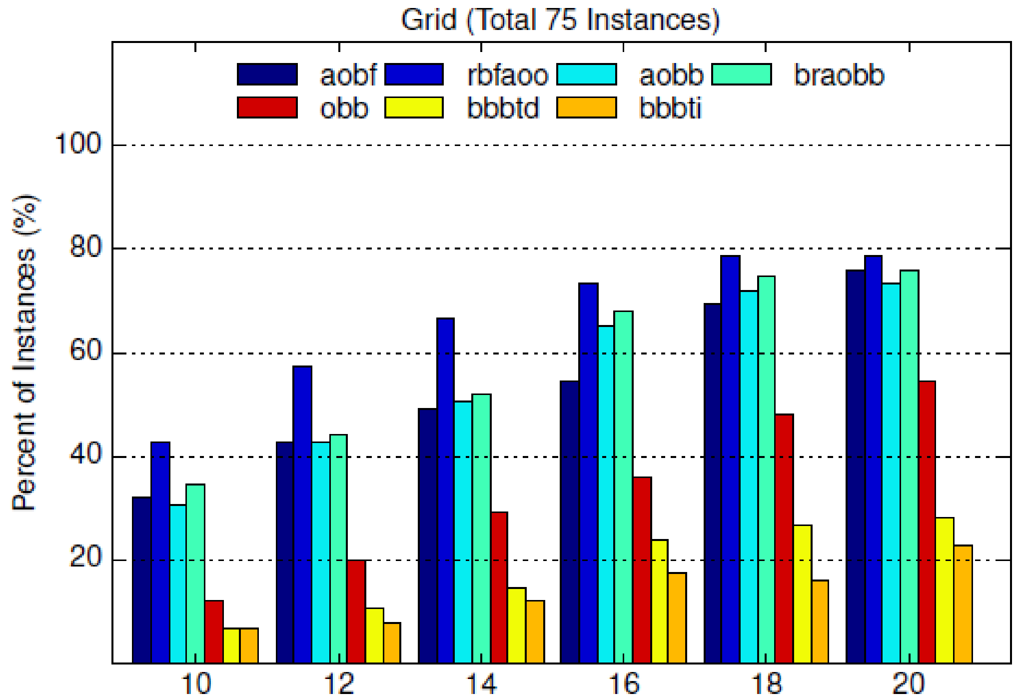
Best-First search



Best-first search expands less full MAP configurations Less conditional sums

MMAP: Exact AND/OR solvers

Benchmarks:
Grids (128)
Pedigrees (88)
Promedas (100)



AOBF
RBFAOO - recursive
BRAOBB
Yuan, Park BBTDi, BBBTD
Time-bound 2 hours

- **AND/OR search+ MB-heuristic are superior**
- to OR search using "unordered heuristic" [Park and Darwiche 2003, Yuan and Hansen 2009] when the constrained induced-width is not bounded.

- **Best-first schemes are better because less summations evaluation**

Anytime Solvers for Marginal MAP

- **Weighted Heuristic:** [Lee et. al. AAAI-2016, JAIR 2019]
  - Weighted Restarting AOBF (WAOBF)
  - Weighted Restarting RBFAOO (WRBFAOO)
  - Weighted Repairing AOBF (WRAOBF)

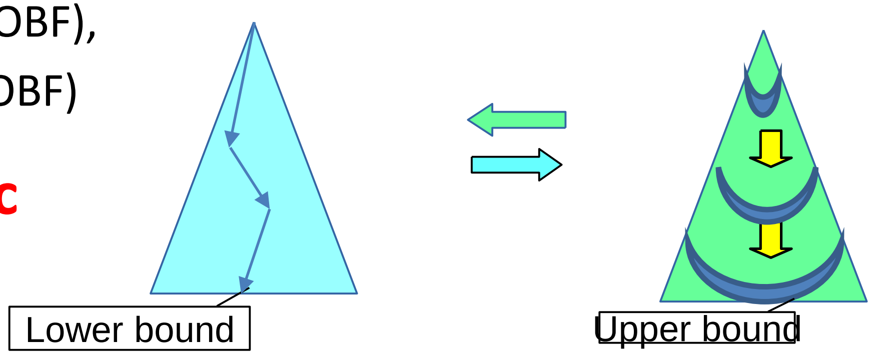
Weighted A* search [Pohl 1970]

- non-admissible heuristic
- Evaluation function:

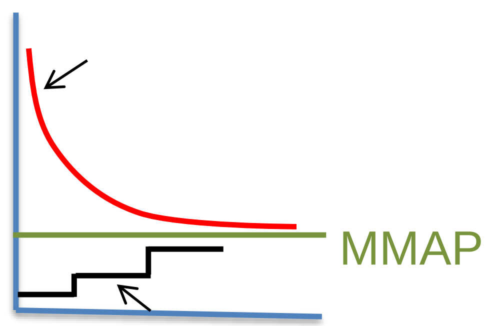
  $f(n)=g(n)+w \cdot h(n)$
- **Guaranteed w-optimal solution, cost** $C \leq w \cdot C^*$

