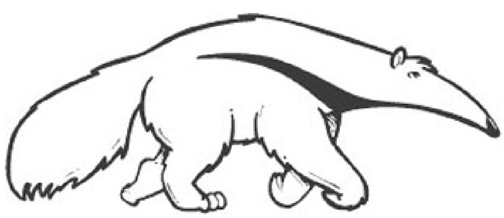
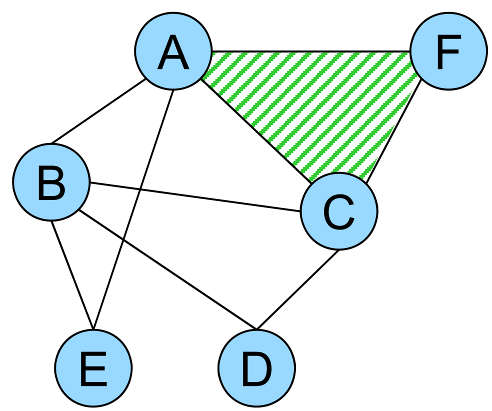
Probabilistic Reasoning Meets Heuristic Search

Rina Dechter

Collaborators:
Radu Marinescu
Alex Ihler
Junkyu Lee





BREN:ICS
INFORMATION AND COMPUTER SCIENCES

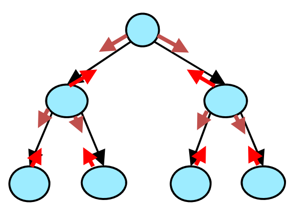
UNIVERSITY of CALIFORNIA IRVINE

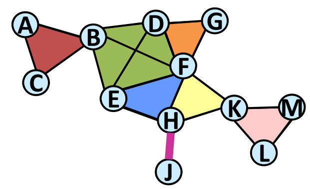
Models in AI

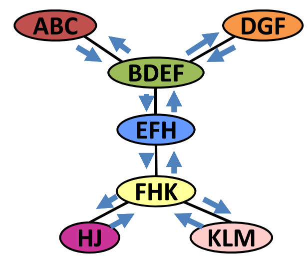
- Models based on states (e.g., planning)
- Models based on variables (SAT/CSP, Bayesian networks, Markov networks, MDPs)
- State-based search models are more general
- Variable-based models have more structural information
- Search was always considered for variable-based models, e.g., Backtracking for CSP/SAT, Integer programming, search for mpe in Bayesian networks.
- Here we will take it few steps further

Search Collaborates with Inference

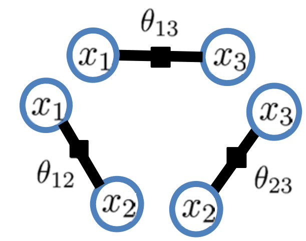
- Inference: message-passing on cluster-tree





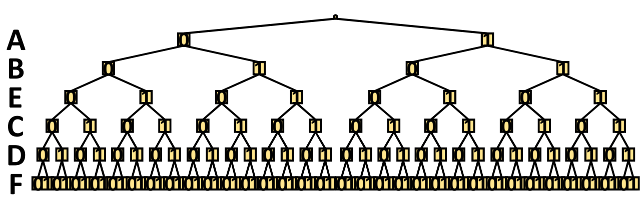


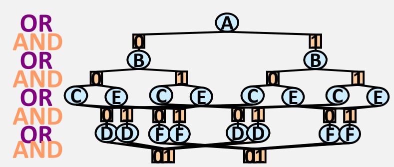
decomposition bounds



Heuristics
To guide AND/OR
search

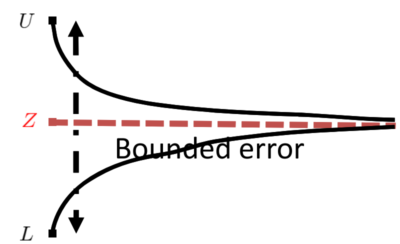
- Search → exploiting structure





Context minimal AND/OR search graph

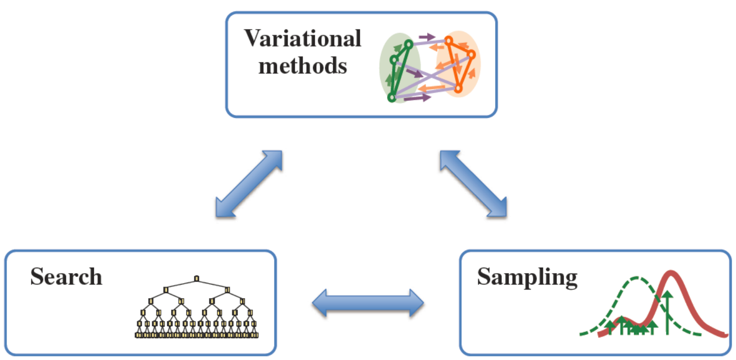
Anytime algorithms.

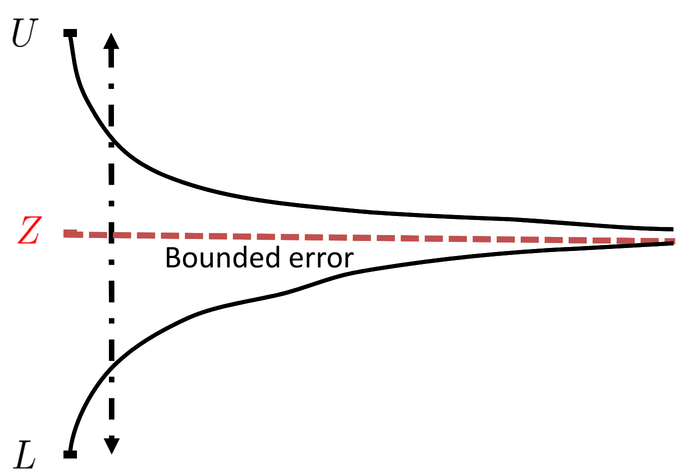


Sampling → replaces search when summation is hard

Outline

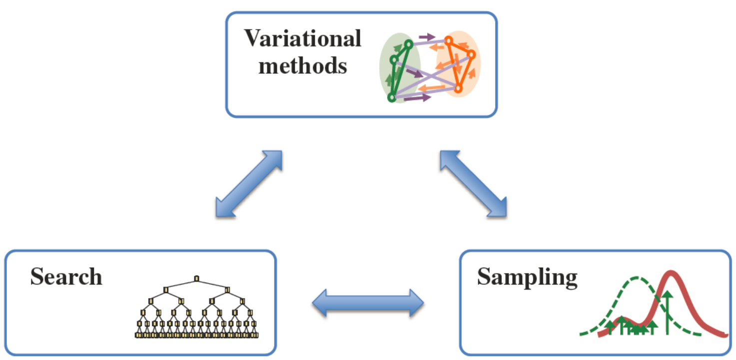
- Graphical models, The Marginal Map task,
- Exact Inference
- AND/OR search spaces
- Variational bounds as search heuristics
- Combining methods: Heuristic Search for Marginal Map
- Combining methods: Sampling
- Conclusion

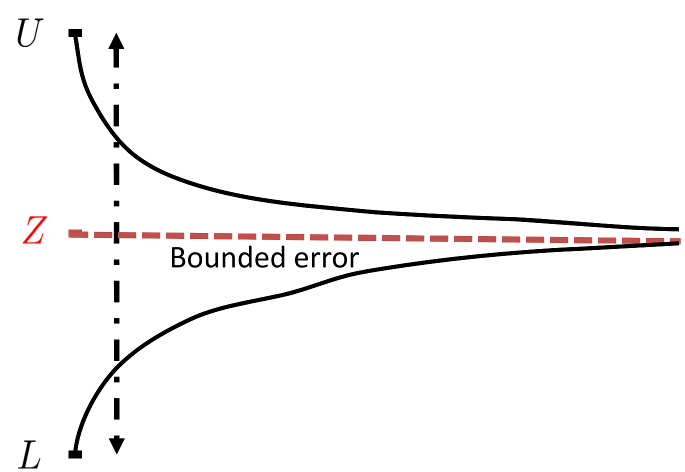




Outline

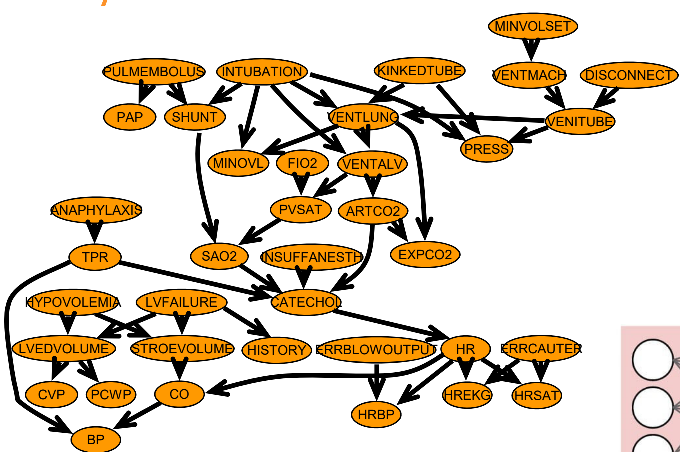
- Graphical models, The Marginal Map task,
- Exact Inference
- AND/OR search spaces
- Variational bounds as search heuristics
- Combining methods: Heuristic Search for Marginal Map
- Combining methods: Sampling
- Conclusion



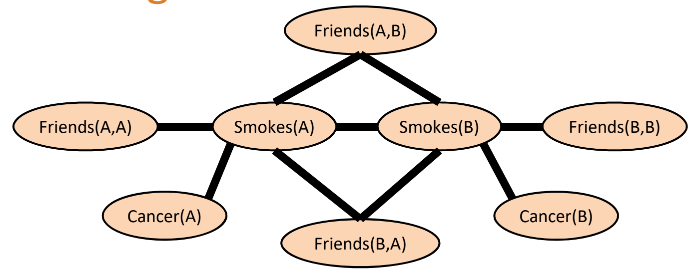


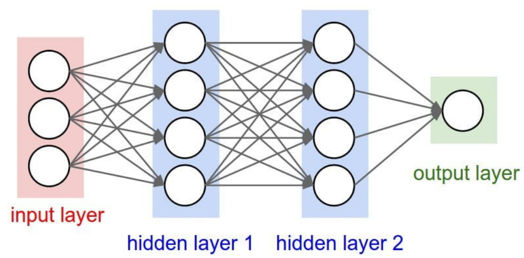
Overview: Graphical Models

Bayesian Networks

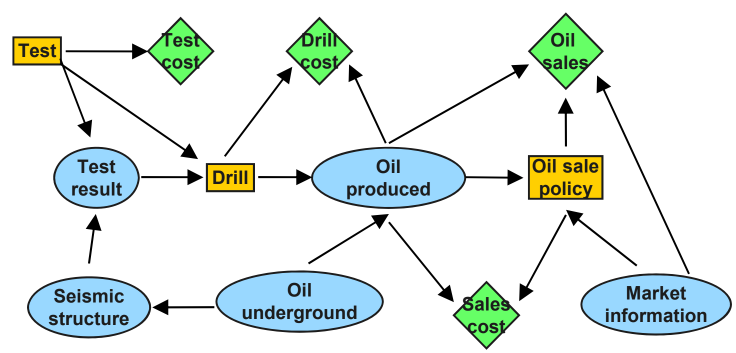


Markov Logic

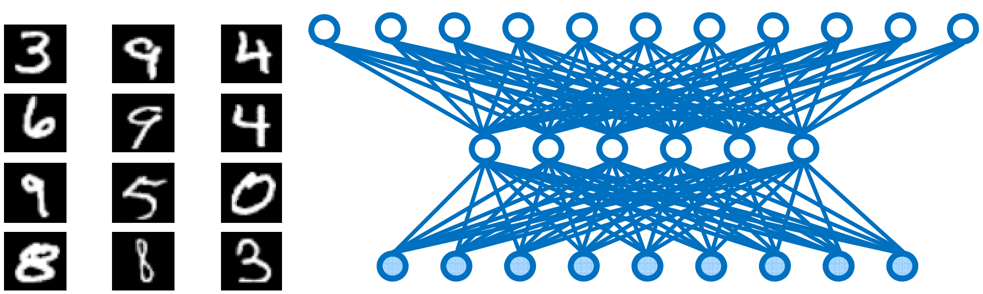




Influence Diagrams

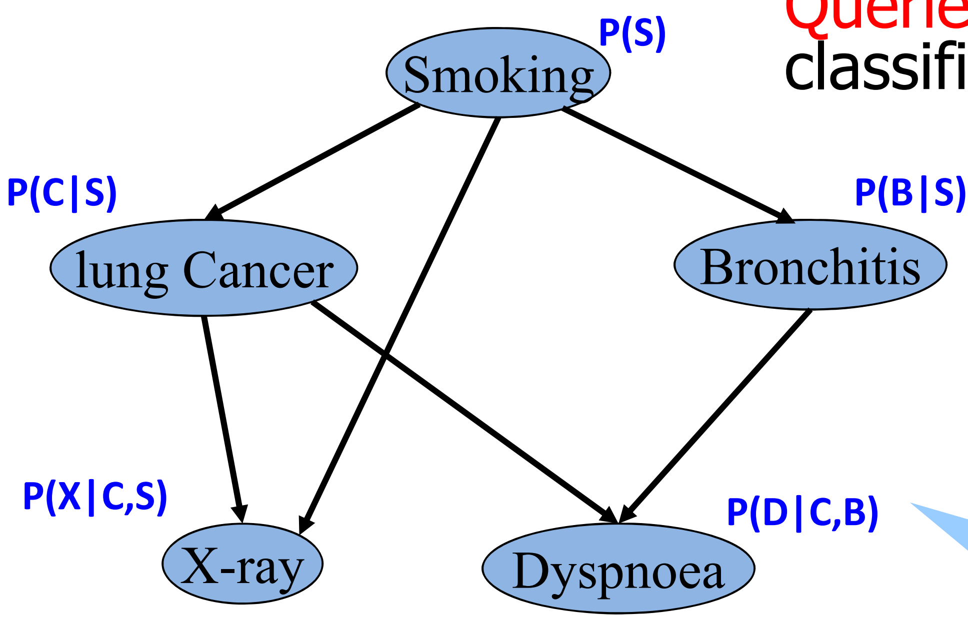


Deep Boltzmann Machines



Bayesian Networks (Pearl 1988)

Queries: prediction, diagnosis, classification, decision making



CPD:

C	B	P(D|C,B)	
0	0	0.1	0.9
0	1	0.7	0.3
1	0	0.8	0.2
1	1	0.9	0.1

P(S, C, B, X, D) = $P(S)\,P(C|S)\,P(B|S)\,P(X|C,S)\,P(D|C,B)$

P (lung cancer=yes | smoking=no, dyspnoea=yes) = ?

- P(D= 0) = $\sum_{S,L,B,X}$ P(S)· P(C|S)· P(B|S)· P(X|C,S)· P(D|C,B)
- MAP/MPE = $max_{S,L,B,X}$ P(S)· P(C|S)· P(B|S)· P(X|C,S)· P(D|C,B)

Graphical models

A ***graphical model*** consists of:

$X = \{X_1, \ldots, X_n\}$ -- variables

$D = \{D_1, \ldots, D_n\}$ -- domains

$F = \{f_{\alpha_1}, \ldots, f_{\alpha_m}\}$ -- functions

Operators:

combination operator

(sum, product, join, ...)

elimination operator

(projection, sum, max, min, ...)

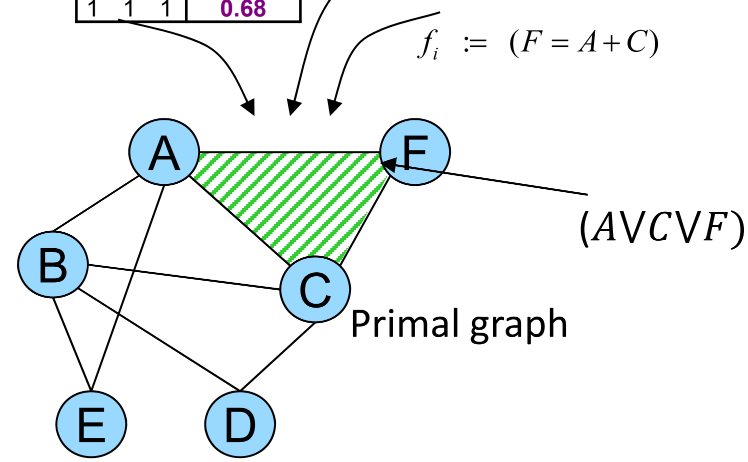
Conditional Probability Table (CPT)

A	C	F	P(F|A,C)
0	0	0	0.14
0	0	1	0.96
0	1	0	0.40
0	1	1	0.60
1	0	0	0.35
1	0	1	0.65
1	1	0	0.72
1	1	1	0.68

Relation

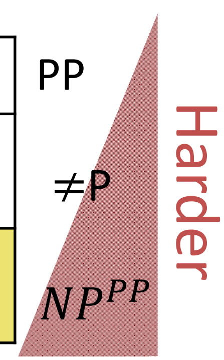
A	C	F
red	green	blue
blue	red	red
blue	blue	green
green	red	blue

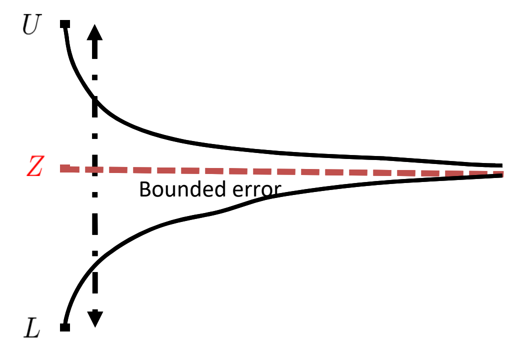
$f_i := (F = A + C)$



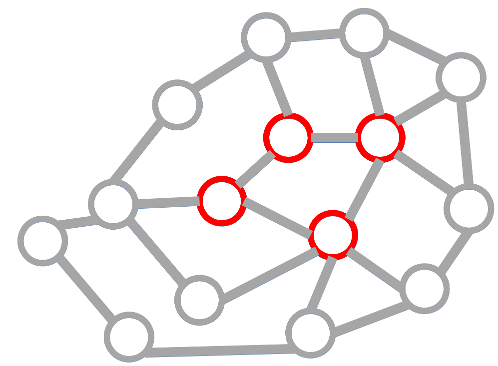
Types of queries:

Max-Inference (MAP)	$f(\mathbf{x}^*) = \max_{\mathbf{x}} \prod_{\alpha} f_\alpha(\mathbf{x}_\alpha)$
Sum-Inference (Z)	$Z = \sum_{\mathbf{x}} \prod_{\alpha} f_\alpha(\mathbf{x}_\alpha)$
Mixed-Inference	$f(\mathbf{x}_M^*) = \max_{\mathbf{x}_M} \sum_{\mathbf{x}_S} \prod_{\alpha} f_\alpha(\mathbf{x}_\alpha)$





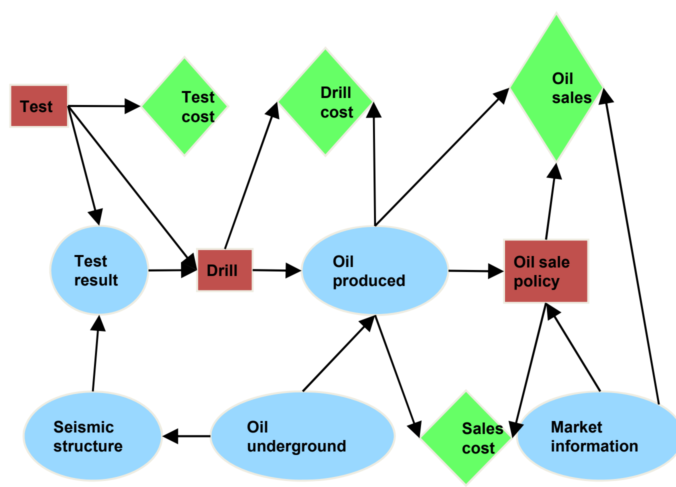
Why Marginal MAP?