- **Interleaving Best and depth-first search**: (Marinescu et. al AAAI-2017)
  - Look-ahead (LAOBF),
  - alternating (AAOBF)

Exploiting heuristic search ideas



Goal: anytime bounds
And anytime solution



Anytime Bounds of Marginal MAP

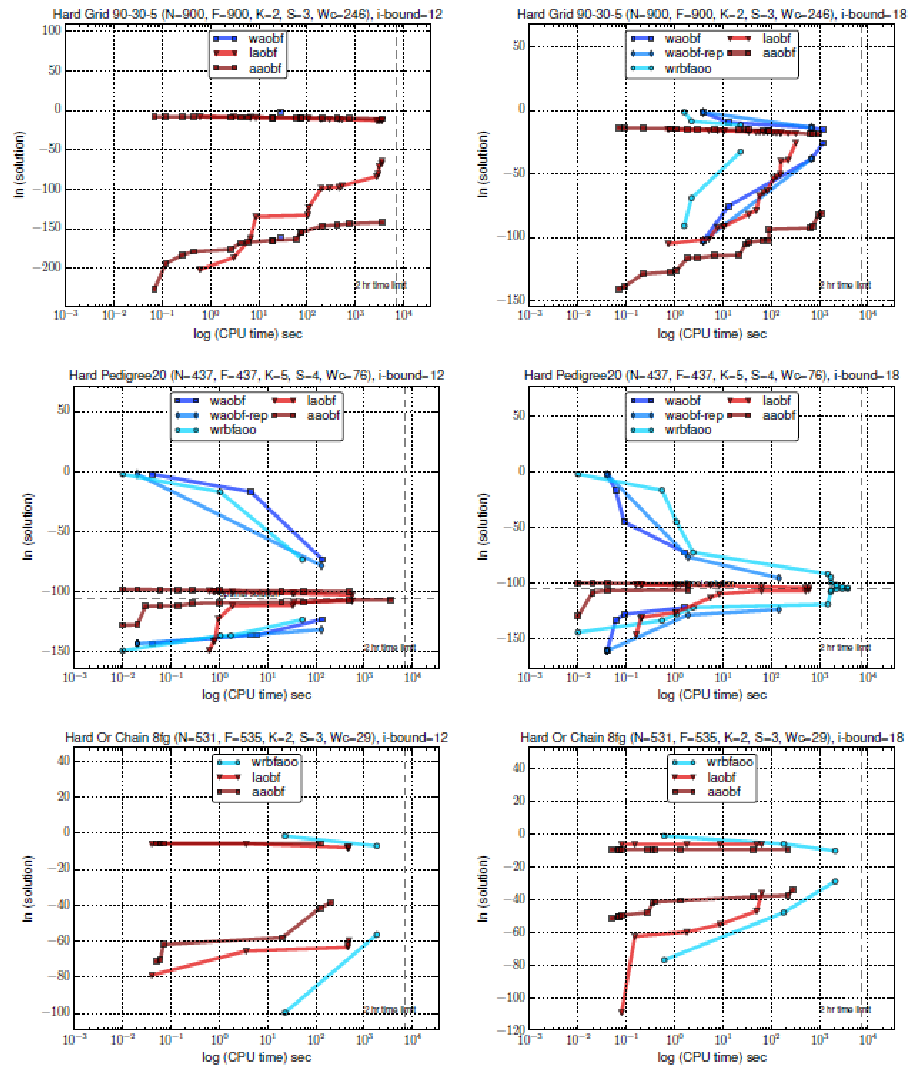
(UAI'14, IJCAI'15, AAAI'16, AAAI'17, JAIR 2019 (Marinescu, Lee, Ihler, Dechter)

- Search: LAOBF, AAOBF, BRAOBB, WAOBF,WAOBF-rep
- heuristic: WMB-MM (20)
- memory: 24 GB

- Anytime lower and upper bounds from hard problem instances with **i-bound 12 (left) and 18 (right)**.

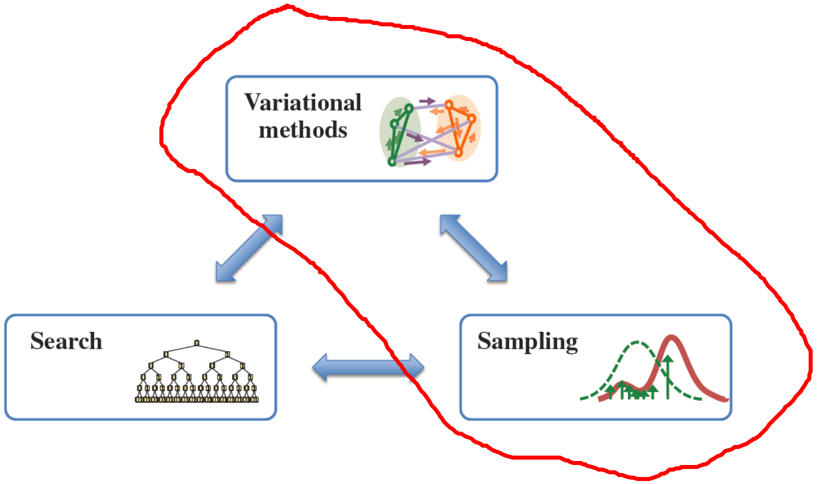
- The horizontal axis is the CPU time in log scale and the vertical axis is the value of marginal MAP in log scale.

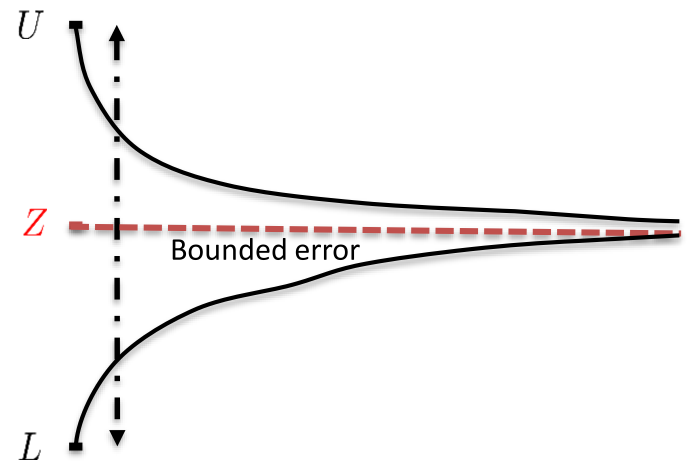
But, limited to tractable condition-summation



Outline

- Graphical models, The Marginal Map task
- AND/OR search spaces
- Variational bounds as search heuristics
- Combining methods: Heuristic Search for Marginal Map
- Combining methods: Sampling
- Conclusion





Choose a Proposal Combine w Search

Building blocks in current algorithms for Markov Logic Networks

- Probabilistic Theorem Proving: Gogate and Domingos, CACM 2016,
- Lifted Importance Sampling: Venugopal and Gogate, NeurIPS 2014.

- 2))]
- Sampling based lower bounds [Gogate, Dechter (Intelligenza Artificiale, 2011)]
- Dynamic Importance Sampling (DIS) [Lou, Dechter, and Ihler (NIPS 2017)]
- Abstraction Sampling [Broka, Dechter, Ihler and Kask (UAI, 2018)].
- Finite-sample Bounds for MMAP [Lou, Dechter, and Ihler. (UAI 2018)]
- WMB Importance Sampling (WMB-IS) [Liu, Fisher, Ihler (ICML 2015)]

Choosing a Proposal- WMB-IS

[Liu, Fisher, Ihler 2015]

- Can use WMB upper bound to define a proposal $q_{\mathrm{wmb}}(x)$

$$\tilde{\mathbf{b}} \sim w_1\, q_1(b|\tilde{a},\tilde{c}) \;+\; w_2\, q_2(b|\tilde{d},\tilde{e})$$

Weighted mixture:

use minibucket 1 with probability w_1

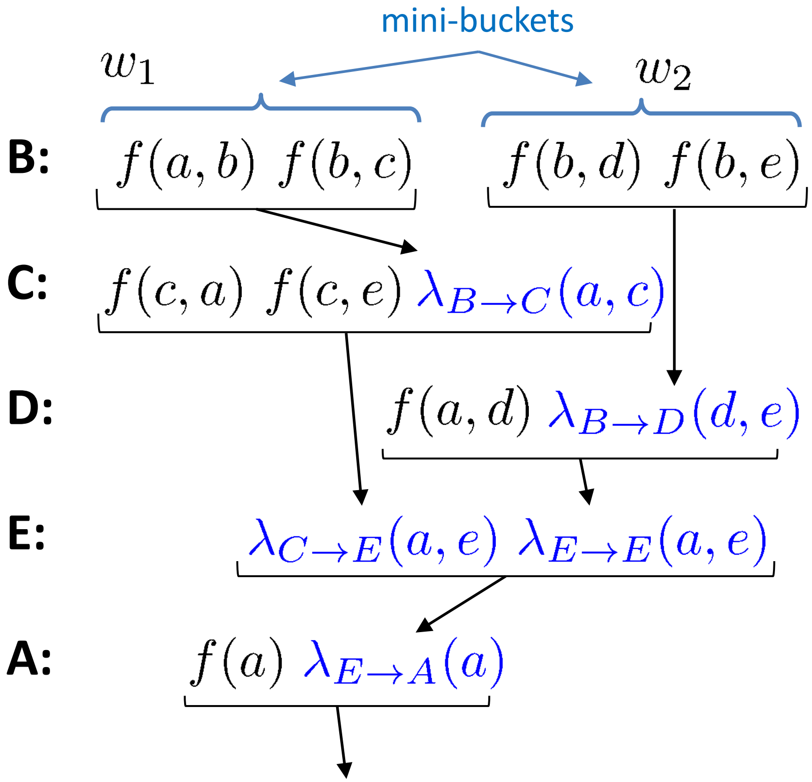
or, minibucket 2 with probability $w_2 = 1 - w_1$