- 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

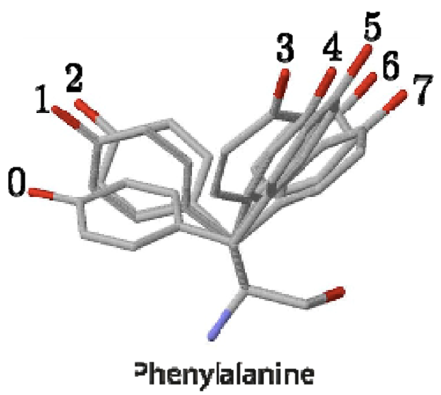
Influence diagrams
For planning

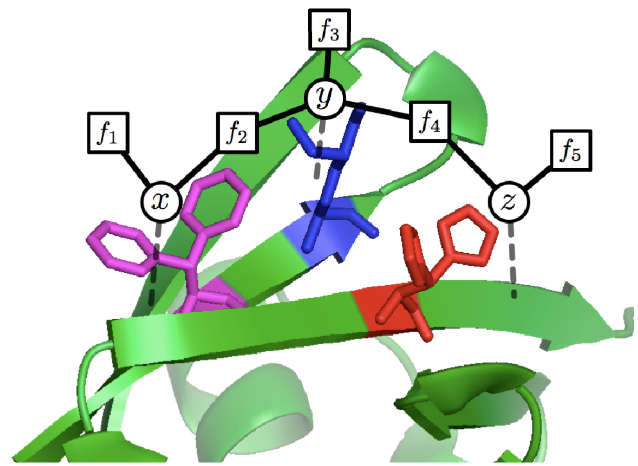


Probabilistic Graphical Models

- Examples & Tasks
  - Maximization (MAP): compute the most probable configuration

$$Obj = \underset{R_1 \ldots R_N}{\text{argmax}} \max_{C_1,\ldots,C_N} \prod_{E_i \in E_{sb}} e^{-\frac{E_i(R_i, C_i)}{\mathcal{R}T}} \cdot \prod_{E_{ij} \in E_{pw}} e^{-\frac{E_{ij}(R_i, C_i, R_j, C_j)}{\mathcal{R}T}}$$





[Yanover & Weiss 2002]

  - Mixed Max-sum (Marginal Map): compute the most likely marginal

$$Obj = \underset{R_1 \ldots R_N}{\text{argmax}} \sum_{C_1,\ldots,C_N} \prod_{E_i \in E_{sb}} e^{-\frac{E_i(R_i, C_i)}{\mathcal{R}T}} \cdot \prod_{E_{ij} \in E_{pw}} e^{-\frac{E_{ij}(R_i, C_i, R_j, C_j)}{\mathcal{R}T}}$$

Probabilistic Reasoning Problems

- Tasks:

Constraint Satisfaction/Satisfiability

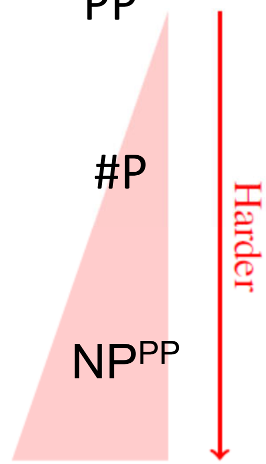
NP

Max-Inference:	$f(x^*) = \max_x \prod_\alpha f_\alpha(x_\alpha)$
Sum-Inference:	$Z = \sum_x \prod_\alpha f_\alpha(x_\alpha)$
Mixed-Inference (MMAP):	$f_M(x_M^*) = \max_{x_M} \sum_{x_S} \prod_\alpha f_\alpha(x_\alpha)$
Mixed-Inference (MEU):	$\text{MEU} = \max_{D_1,\ldots,D_m} \sum_{X_1,\ldots X_n} (\prod_{P_i \in P} P_i) \times (\sum_{r_i \in R} r_i)$

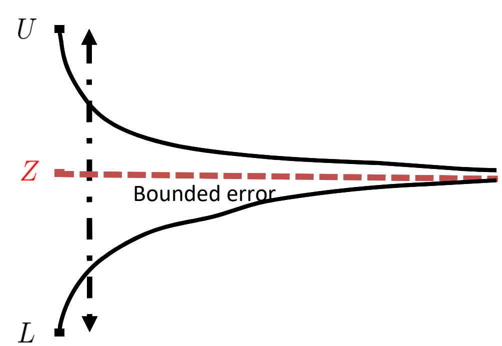
PP

#P

NP^{PP}



Counting optimal solutions



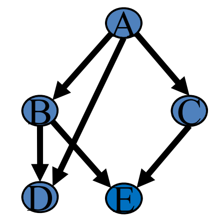
Variable-elimination allows exploiting the structure

Inference, message-passing

Over variable-based models, over graphical mdeols.

Query 1: 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

bucket B: P(b|a) P(d|b,a) P(e|b,c)

bucket C: P(c|a) $\lambda_{B \to C}$ ***(a, d, c, e)***

bucket D: $\lambda_{C \to D}$ ***(a, d, e)***

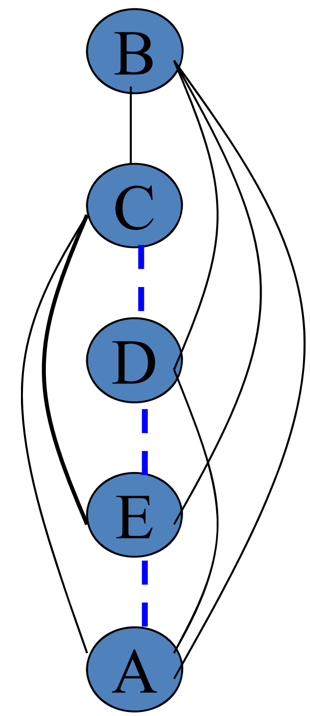
bucket E: e=0 $\lambda_{D \to A}$ ***(a, e)***

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

P(e=0)

P(a|e=0)

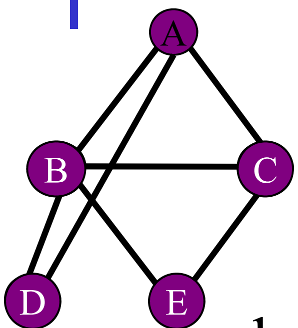
W*=4
"induced width"
(max clique size)



Complexity time and space O(nk^{w*+1})

Query 2: Finding MAP by Bucket Elimination

Algorithm BE-mpe (Dechter 1996, Bertele and Bri



$$MPE = \max_{a,e,d,c,b} P(a)P(c \mid a)P(b \mid$$

$$= \max_b P(b \mid a) \cdot P(d \mid b,a) \cdot P(e \mid b,c)$$

$$\max_X \prod$$

bucket B: P(b|a) P(d|b,a) P(e|b,c)

bucket C: P(c|a) $h_{B \to C}(a, d, c, e)$

bucket D: $h_{C \to D}(a, d, e)$

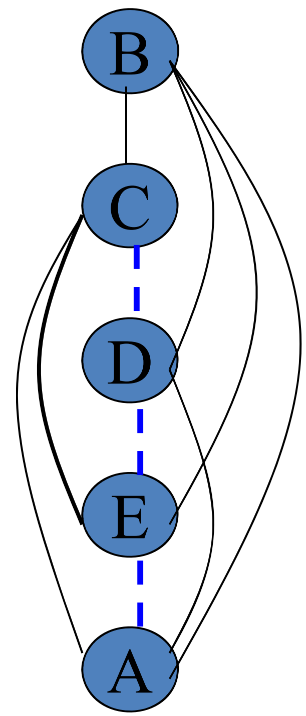
bucket E: e=0 $h_{D \to E}(a, e)$

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

OPT

W*=4

"induced width" (max clique size)



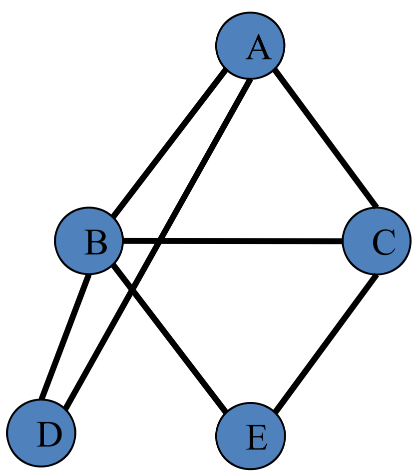
Complexity of Bucket Elimination;

Bucket Elimination is **time** and **space**

$$O\left(r\exp(\,w^*(d)\,)\right)$$

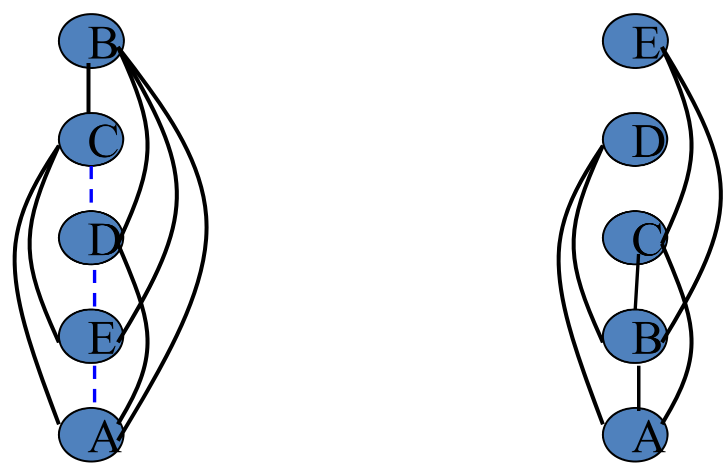
$w^*(d)$ – the induced width of graph along ordering d

r = number of functions



"Moral" graph

The effect of the ordering:

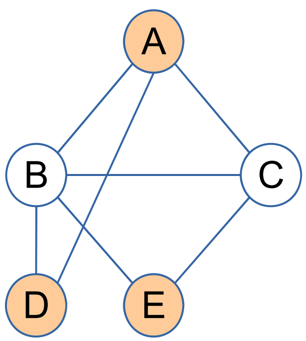


$w^*(d_1) = 4$ $w^*(d_2) = 2$

Finding the smallest induced width is hard!

Why is MMAP Harder for Inference (BE)?

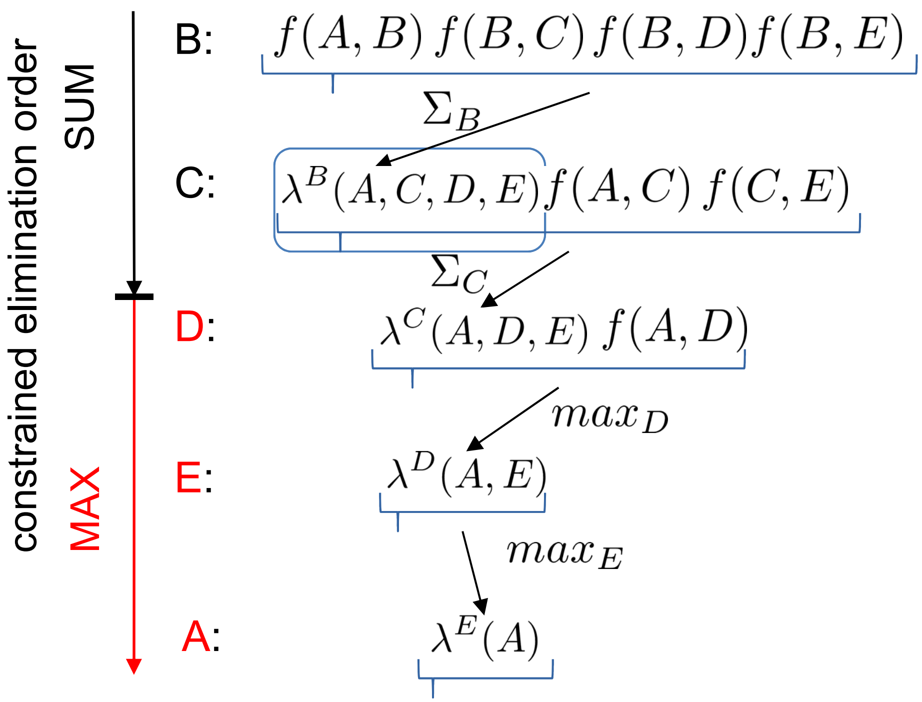
Let's apply Bucket-elimination: Complexity is exponential in the *constrained* induced-width



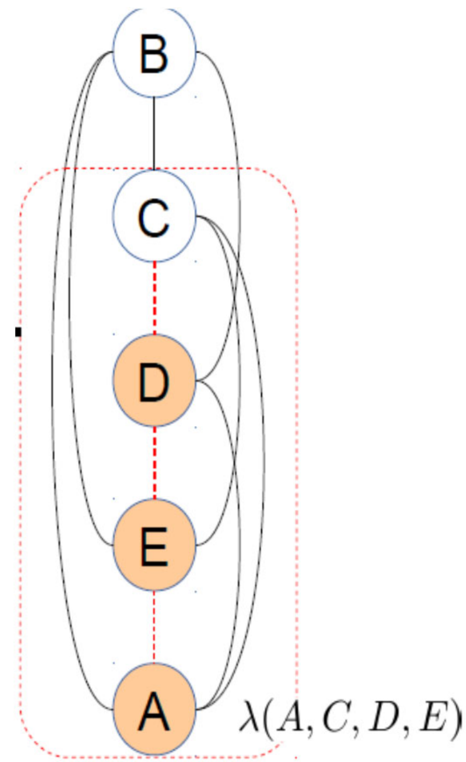
$$\mathbf{X}_M = \{A, D, E\}$$
$$\mathbf{X}_S = \{B, C\}$$

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

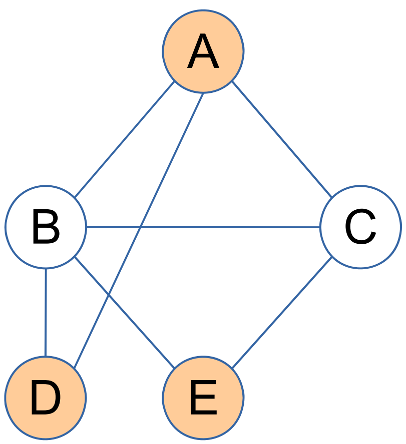
$$P(X) = \prod_j f_j$$



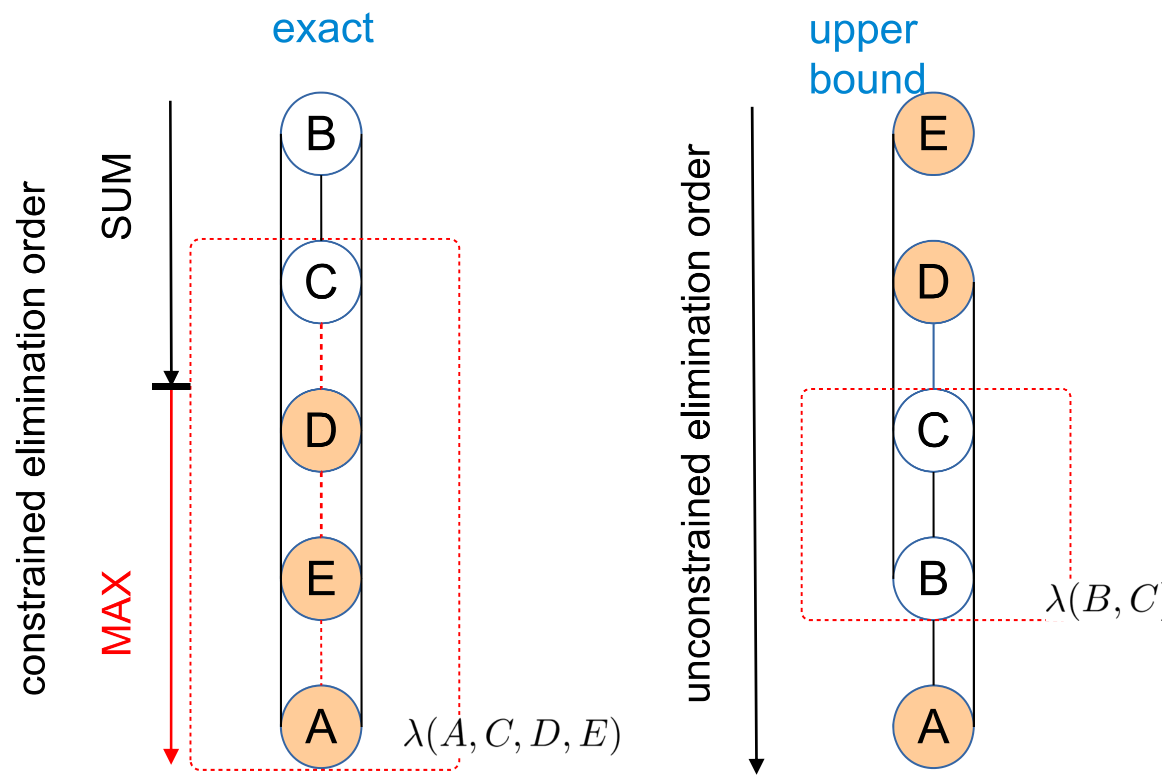
MAP* is the marginal MAP value



Why is MMAP Harder for Inference (BE)?



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



$w^* = 4$ $w^* = 2$

In practice, constrained induced is much larger!

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

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

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

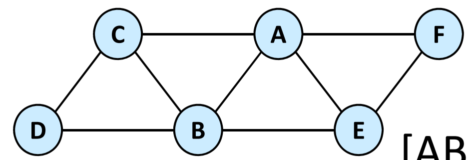
B	C	f_5
0	0	0
0	1	1
1	0	2
1	1	4

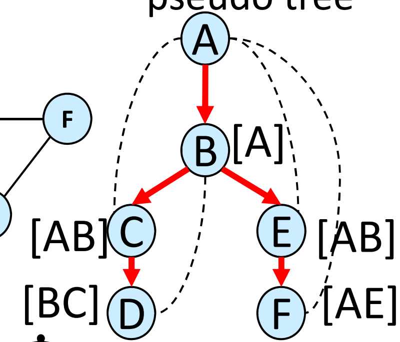
B	D	f_6
0	0	4
0	1	2
1	0	1
1	1	0

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

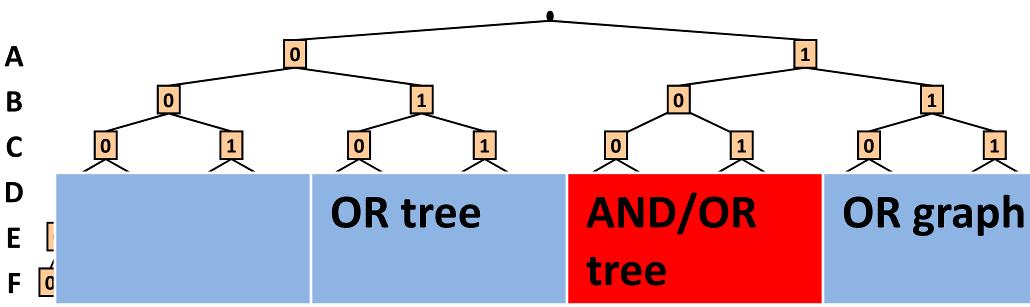
C	D	f_8
0	0	1
0	1	4
1	0	0
1	1	0

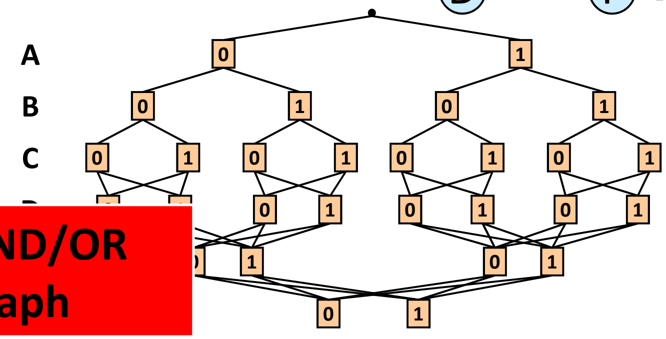
E	F	f_9
0	0	1
0	1	0
1	0	0
1	1	2





[Dechter & Mateescu, 2007]