where

$$q_1(b|a,c) = \left[\frac{f(a,b)\cdot f(b,c)}{\lambda_{B\to C}(a,c)}\right]^{\frac{1}{w_1}}$$

⋮

$$\tilde{\mathbf{a}} \sim q(A) = f(a)\cdot\lambda_{E\to A}(a)/U$$



***U* = upper bound**

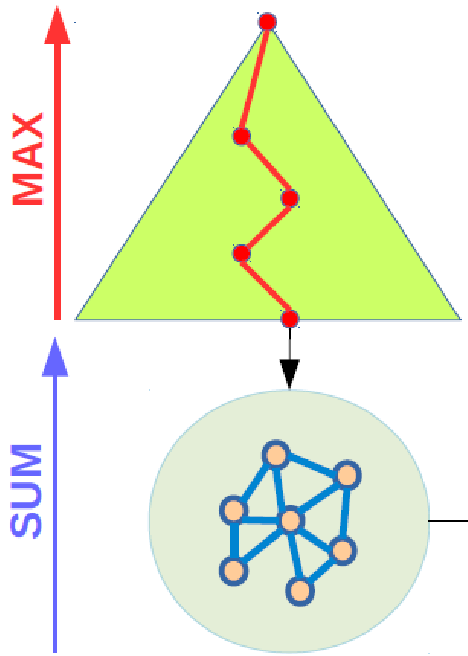
Key insight: provides bounded importance weights!

$$0 \le f(x)/q_{\mathrm{wmb}}(x) \le U \quad \forall x$$

Probabilistic Lower Bounds For MMAP

[Liu *et al*. 2015]

$$\max_{\mathbf{X}_M} \sum_{\mathbf{X}_S} P(\mathbf{X})$$



$$Z = \sum_{\mathbf{X}_S} P(X)|_{\bar{x}_M}$$

Solving the conditioned SUM subproblem is hard!

$$\#P - complete$$

Compute a (probabilistic) lower bound on the conditioned sum subproblem

$$Pr(\hat{Z} - \Delta(n, \delta) \leq Z) \geq (1 - \delta)$$

WMB based importance sampling scheme:

n - number of samples

δ - confidence value

Z_{wmb} - result of WMB

$\hat{Z}$ - Importance Sampling estimate

$$\Delta(n, \delta) = \sqrt{\frac{2\hat{var}(w(x))\log(2/\delta)}{n}} + \frac{7Z_{wmb}\log(2/\delta)}{3(n-1)}$$

Empirical variance, decreasing as $1/n^{1/2}$

Upper bound U, decreasing as $1/n$

Experiments

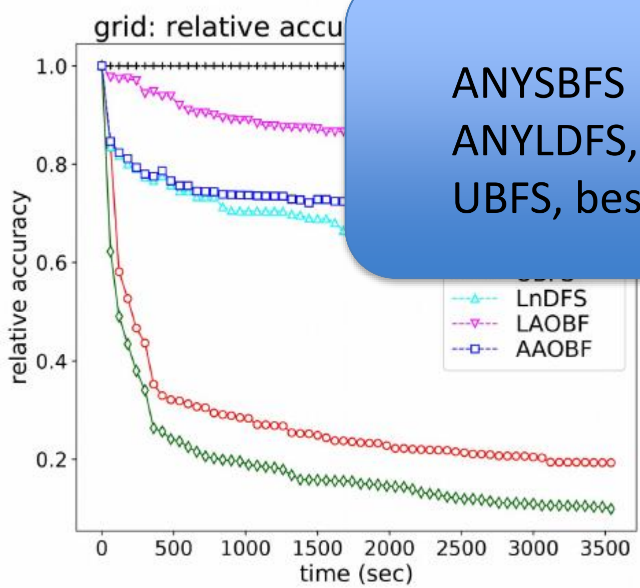
- Anytime Algorithms
  - State-of-the-art
    - LAOBF [Marinescu, Lee, Dechter, Ihler, 2017]
    - AAOBF [Marinescu, Lee, Dechter, Ihler, 2017]
    - LnDFS [Marinescu, Lee, Dechter, Ihler, 2017]
    - UBFS [Qi, Ihler, 2018]