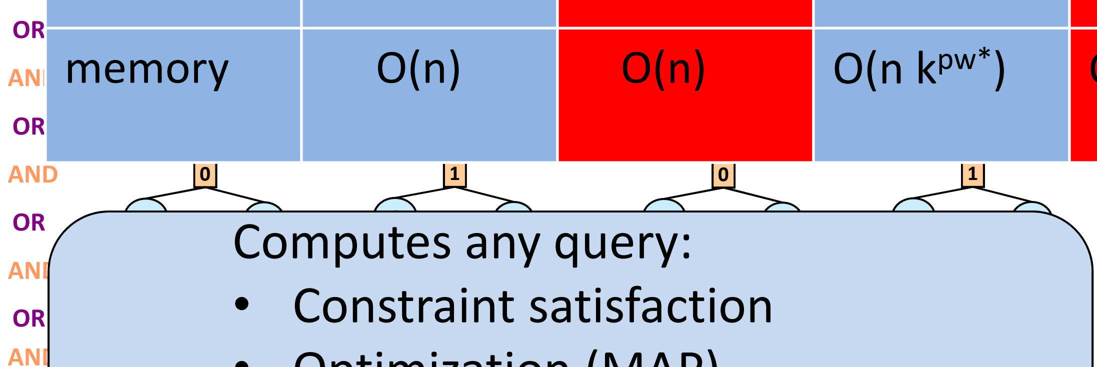


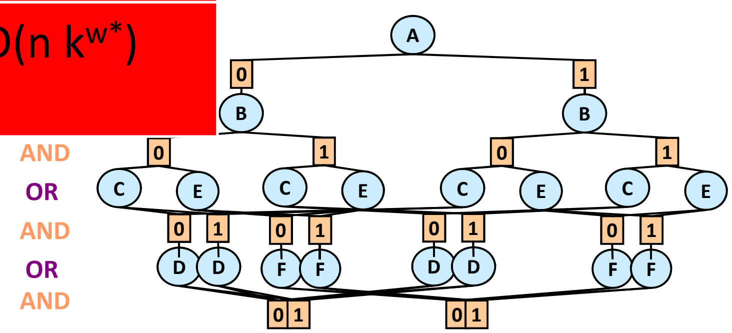


Context minimal OR search graph

28 nodes

	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 AND/OR search graph

18 AND nodes

Computes any query:

- Constraint satisfaction
- Optimization (MAP)
- Marginal (P(e))
- Marginal map

Any query is best computed
Over the c-minimal AO search space

Cost of a Solution Tree

$P(E \mid A,B)$

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

$P(B \mid A)$

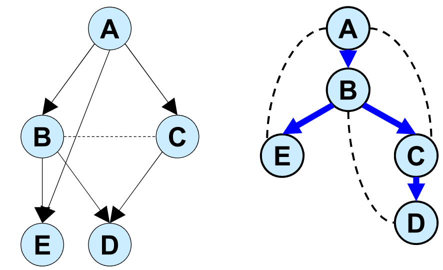
A	B=0	B=1
0	.4	.6
1	.1	.9

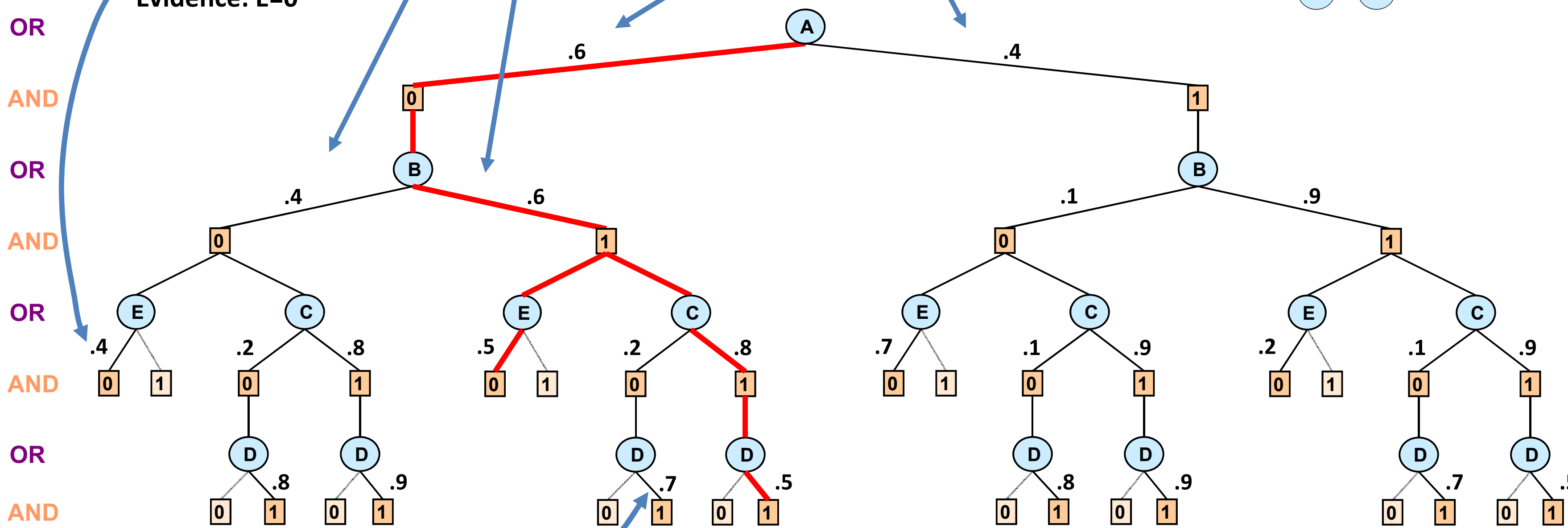
$P(C \mid A)$

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

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 · 0.6 · 0.5 · 0.8 · 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

$P(B \mid A)$

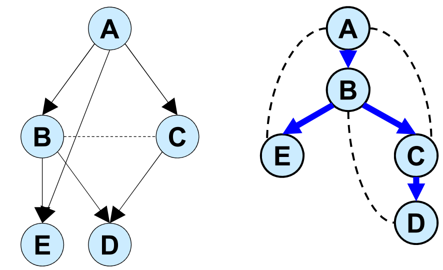
A	B=0	B=1
0	.4	.6
1	.1	.9

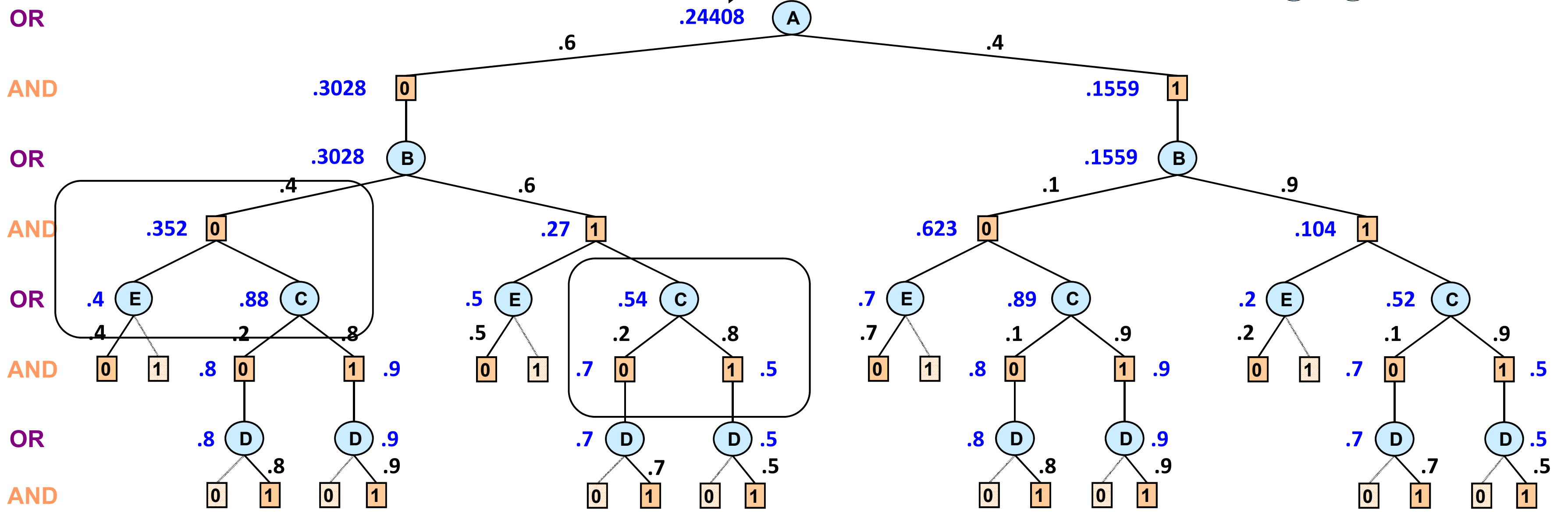
$P(C \mid A)$

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)

$P(E \mid A,B)$

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

$P(B \mid A)$

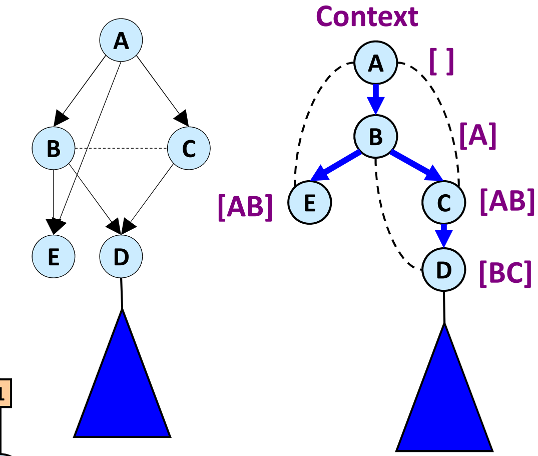
A	B=0	B=1
0	.4	.6
1	.1	.9

$P(C \mid A)$

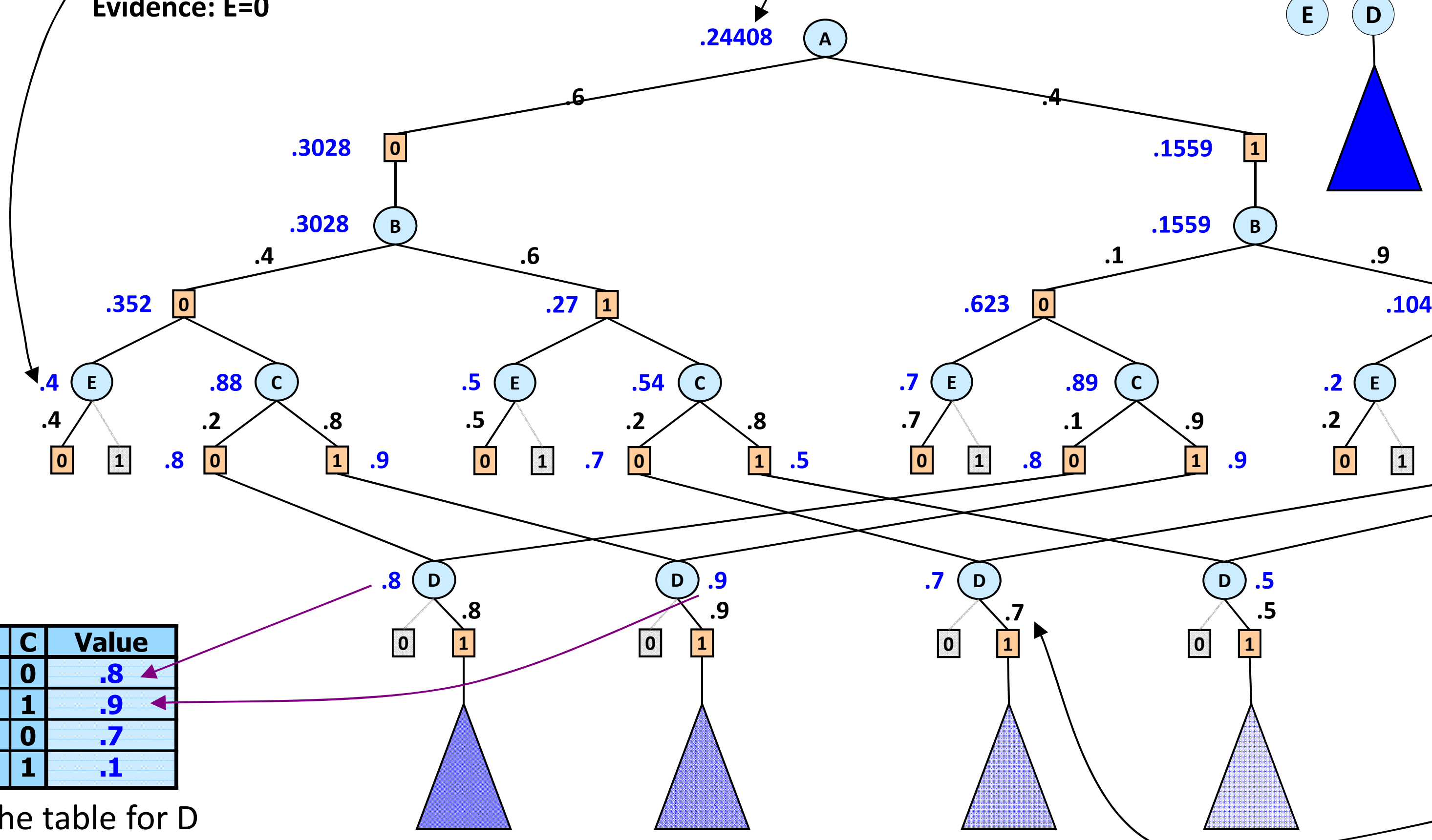
A	C=0	C=1
0	.2	.8
1	.7	.3

$P(A)$

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

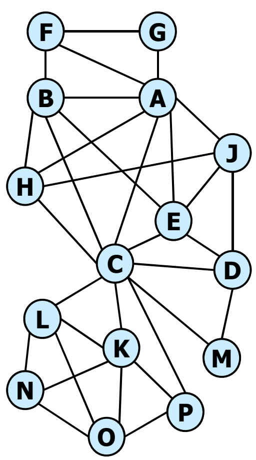
Cache table for D

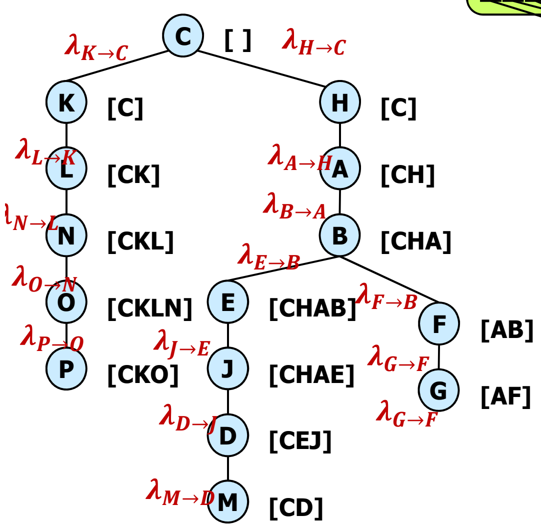
$P(D \mid B,C)$

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

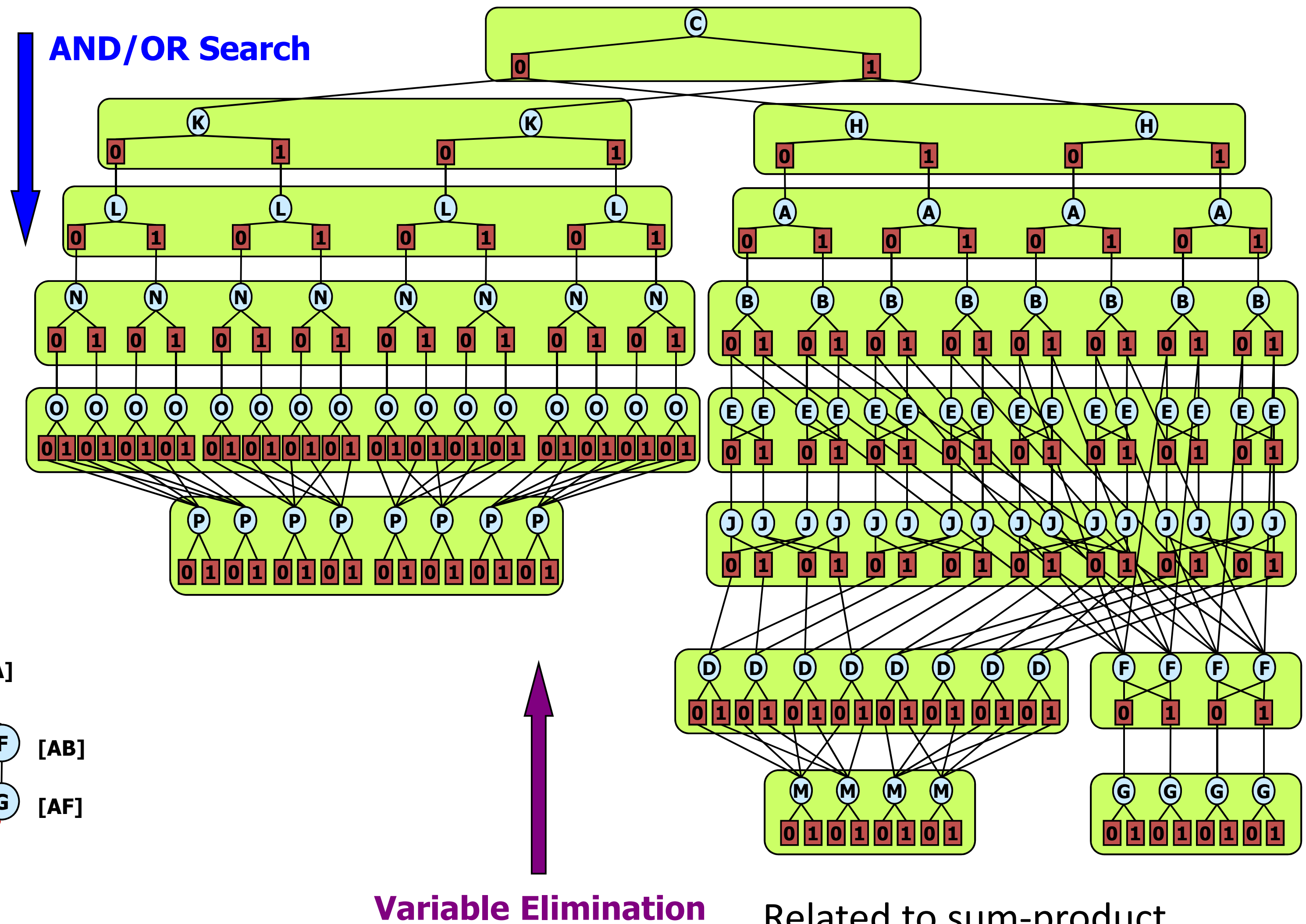
Evidence: D=1

AND/OR Search and Variable Elimination





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

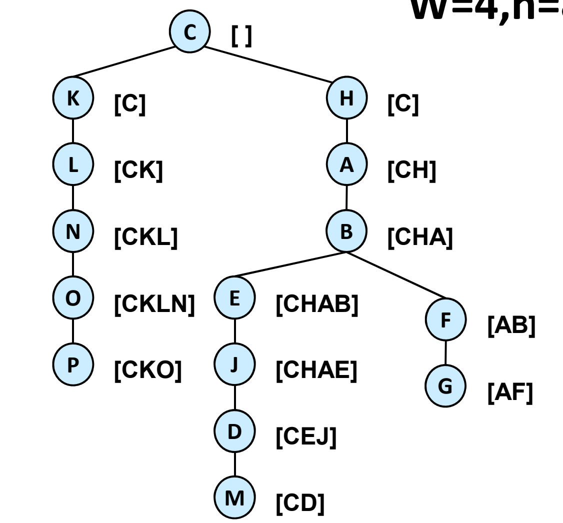


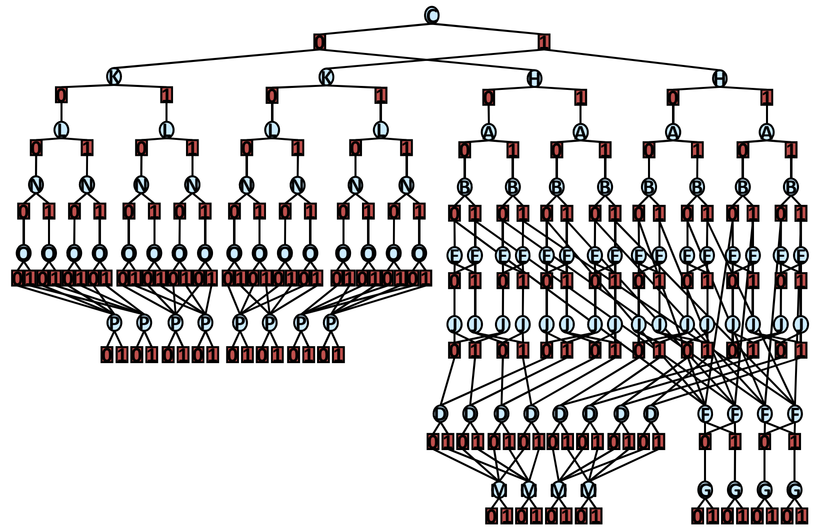
Related to sum-product Networks or Arithmetic circuit

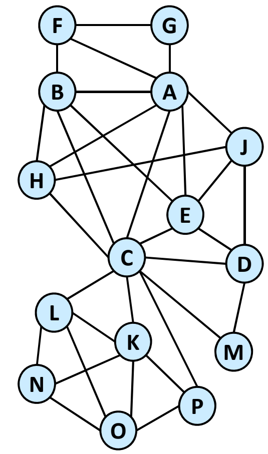
The Impact of the Pseudo-Tree

N=15

W=4,h=8

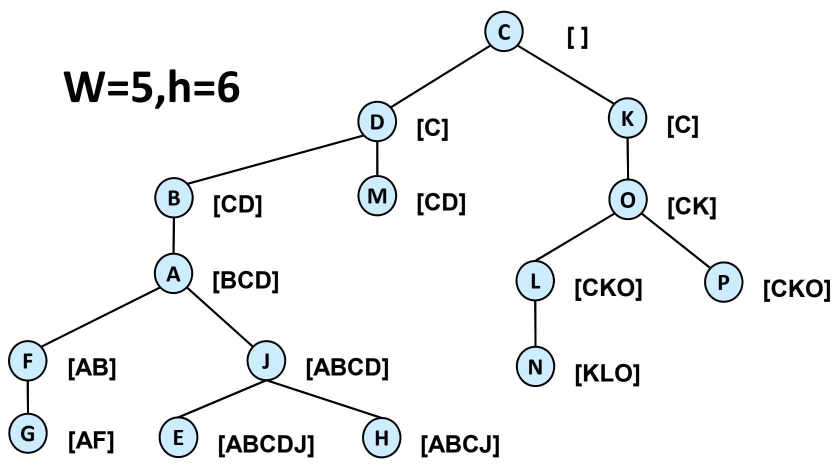


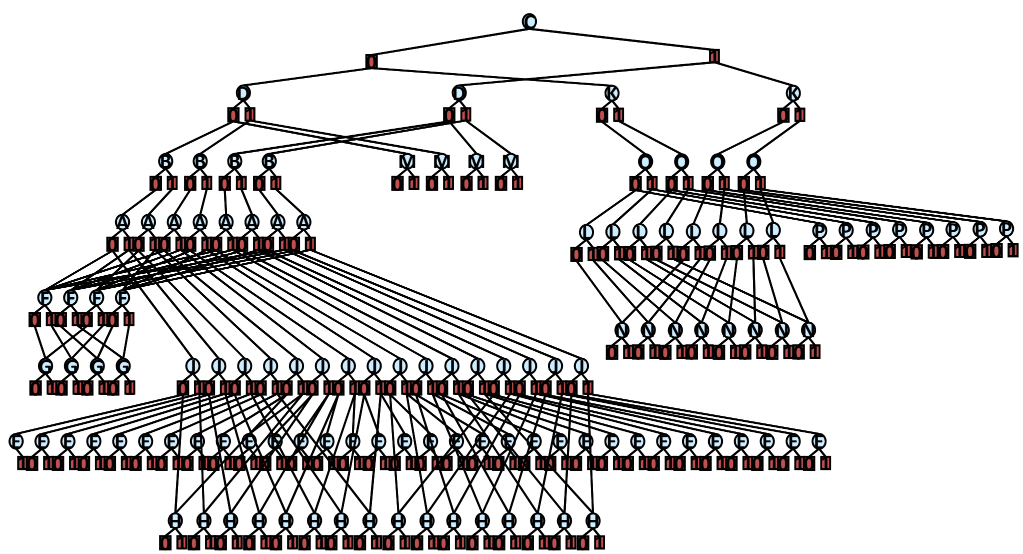




(C K H A B E J L N O D P M F G)

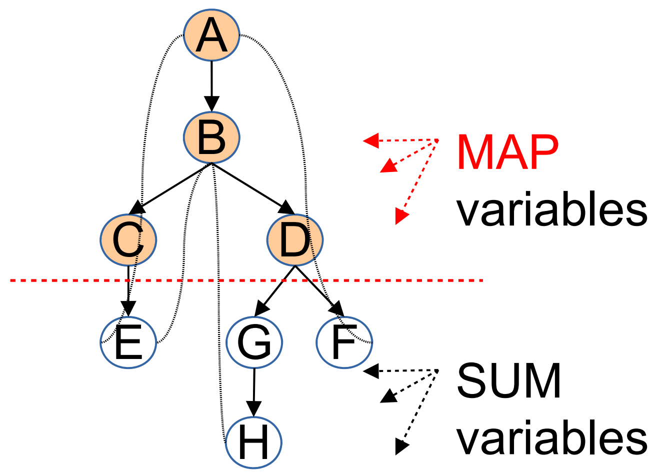
W=5,h=6



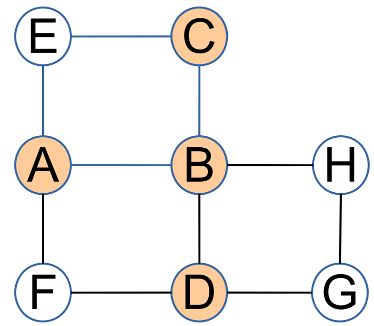


(C D K B A O M L N P J H E F G)

AND/OR Search for Marginal MAP



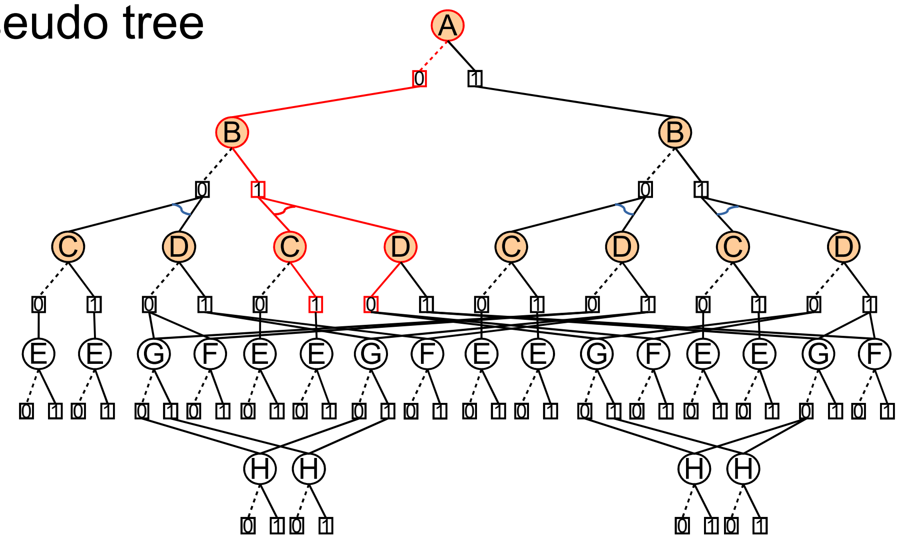
constrained pseudo tree



primal graph

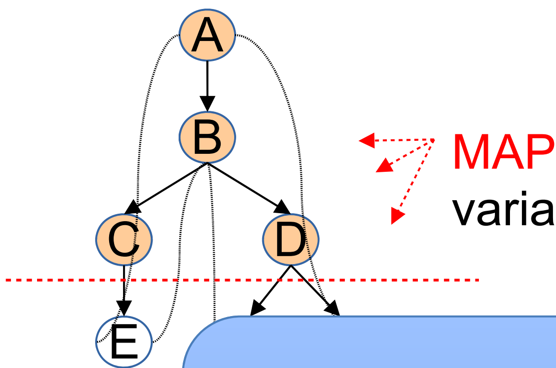
$X_M = \{A, B, C, D\}$

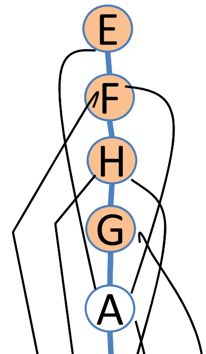
$X_S = \{E, F, G, H\}$

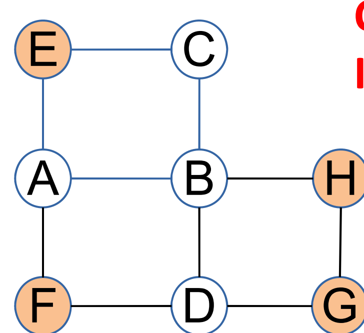


- Node types
  - OR (MAP): max
  - OR (SUM): sum
  - AND: multiplication

AND/OR Search for Marginal MAP



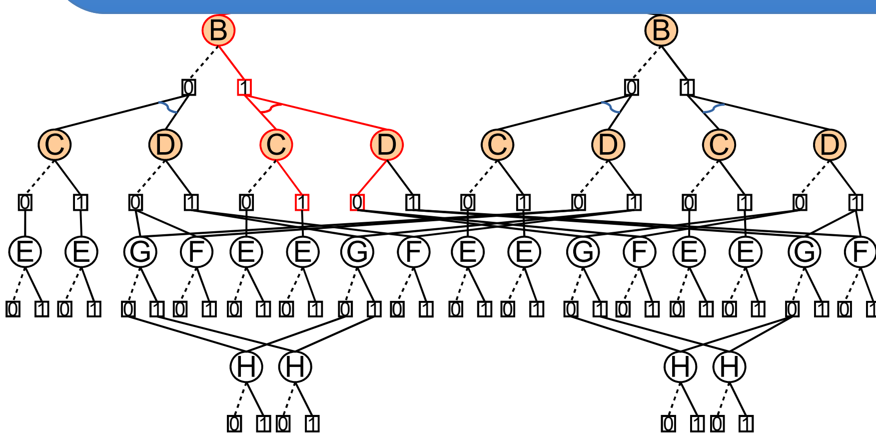


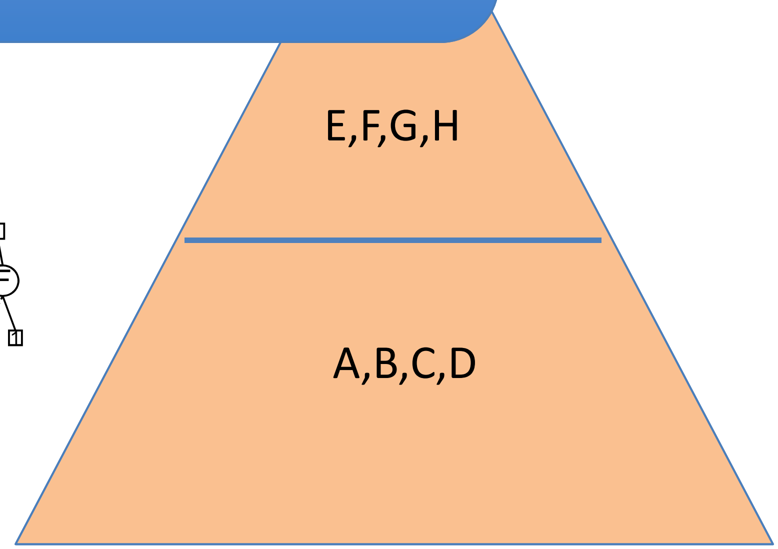


C-induced-width = 6
Induced-width = 2

const
pseu

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





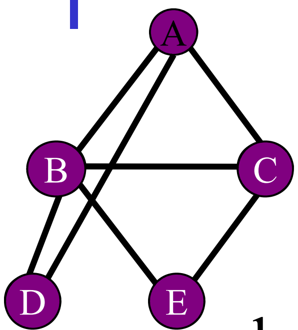
[Marinescu, Dechter and Ihler, 2014]

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

Generating Heuristic Using Relaxed Tractable Models

Query 2: Finding MAP by Bucket Elimination

Algorithm BE-mpe (Dechter 1996, Bertele and Briochi, 1977)



$$MPE = \max_{a,e,d,c,b} P(a)P(c\,|\,a)P(b\,|\,a)P(d\,|\,a,b)P(e\,|\,b,c)$$

$$\max_X \prod$$

bucket B: P(b|a) P(d|b,a) P(e|b,c)

bucket C: P(c|a) $h_{B \to C}(a, d, c, e)$

bucket D: $h_{C \to D}(a, d, e)$

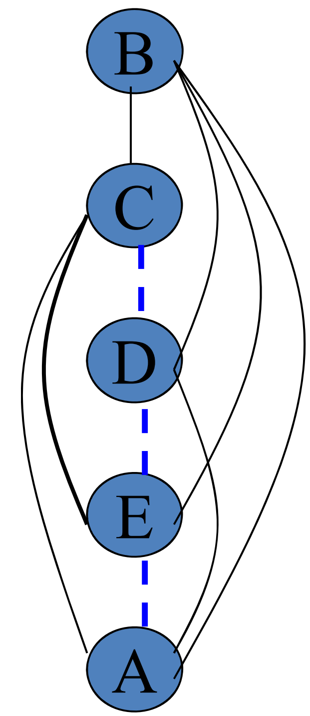
bucket E: e=0 $h_{D \to E}(a, e)$

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

OPT

W*=4

"induced width"
(max clique size)



Query 2: Decoding the MAP-Tuple

5. $b' = \arg\max_b P(b \mid a') \times P(d' \mid b, a') \times P(e' \mid b, c')$

4. $c' = \arg\max_c P(c \mid a') \times h^B(a', d', c, e')$

3. $d' = \arg\max_d h^C(a', d, e')$

2. $e' = 0$

1. $a' = \arg\max_a P(a) \cdot h^E(a)$

B:	$P(b\|a)$ $P(d\|b,a)$ $P(e\|b,c)$	
C:	$P(c\|a)$	$h^B(a, d, c, e)$
D:		$h^C(a, d, e)$
E:	$e=0$	$h^D(a, e)$
A:	$P(a)$	$h^E(a)$

Return (a', b', c', d', e')

Mini-Bucket Approximation

For optimization

Split a bucket into mini-buckets —> bound complexity

bucket (X) =

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

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

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

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

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

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: $f(a,b)\ f(b,c)$ | $f(b,d)\ f(b,e)$

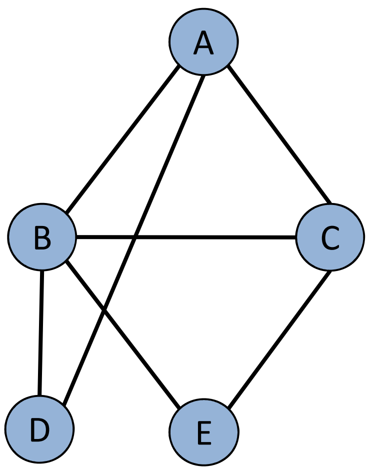
bucket C: $f(c,a)\ f(c,e)\ \lambda_{B\to C}(a,c)$

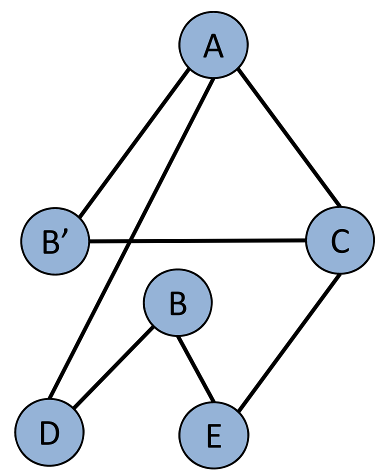
bucket D: $f(a,d)\ \ \lambda_{B\to D}(d,e)$

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

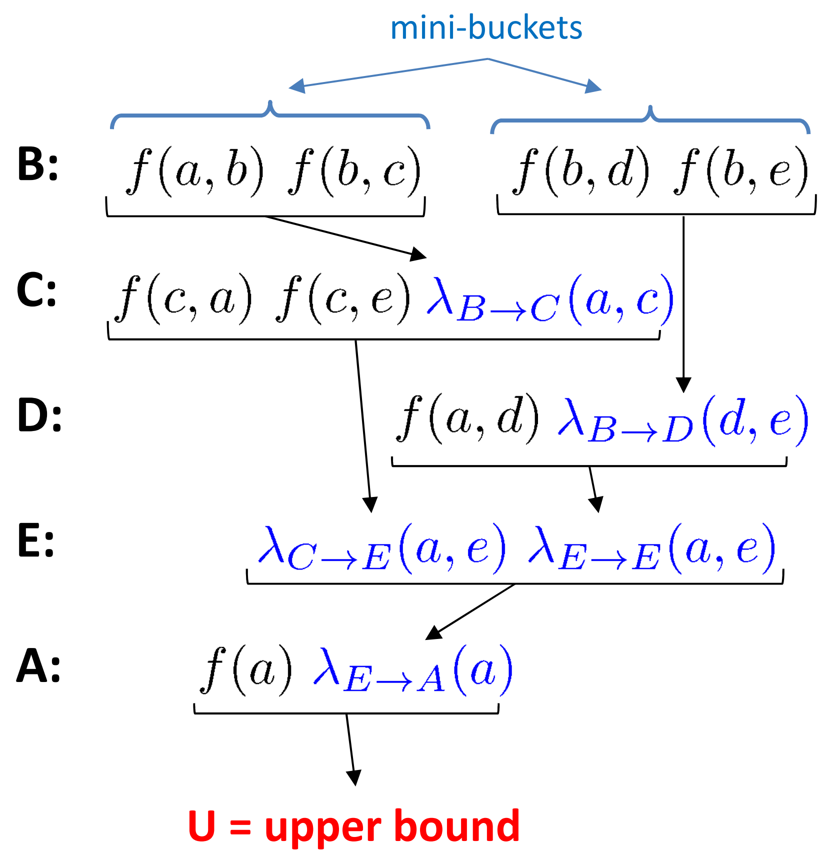
$$\mathbf{b}^* = \arg\max_b f(a^*, b) \cdot f(b, c^*) \cdot f(b, d^*) \cdot f(b, e^*)$$

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

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

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

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

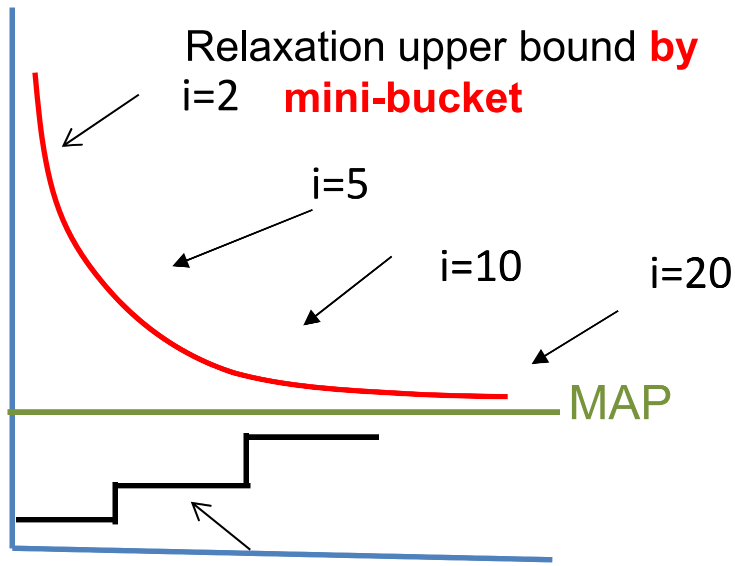


Greedy configuration = lower bound

Properties of Mini-Bucket Eliminaton

(For optimization)

- Bounding from above and below

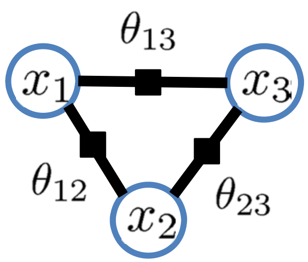


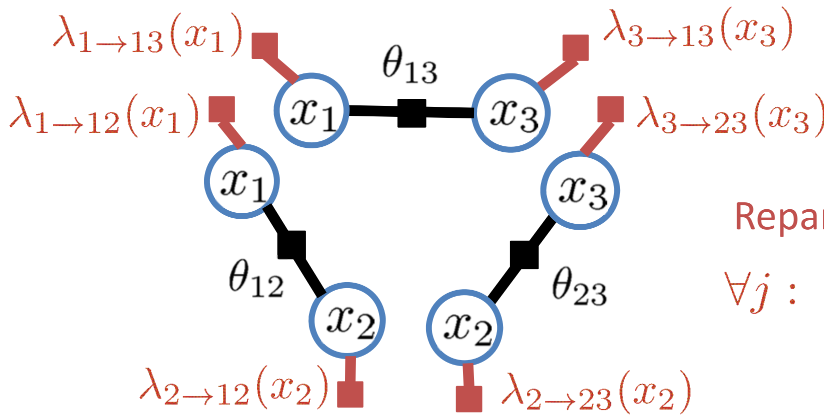
Consistent solutions (greedy search)

- 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:

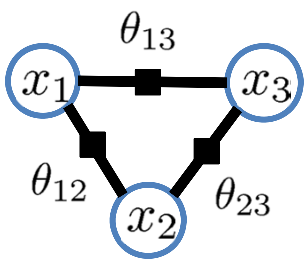
$$\forall j: \sum_{\alpha \ni j} \lambda_{j\to\alpha}(x_j) = 0$$