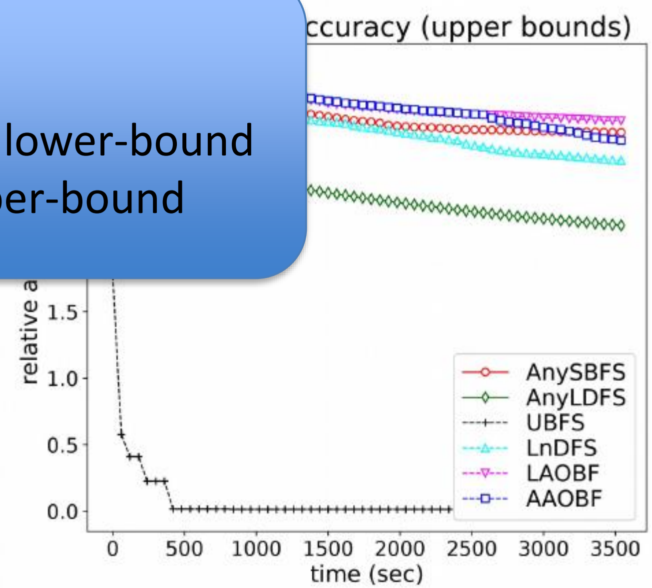
    (LAOBF, AAOBF, LnDFS: Hybrid best+depth-first search)
  - **Proposed schemes**
    - **AnySBFS (p = 0.5)**
    - **AnyLDFS**
- Benchmarks
  - Grid, pedigree, promedas, planning (UCIrvine graphical models repo)
  - 10% of variables selected randomly as MAP variables
    - Hard (intractable) conditioned summation subproblems
  - Parameters: confidence 0.05

MMAP: Combining with Sampling

[Lou, Dechter, Ihler, AAAI-2018: “Anytime Anyspace AND/OR Best-first Search for Bounding Marginal MAP”]
[Lou, Dechter, Ihler, UAI-2018: “Finite Sample Bounds for Marginal MAP”, UAI 2018]
[Marinescu, Ihler, Dechter: IJCAI-2018 “Stochastic Anytime Search for Bounding Marginal MAP”]

$$acc_{lb} = \frac{|l_t - l^*|}{l^*} \qquad acc_{ub} = \frac{|u_t - u^*|}{u^*}$$





ANYSBFS
ANYLDFS, best for lower-bound
UBFS, best for upper-bound

l_t – lower bound at time t
l^* – tightest lower bound found

u_t – upper bound at time t
u^* – tightest upper bound found

Average over 150 instances

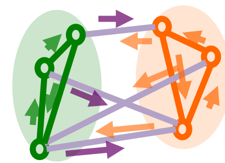
Average over 150 instances

(Lower is better)

Combining Approaches

Variational methods

WMB



provide heuristics

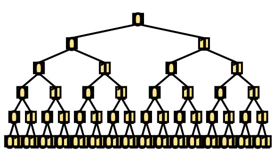
provide proposal

WMB-IS

[Liu et al., NIPS 2015]

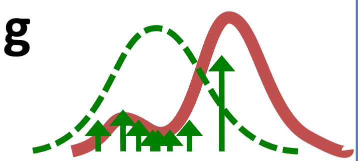
Approximating Summation sub-problems

Search



refine proposal

Sampling



dynamic importance sampling (DIS)

[Lou et al., NIPS 2017]

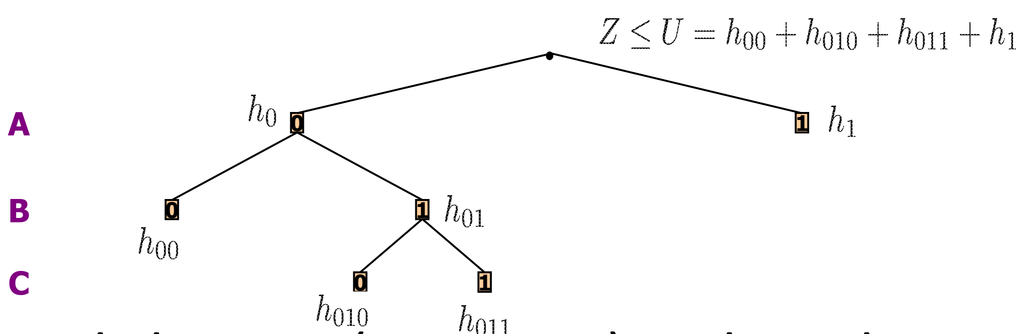
For MAP, marginal map and partition function