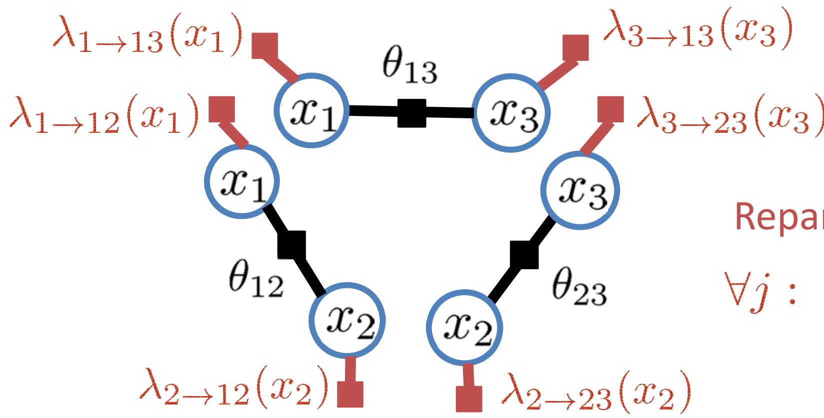
$$\log f(\mathbf{x}^*) = \max_{\mathbf{x}} \sum_{\alpha} \theta_\alpha(\mathbf{x}_\alpha) \quad \le \min_{\{\lambda_{i\to\alpha}\}} \sum_{\alpha} \max_{\mathbf{x}_\alpha} \left[\theta_\alpha(\mathbf{x}_\alpha) + \sum_{i\in\alpha} \lambda_{i\to\alpha}(x_i) \right]$$

- 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:

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

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

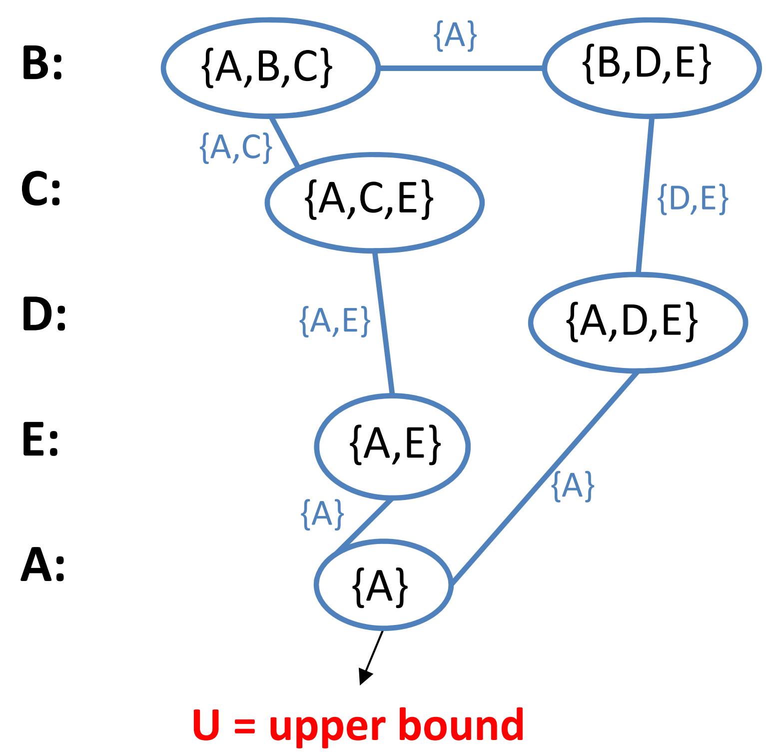
- Many names for the same class of bounds
  - Dual decomposition [Komodakis et al. 2007]
  - TRW, MPLP [Wainwright et al. 2005; Globerson & Jaakkola 2007]
  - Soft arc consistency [Cooper & Schiex 2004]
  - Max-sum diffusion [Warner 2007]

Mini-Bucket with Moment-Matching

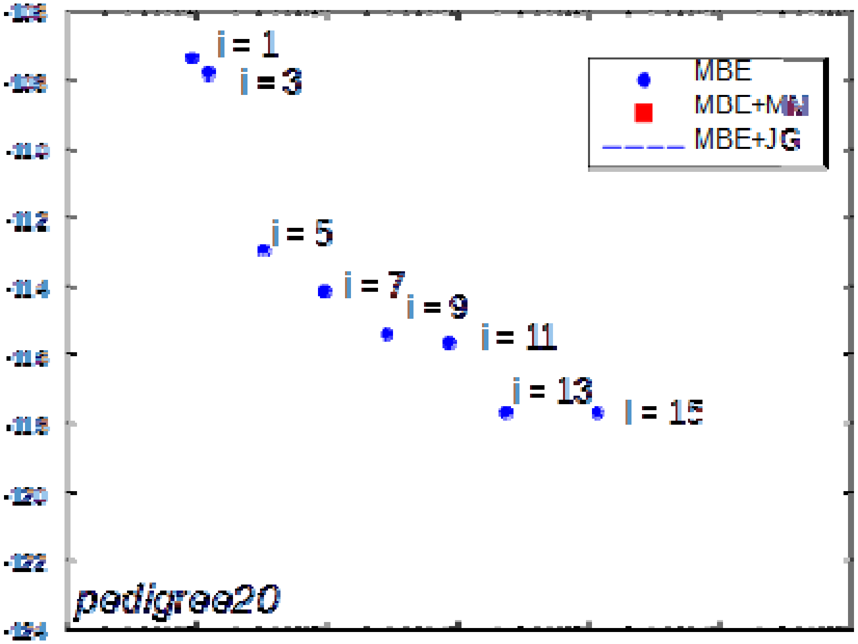
- Downward pass as cost shifting
- Can also do cost shifting within mini-buckets:
  "Join graph" message passing
- "Moment-matching" version:
  One message exchange within each bucket, during downward sweep
- Optimal bound defined by cliques ("regions") and cost-shifting f'n scopes ("coordinates")

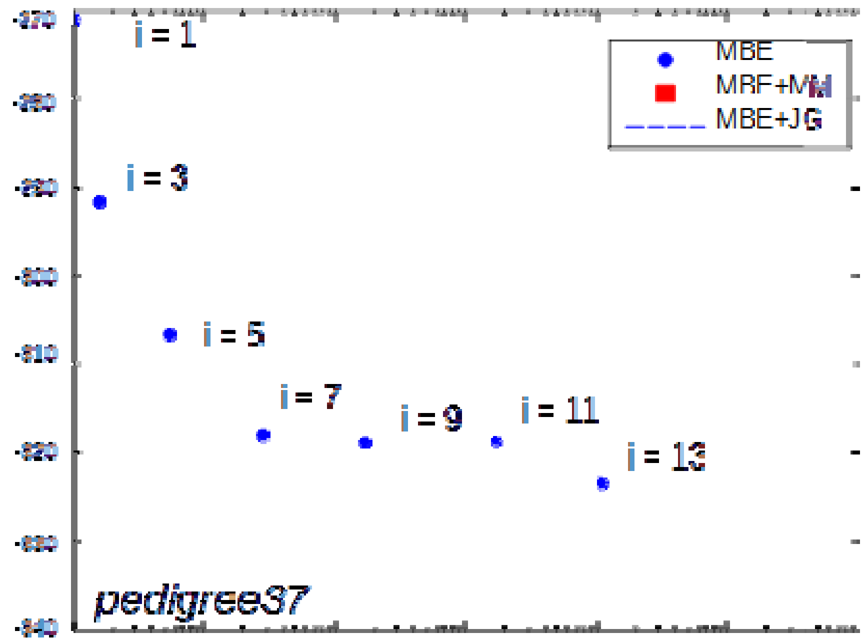
[Ihler et al. 2012]

Join graph:



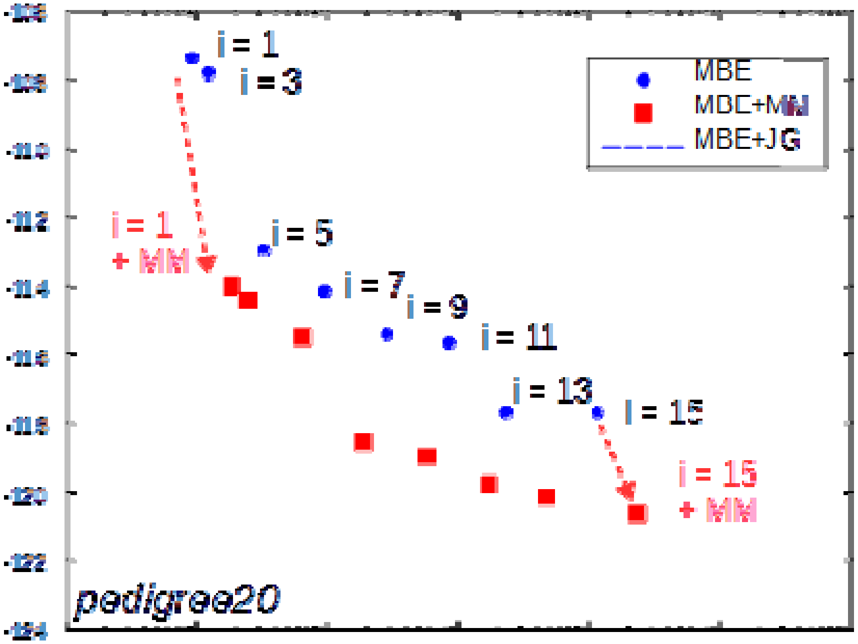
Anytime Approximation

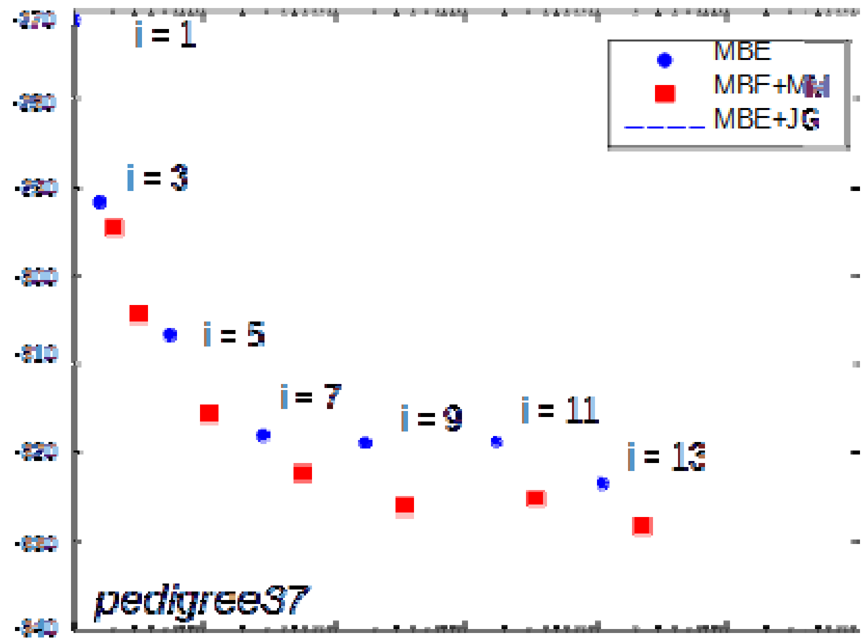




- Can tighten the bound in various ways
  - Cost-shifting (improve consistency between cliques)
  - Increase i-bound (higher order consistency)
- Simple moment-matching step improves bound significantly

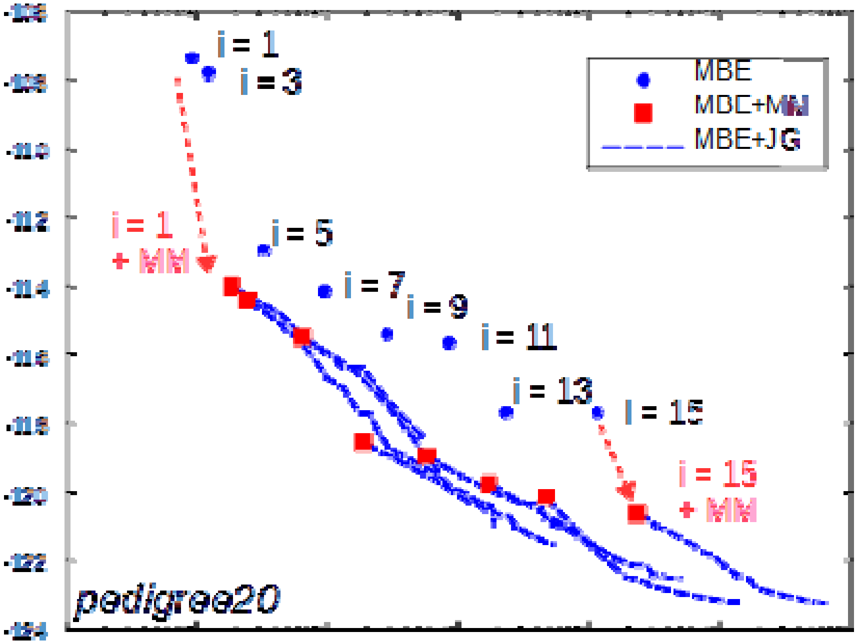
Anytime Approximation

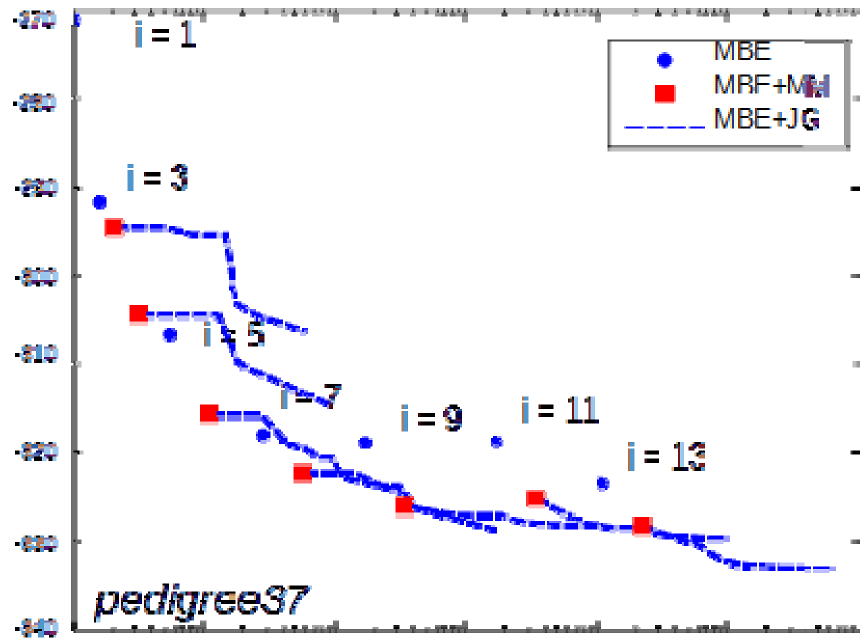




- Can tighten the bound in various ways
  - Cost-shifting (improve consistency between cliques)
  - Increase i-bound (higher order consistency)
- Simple moment-matching step improves bound significantly

Anytime Approximation

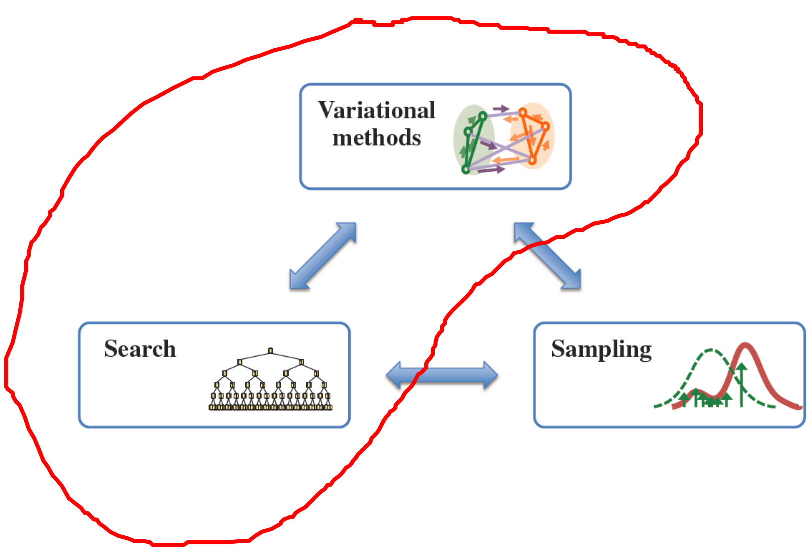


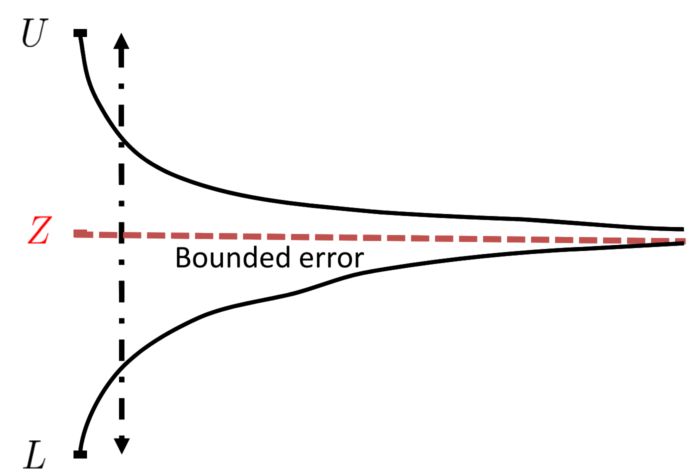


- Can tighten the bound in various ways
  - Cost-shifting (improve consistency between cliques)
  - Increase i-bound (higher order consistency)
- Simple moment-matching step improves bound significantly

Outline

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

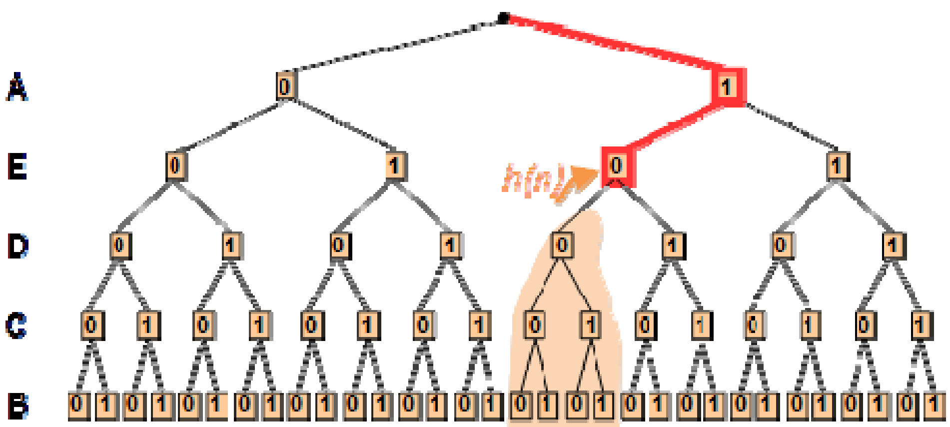


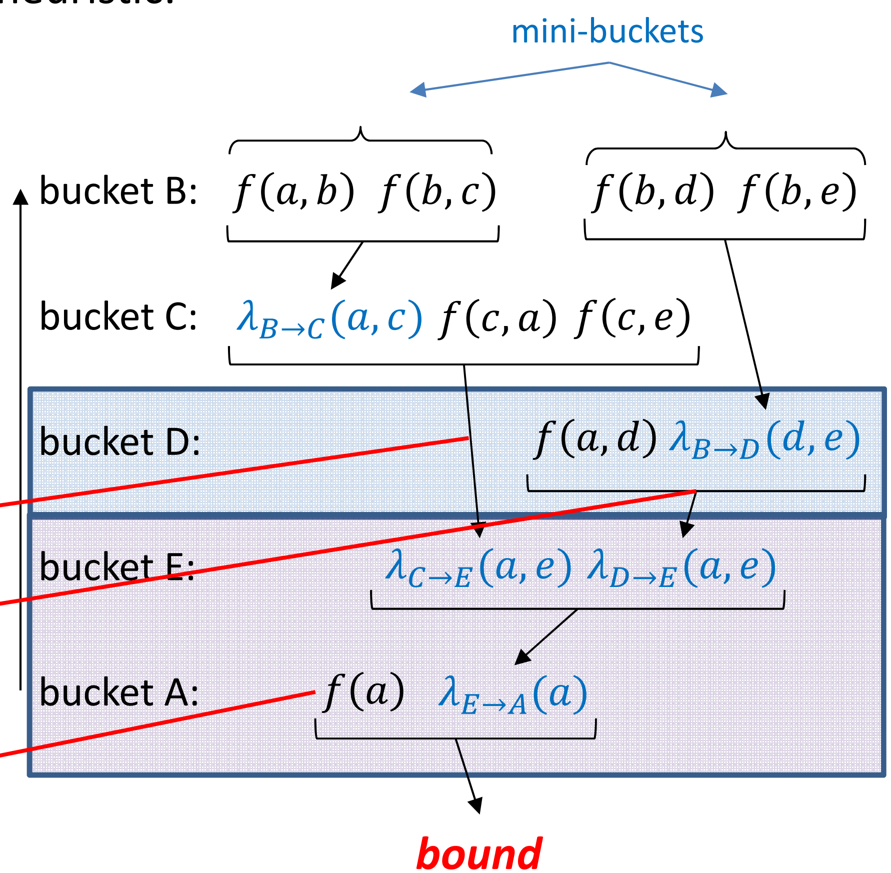


Search Aided by Variational Heuristics

[Kask, Dechter, AIJ 2001]

Given a partial assignment, $[\hat{a} = 1, \hat{e} = 0]$
(weighted) mini-bucket gives an admissible heuristic:





"cost to go":

$$\tilde{h}(\hat{a}, \hat{e}, D) = \lambda_{C\to E}(\hat{a}, \hat{e}) + f(\hat{a}, D) + \lambda_{B\to D}(D, \hat{e})$$

"cost so far":

$$g(\hat{a}, \hat{e}, D) = f(A = \hat{a})$$

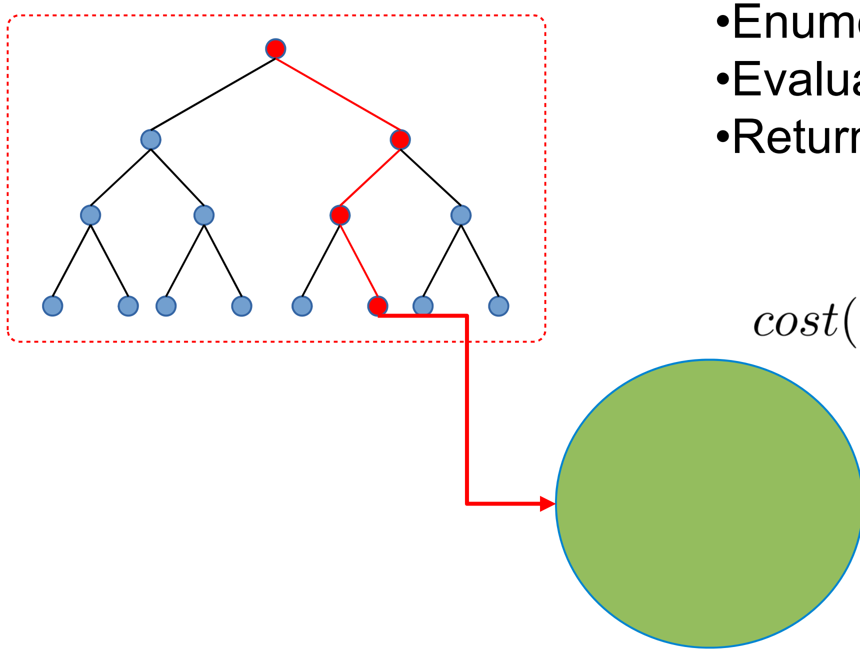
For MAP, marginal map and partition function

Why is MMAP Harder for Search?

Brute-Force Search

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

MAP



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

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

$$\#P - complete$$

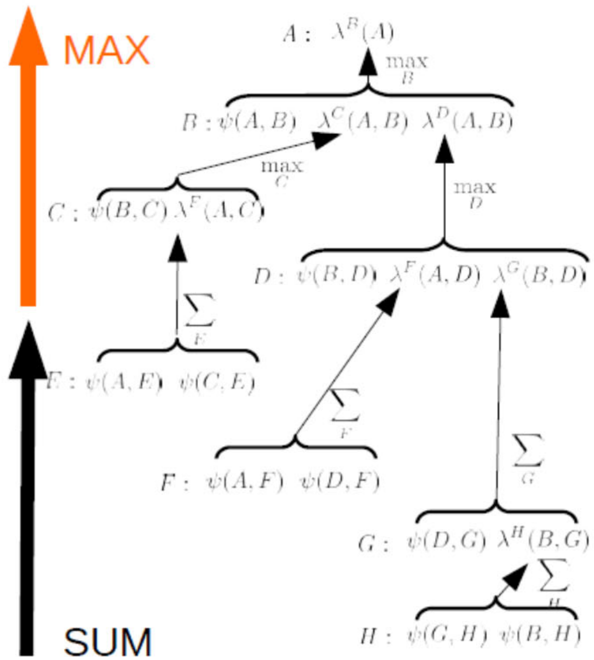
Evaluating a MAP assignment is hard!

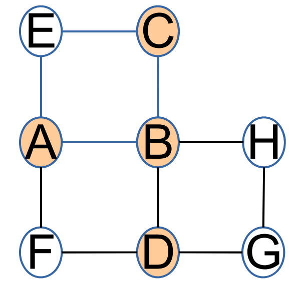
Harder relative to optimization because induced-width is higher and evaluation of a configuration is higher

Harder relative to summation: higher induced-width

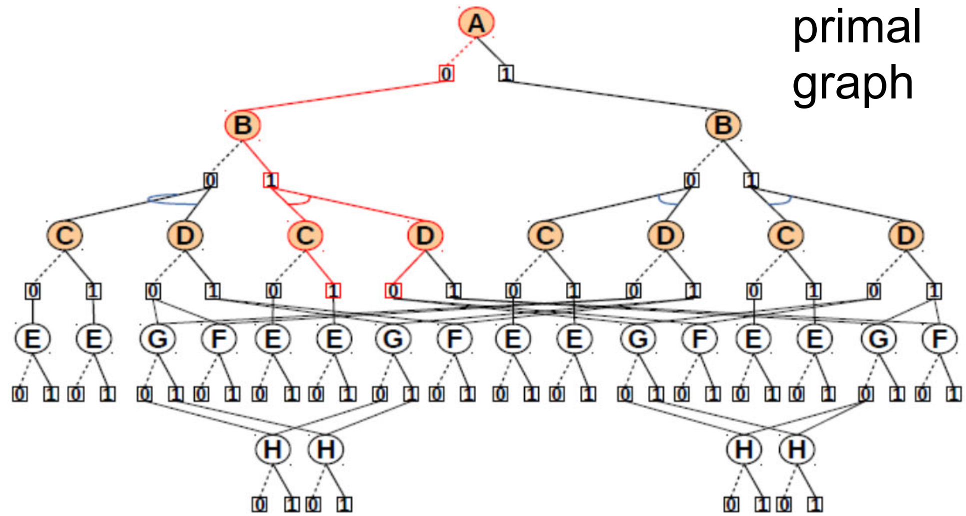
Inference vs Search for MMAP

[Marinescu, Dechter and Ihler, 2014]





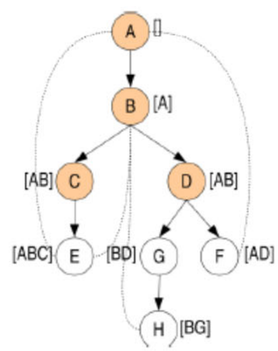
primal graph



O(exp(w*))

- Pseudo tree

[Freuder & Quinn, 1985]

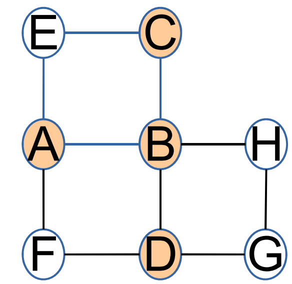


Node value v(n):
- max for MAP vars
- sum for SUM vars

6

AND/OR Search for Marginal MAP

[Marinescu, Dechter and Ihler, 2014]



primal graph

mini-buckets

$f(a,b)\ f(b,c)$ $\quad$ $f(b,d)$

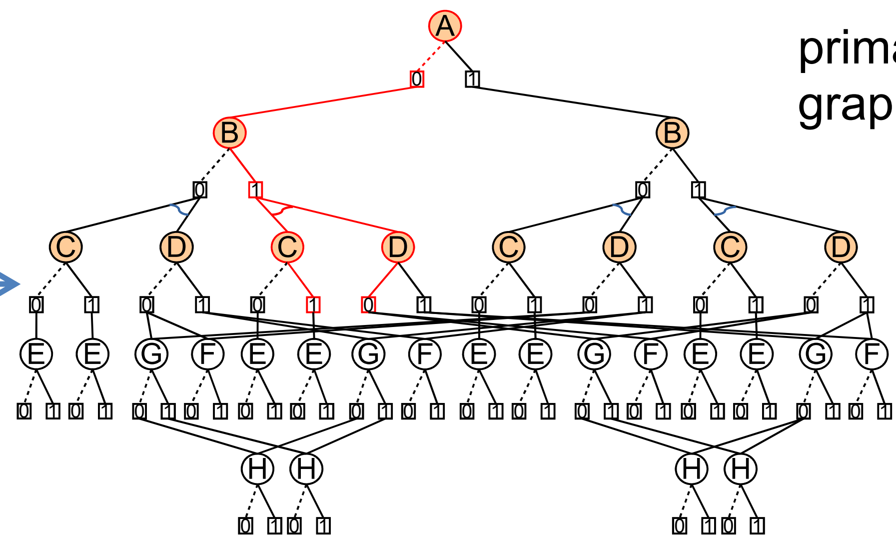
$\lambda_{B\to C}(a,c) f(a,c)\ \ f(c,e)$

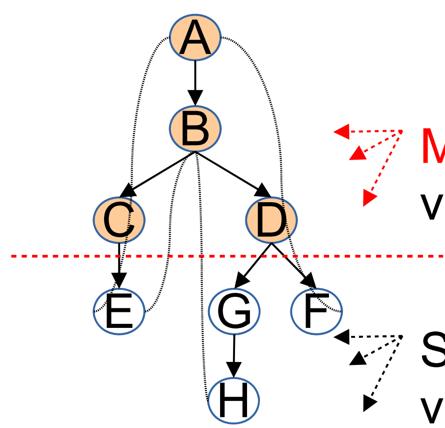
$f(a,d)\,\lambda_{B\to D}(d,e)$

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

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

h



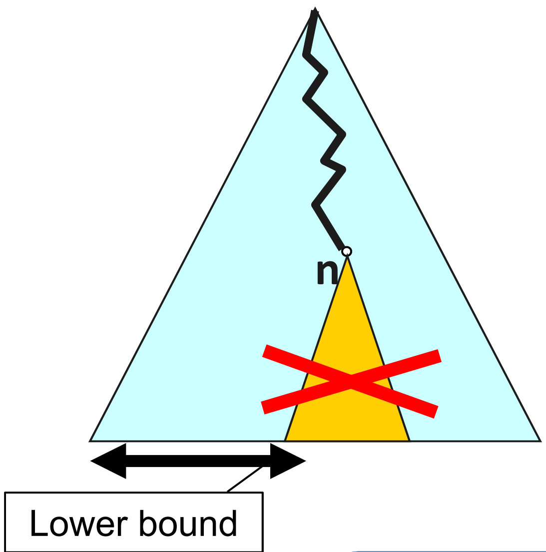


constrained

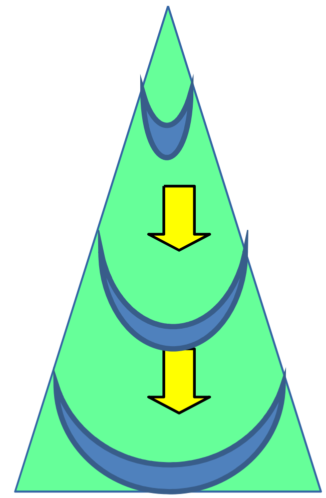
Exact MMAP Solvers: Best or Depth-First Search?

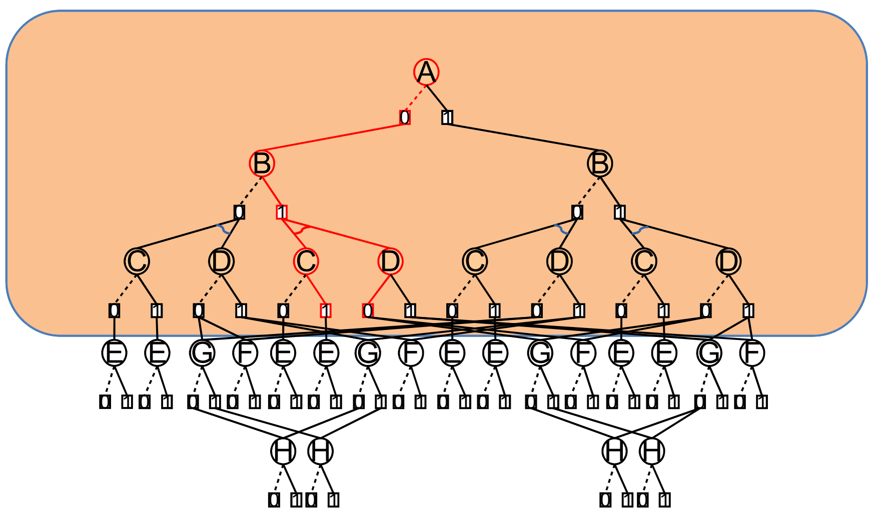
[Marinescu, Dechter, Ihler, AAAI 2014]

Depth-First search



Best-First search



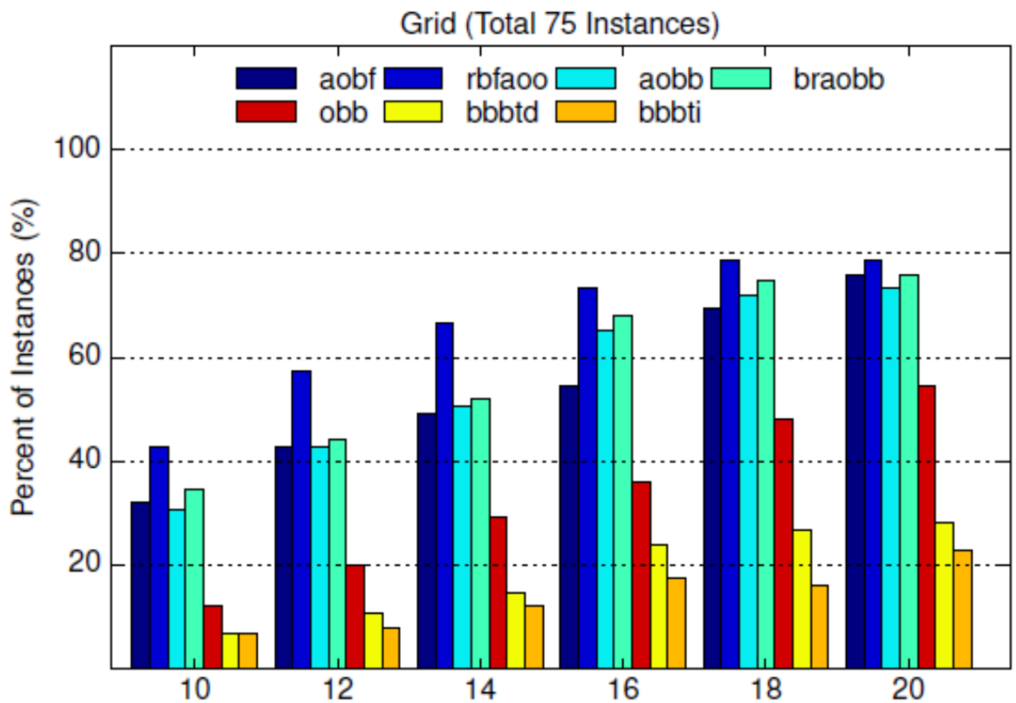


The MAP search space

Best-first search is superior Expanding fewer full MAP Solutions, thus 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

[Marinsecu, Lee, Dechter, Ihler, AAAI-2017, JAIR 2019]

- **Weighted Best-First search:**
  - Weighted Restarting AOBF (WAOBF)
  - Weighted Restarting RBFAOO (WRBFAOO)
  - Weighted Repairing AOBF (WRAOBF)

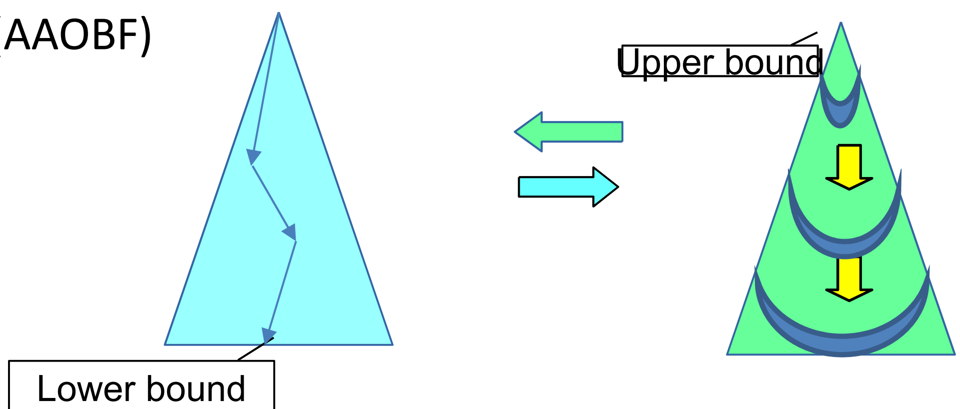
Weighted A* search [Pohl 1970]

- non-admissible heuristic
- Evaluation function:

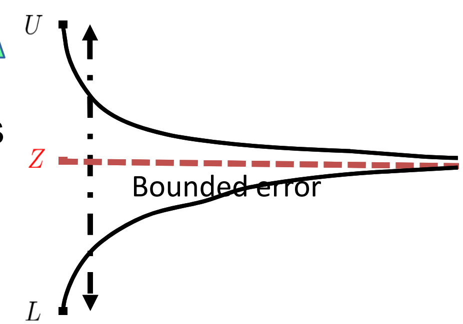
  $f(n)=g(n)+w \cdot h(n)$

- **Guaranteed w-optimal solution, cost $C \leq w \cdot C^*$**

- **Interleaving Best-first and depth-first search**:
  - Look-ahead (LAOBF),
  - alternating (AAOBF)

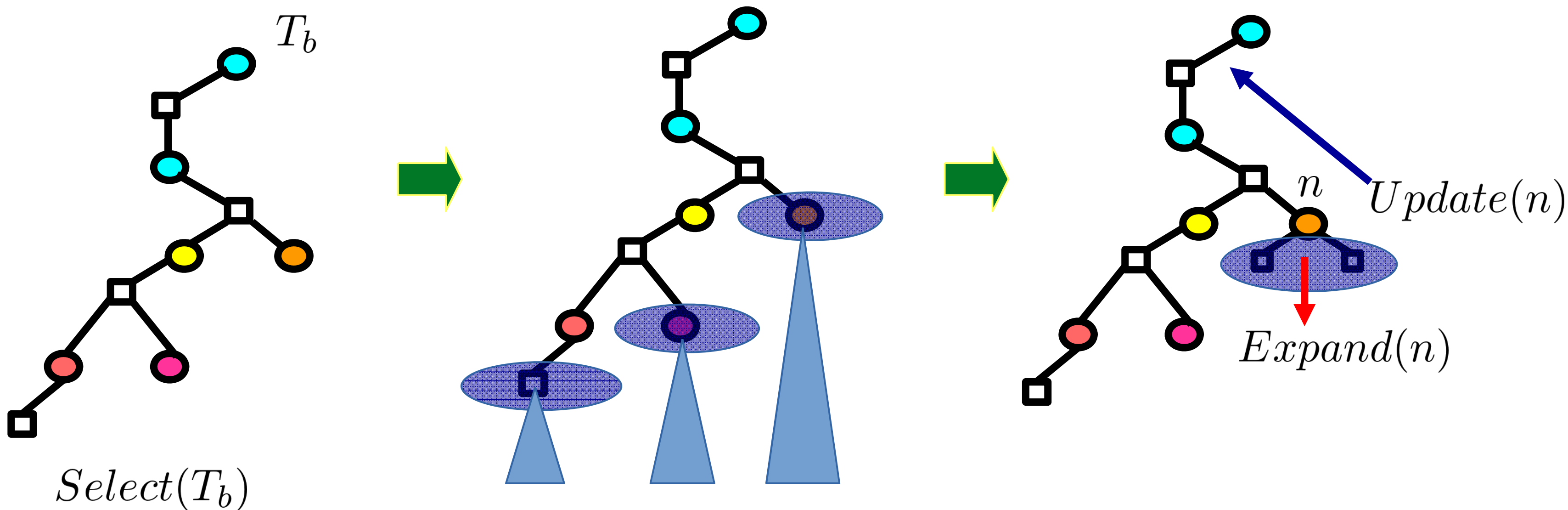


- Better guidance for depth-first dives using improved heuristics
- Memory robust best-first search
- using improved lower bounds