Dynamic Importance Sampling

[Lou, Dechter, Ihler, NIPS 2017, AAAI 2019]

- Interleave
  - Building search tree (expand Nd nodes)
  - Draw samples given search bound (Nl samples)

(For partition function)



- Key insight: proposal changes (improves) with each step
  - Use weighted average: better samples get more weight

$$\widehat{Z} = \frac{\mathrm{HM}(\boldsymbol{U})}{N} \sum_{i=1}^{N} \frac{\widehat{Z}_i}{U_i}, \quad \mathrm{HM}(\boldsymbol{U}) = \left[\frac{1}{N} \sum_{i=1}^{N} \frac{1}{U_i}\right]^{-1}$$

  - Derive corresponding concentration bound on Z

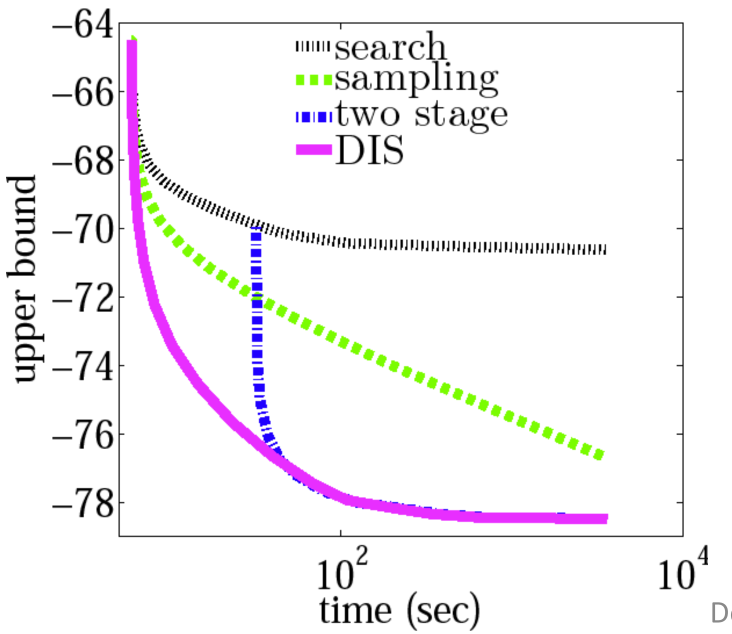
Finite-sample Bounds for DIS

Theorem: Define the deviation term

$$\Delta = \mathrm{HM}(\boldsymbol{U})\Big(\sqrt{\frac{2\widehat{\mathrm{Var}}(\{\widehat{Z}_i/U_i\}_{i=1}^{N})\ln(2/\delta)}{N}} + \frac{7\ln(2/\delta)}{3(N-1)}\Big)$$

then, $\Pr[Z \leq \widehat{Z} + \Delta] \geq 1-\delta$ and $\Pr[Z \geq \widehat{Z} - \Delta] \geq 1-\delta$.

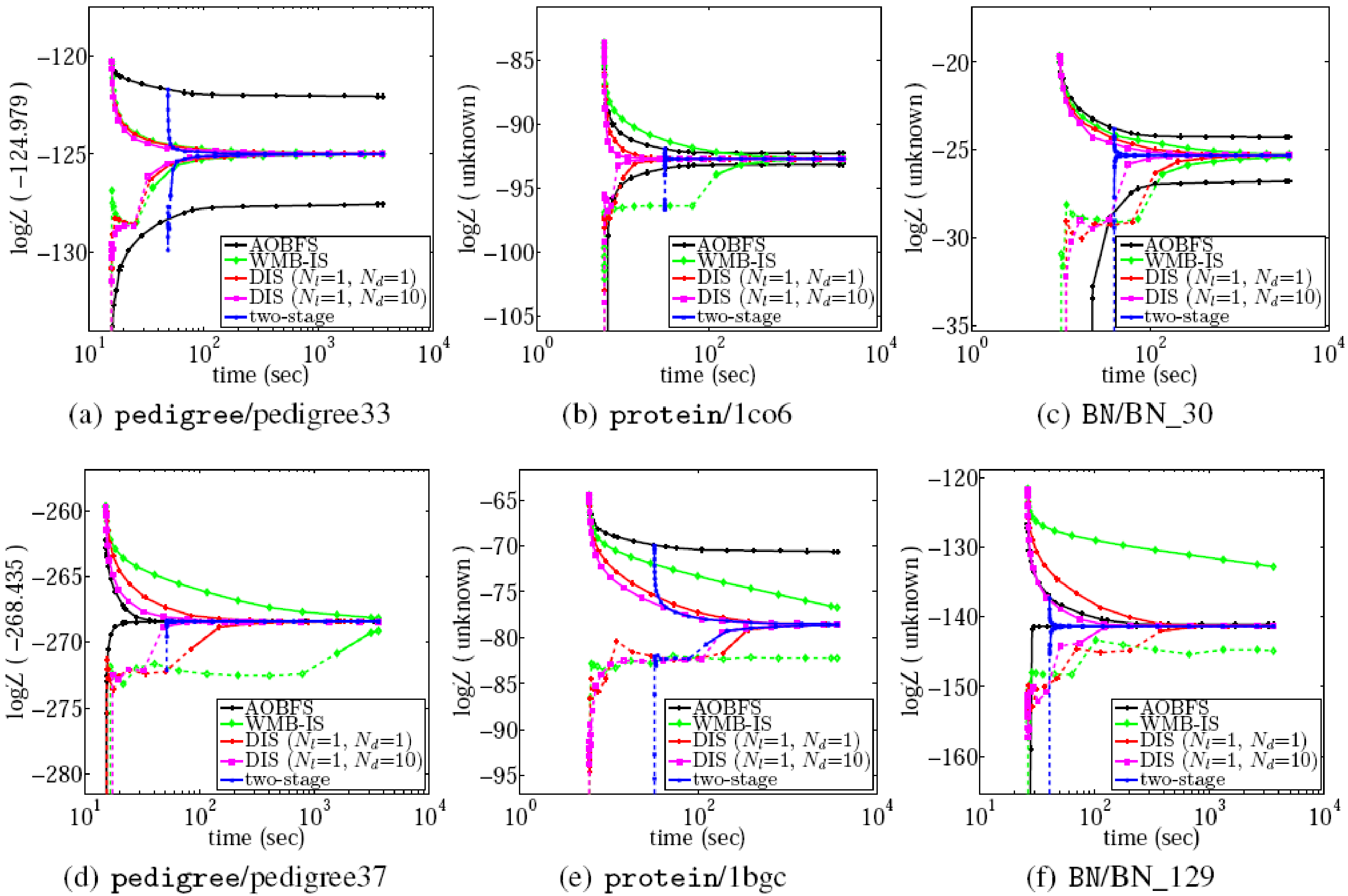
$\widehat{\mathrm{Var}}(\{\widehat{Z}_i/U_i\}_{i=1}^{N})$: empirical variance of $\{\widehat{Z}_i/U_i\}_{i=1}^{N}$.



Individual Results

(For partition function)

[Lou, Dechter, Ihler, NIPS 2017]

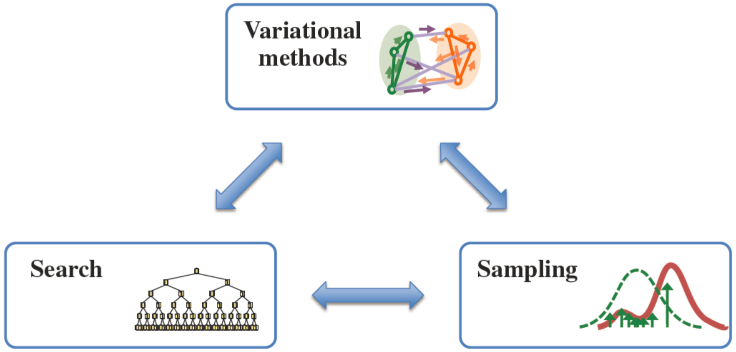


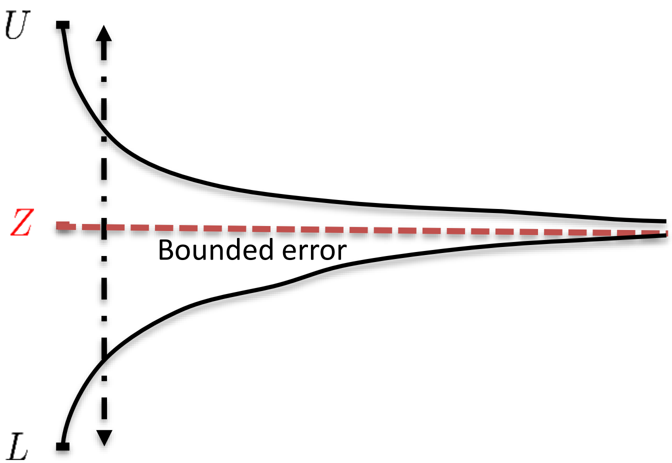
(a) pedigree/pedigree33 (b) protein/1co6 (c) BN/BN_30