LAOBF (Best-first AND/OR Search with Depth-First lookaheads)

Best-first selection | Depth-first lookahead | Best-first expansion & update

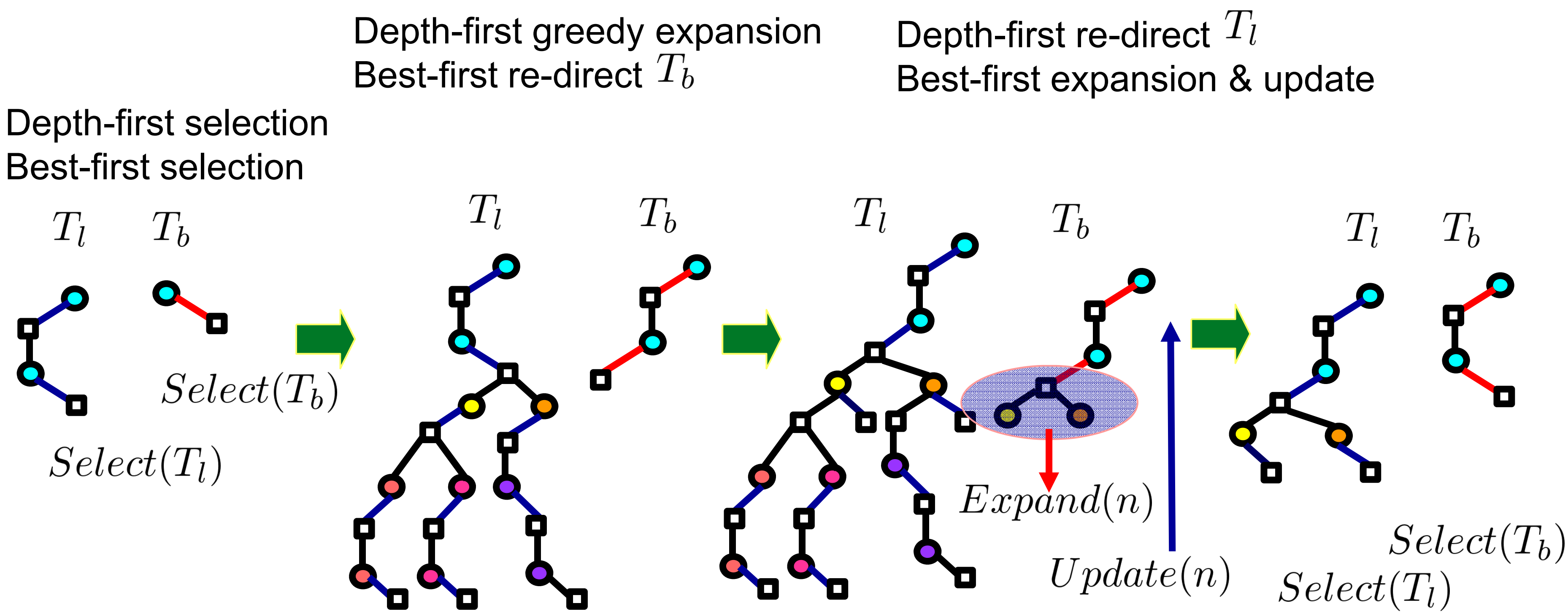


- depth-first dive at the tip of T_b
- compute global lower bound
- cache summation subproblems

- Select a tip node n
- Expand and Update n

Cutoff parameter: perform depth-first dive at every θ number of node expansions.
best partial solution tree: T_b

AAOBF (Alternating Best-First and Depth-First)



AO search for MAP winning UAI Probabilistic Inference Competitions

- **2006**  (aolib)
- **2008**  (aolib)
- **2011**  (daoopt)
- **2014**  (daoopt) |  (daoopt)  (merlin)

MPE/MAP | MMAP

Benchmarks and Evaluation Methods

Benchmark		#. inst	n	k	w_c	h_c	w_u	h_u
grid	easy	15	144 – 1156	2 – 2	16 – 52	50 – 164	15 – 49	48 –198
	hard	60	144 – 1156	2 – 2	25 – 375	42 – 421	–	–
pedigree	easy	10	334 – 1289	4 – 7	35 – 237	51 – 134	15 – 29	60 –160
	hard	40	334 – 1289	4 – 7	35 – 237	63 – 259	–	–
promedas	easy	10	453 – 1849	2 – 2	10 – 122	42 – 174	10 – 106	43 –157
	hard	40	453 – 1849	2 – 2	11 – 490	36 – 507	–	–

Table 1: Benchmark instances. #. inst is the number of instances in each domain. We also distinguish easy and hard instances. The minimum and the maximum values from the set of problems are shown in the following parameters: n is the number of variables, k is the maximum domain size, w_c is the constrained induced width, h_c is the height of the pseudo tree corresponding to the constrained elimination ordering. The unconstrained induced width, w_u and pseudo tree height, h_u are also shown to highlight the difficulty of hard Marginal MAP problem instances.

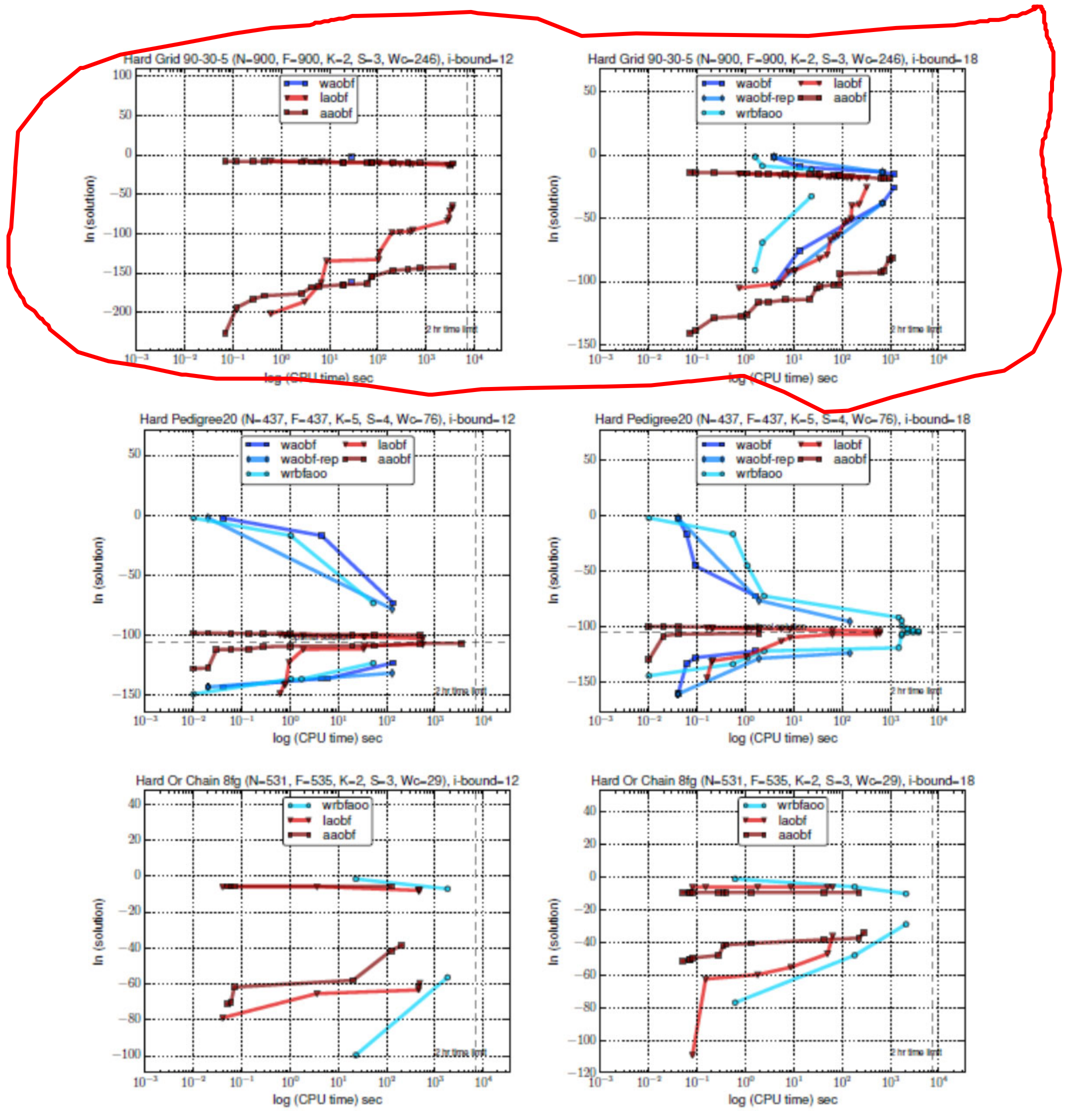
Anytime Bounds of Marginal MAP

- 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)**.

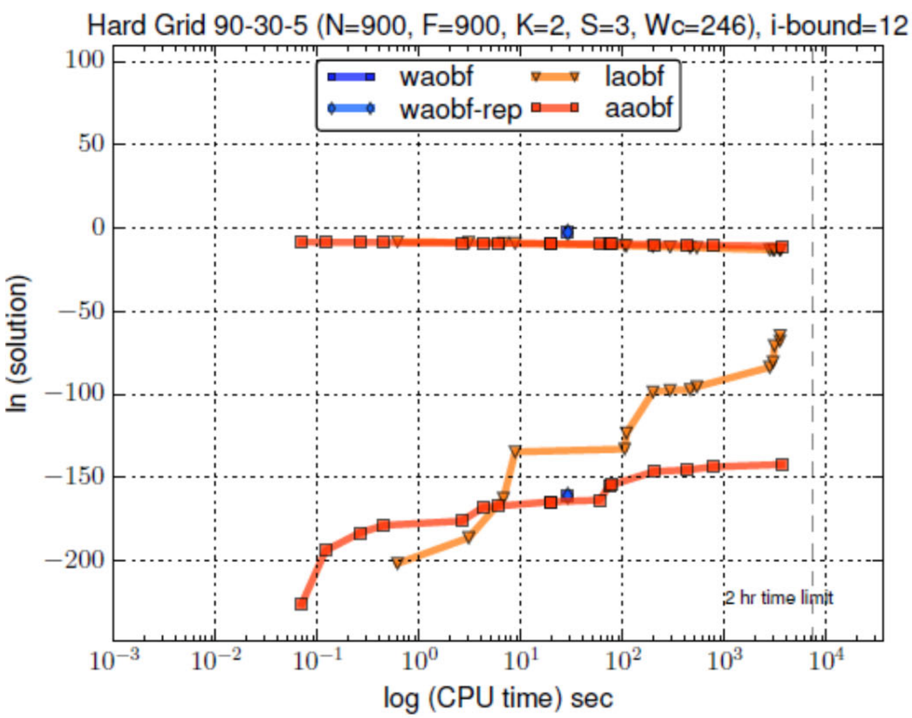
- The horizontal axis is the CPU time in log scale and the vertical axis is the value of marginal MAP in log scale.

- Benchmarks: Pedigrees, promedas, grids, planning. A fraction of variables selected as MAP (10% hard instances).

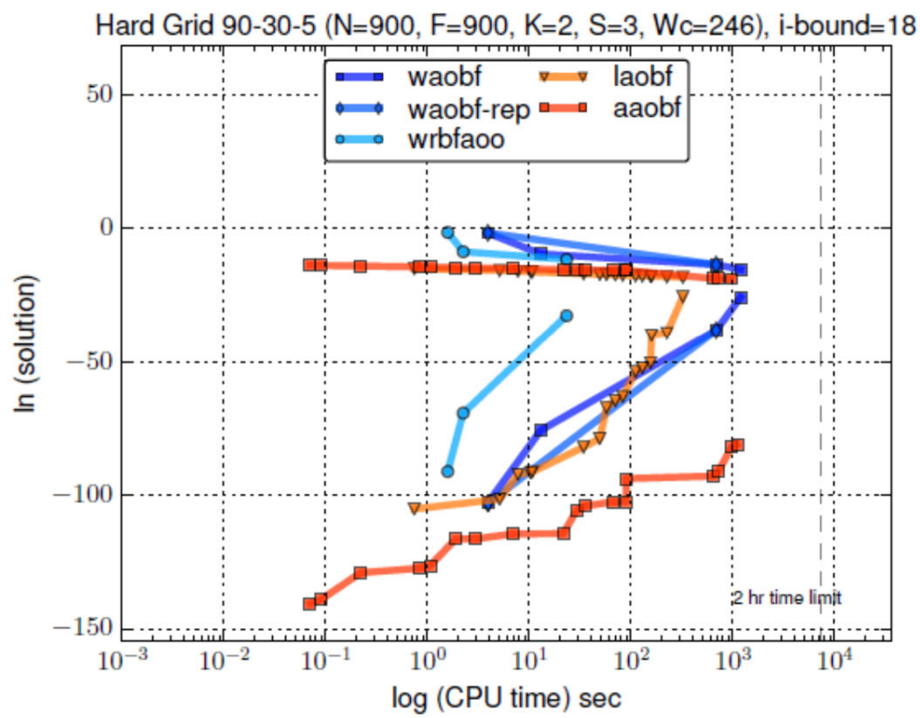


Anytime Bounds of Marginal MAP

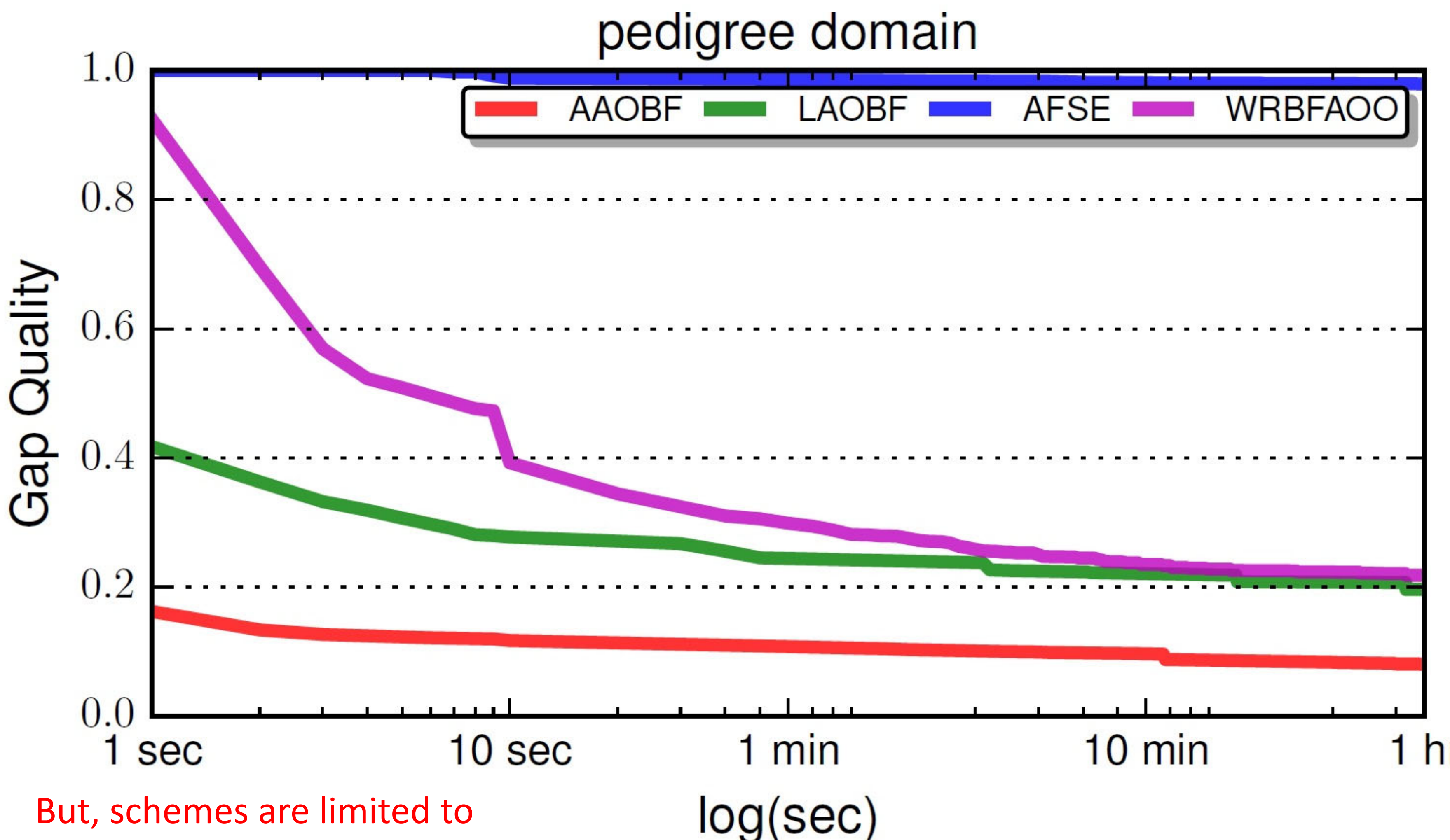
i-bound = 12



i-bound = 18



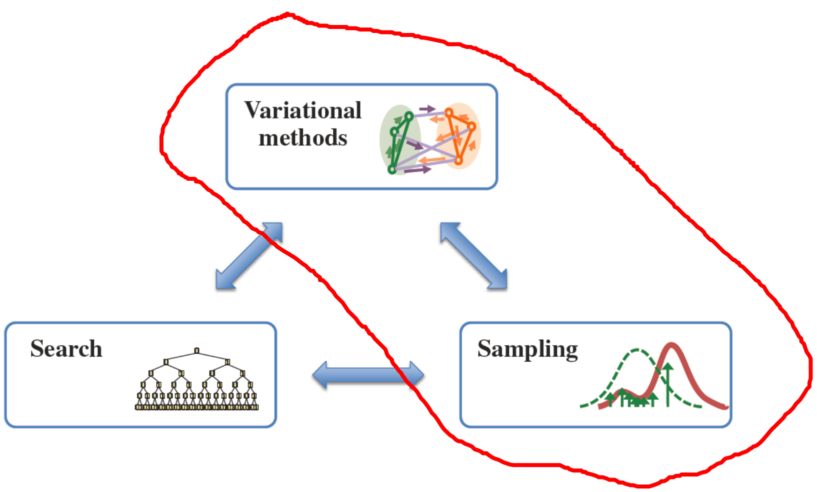
Average Gap Quality

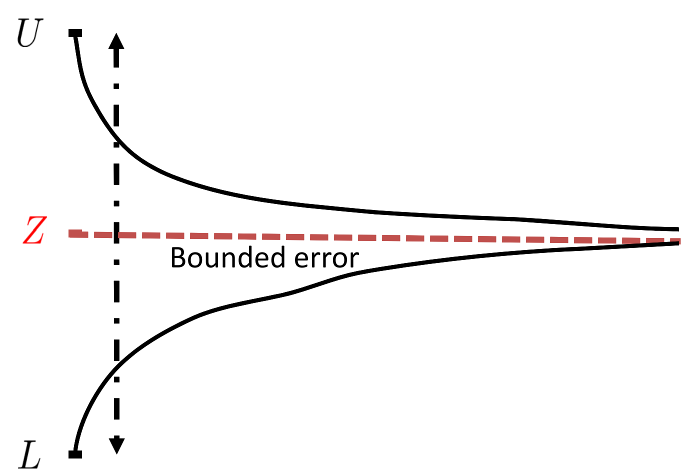


But, schemes are limited to tractable conditioned-summation

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





Importance Sampling

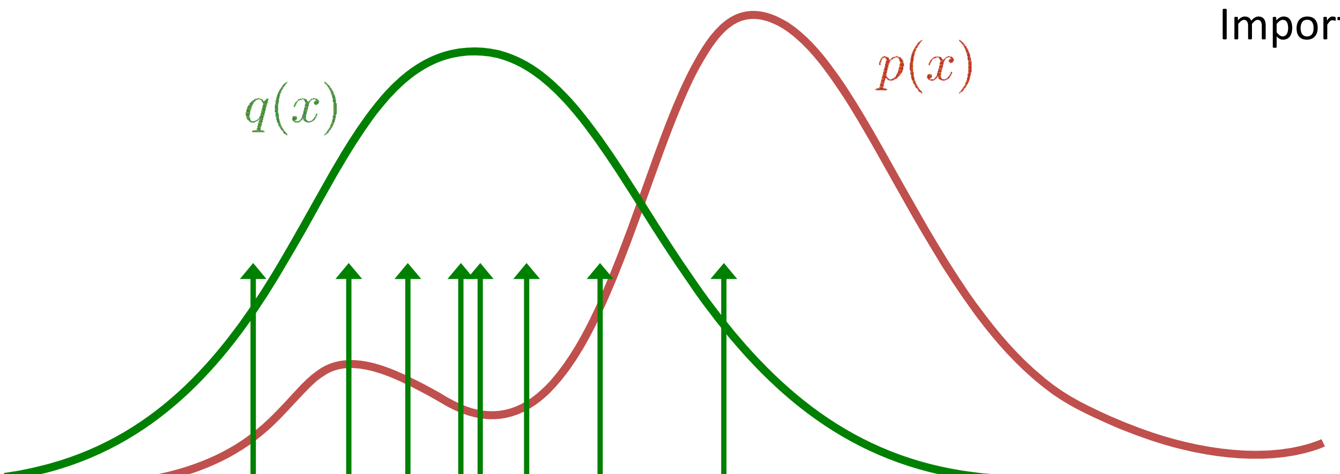
- Basic empirical estimate of probability:

$$\mathbb{E}[u(x)] = \int p(x)u(x) \;\approx\; \hat{u} = \frac{1}{m}\sum_i u(\tilde{x}^{(i)}) \qquad \tilde{x}^{(i)} \sim p(x)$$