(d) pedigree/pedigree37 (e) protein/1bgc (f) BN/BN_129

Outline

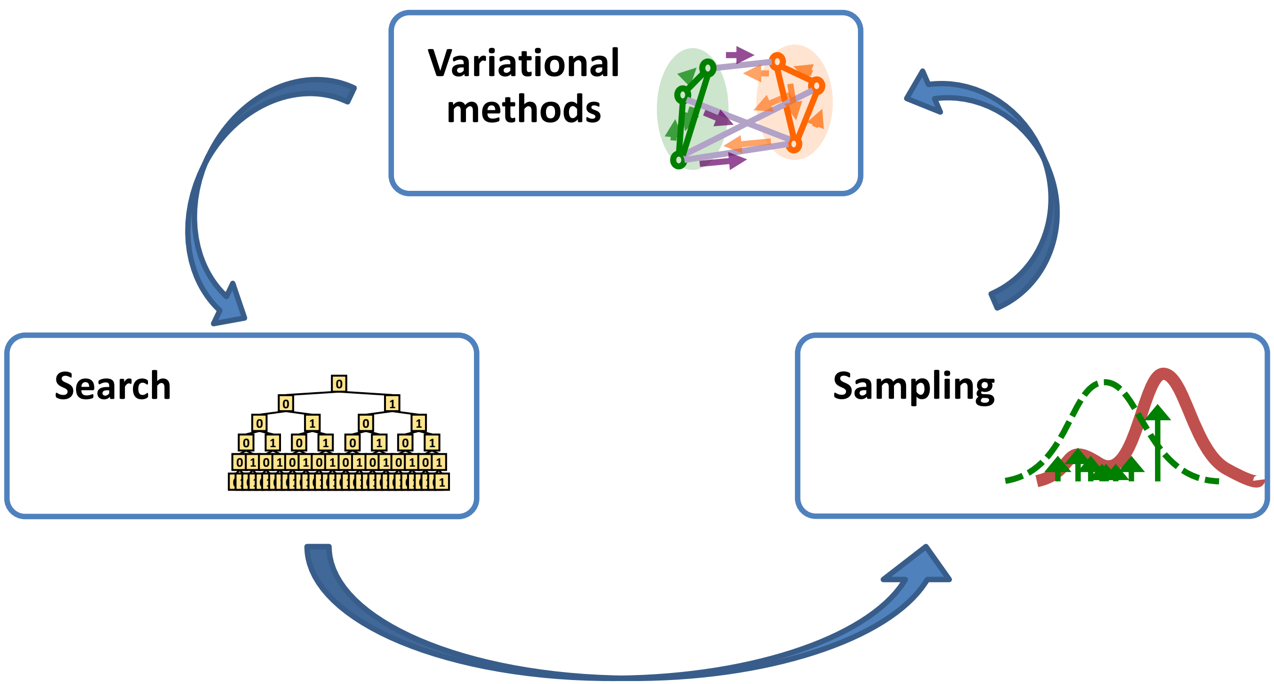
- Graphical models, The Marginal Map task
- AND/OR search spaces
- Variational bounds as search heuristics
- Combining methods: Heuristic Search for Marginal Map
- Combining methods: Sampling
- Conclusion





Continuing work and Conclusions

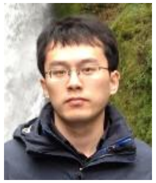
- Exploiting the graph-based tractability coupled with variational improvements can take us far into non-tractabile lands when pursuing anytime probabilistic reasoning



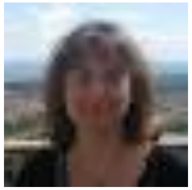
Thank You !

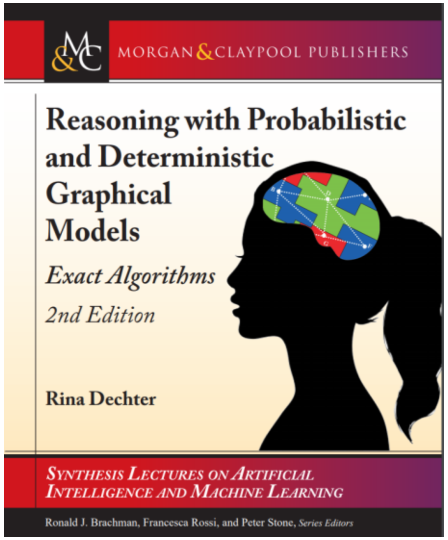
For publication see:
http://www.ics.uci.edu/~dechter/publications.html











Alex Ihler
Kalev Kask
Irina Rish
Bozhena Bidyuk
Robert Mateescu
Radu Marinescu
Vibhav Gogate
Emma Rollon
Lars Otten
Natalia Flerova
Andrew Gelfand
William Lam
Junkyu Lee
Qi Lou