- Importance sampling:

$$\int p(x)u(x) = \int q(x)\frac{p(x)}{q(x)}u(x) \approx \frac{1}{m}\sum_i \frac{p(\tilde{x}^{(i)})}{q(\tilde{x}^{(i)})}u(\tilde{x}^{(i)}) \qquad \tilde{x}^{(i)} \sim q(x)$$

Importance weights



Choosing a Proposal- WMB-IS

[Liu, Fisher, Ihler 2015]

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

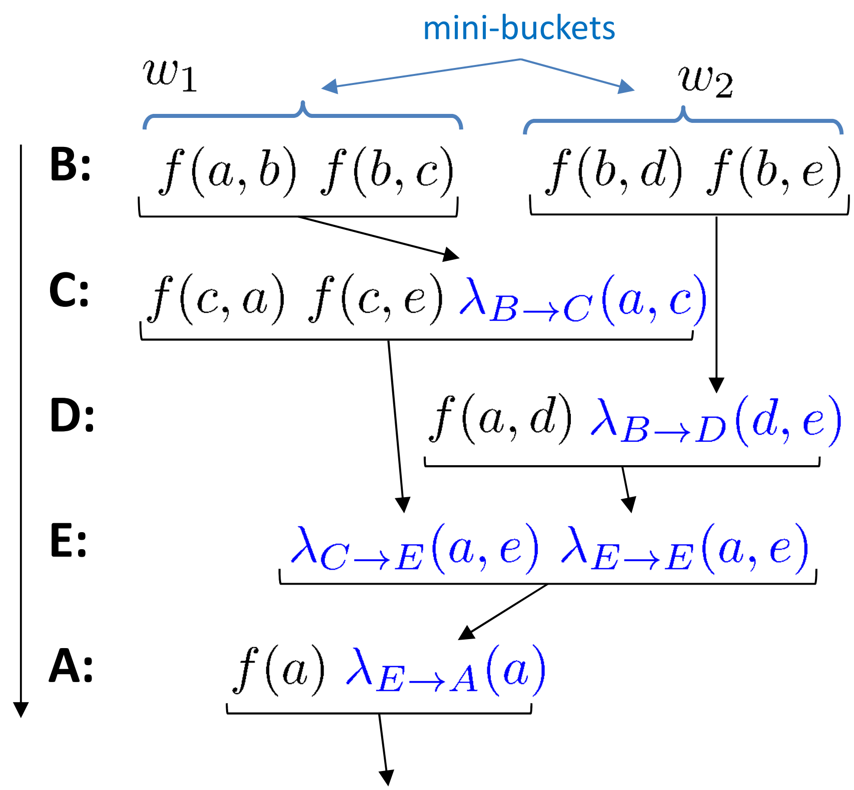
$$\tilde{\mathbf{b}} \sim w_1\, q_1(b|\tilde{a},\tilde{c}) \;+\; w_2\, q_2(b|\tilde{d},\tilde{e})$$

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

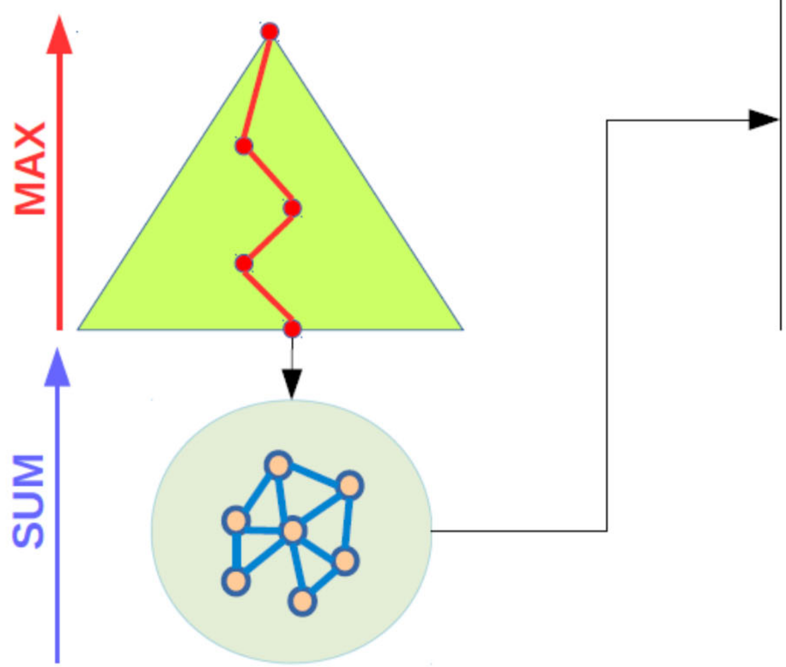
Key insight: provides bounded importance weights!

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

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:

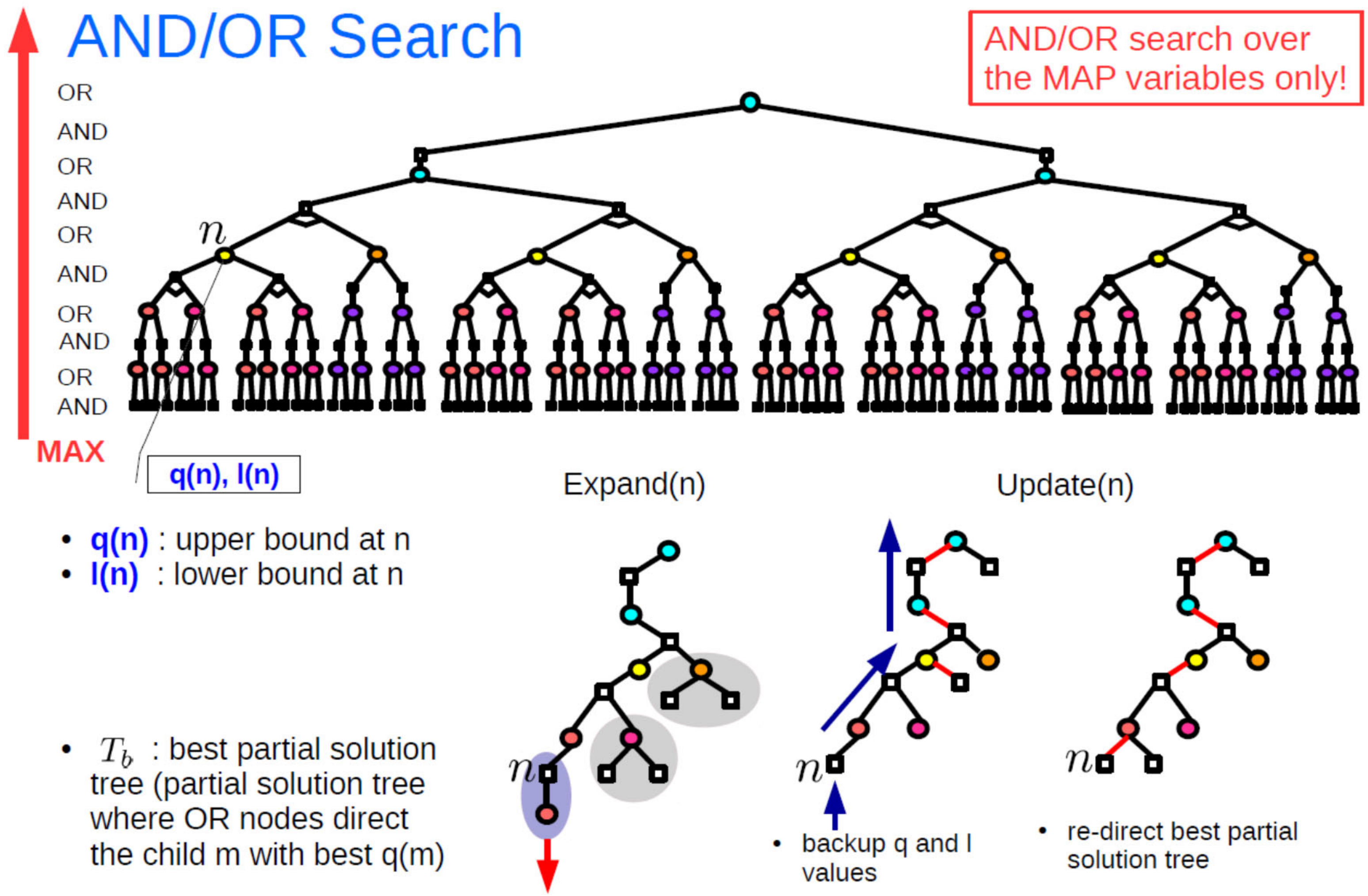
- n - number of samples
- δ - confidence value
- Z_{wmb} - result of WMB
- $\hat{Z}$ - Importance Sampling estimate

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

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

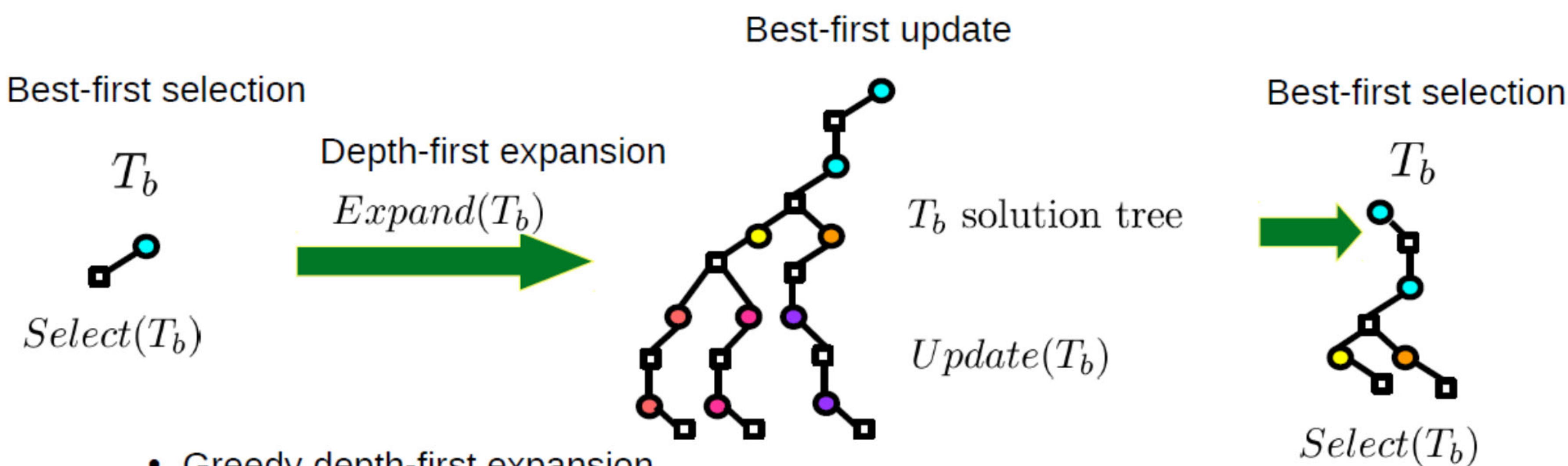
Upper bound U, decreasing as $1/n$

Stochastic Anytime Search for MMAP



ANYLDFS

AnyLDFS (anytime learning depth-first search)



- Greedy depth-first expansion to a solution tree

T_b solution tree

- **Lower bound SUM evaluation based on the WMB-IS scheme**

Search is conduced over the MAP variables only!

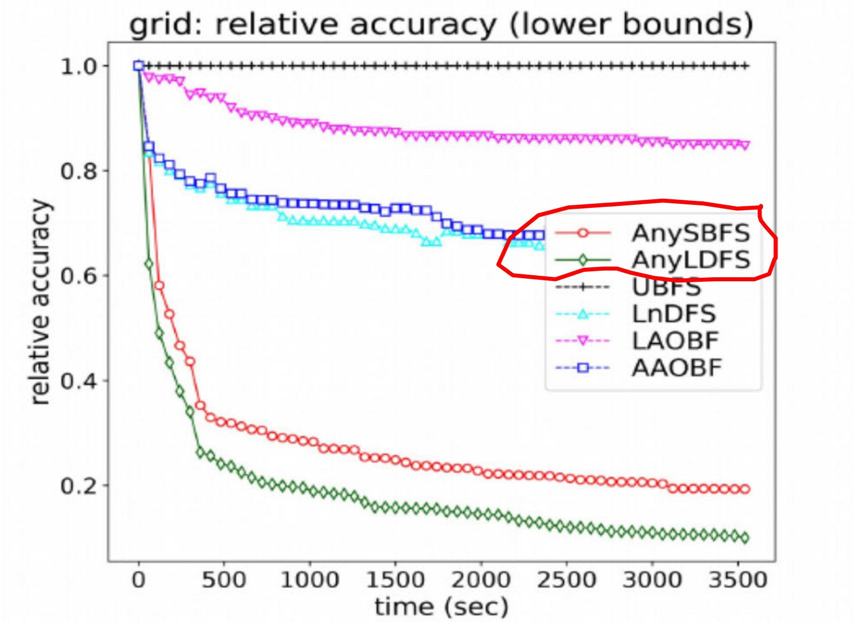
Stochastic Anytime Search for MMAP (Grids)

[Marinescu, Ihler, Dechter IJCAI-2018, Lou, Dechter, Ihler AAAI-2018]

ANYSBFS: Anytime Stochastic Best-First Search

ANYLDFS: Anytime Learning Depth-First Search

$$acc_{lb} = \frac{|l_t - l^*|}{l^*}$$

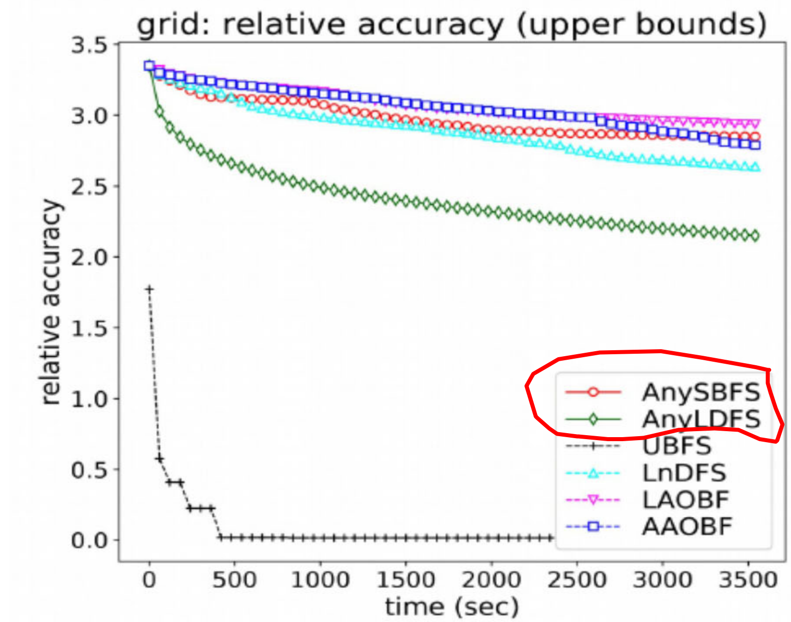


l_t – lower bound at time t

l^* – tightest lower bound found

Average over 150 instances

$$acc_{ub} = \frac{|u_t - u^*|}{u^*}$$



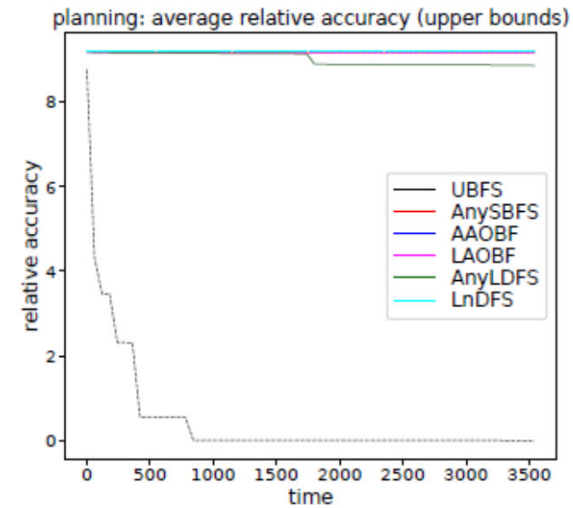
u_t – upper bound at time t

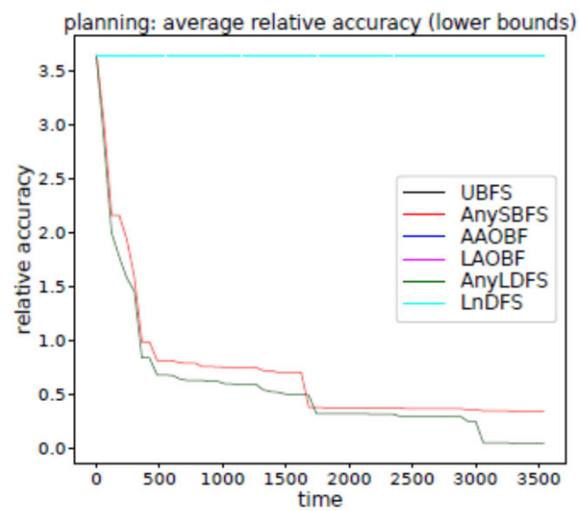
u^* – tightest upper bound found

Average over 150 instances

(Lower plots are better)

Stochastic Anytime Search for MMAP (Planning)





Software

- **daoopt**
  - https://github.com/lotten/daoopt

  (distributed and standalone AOBB solver)
- **merlin**
  - https://developer.ibm.com/open/merlin

  (standalone WMB, AOBB, AOBF, RBFAOO solvers)

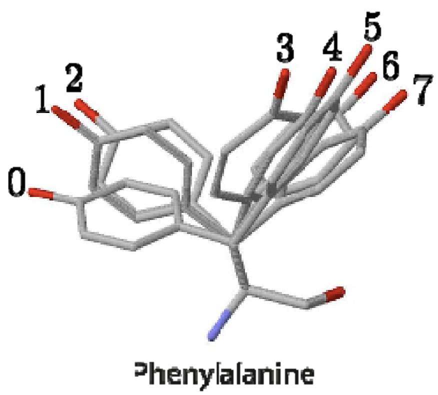
  open source, BSD license

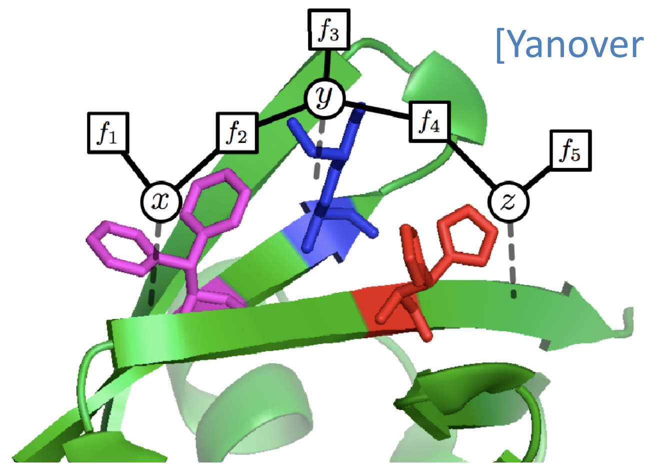
pyGM : Python Toolbox for Graphical Models by Alexander Iher.

Future work

- Examples & Tasks
  - Maximization (MAP): compute the most probable configuration

$$Obj = \underset{R_1 \ldots R_N}{\operatorname{argmax}} \max_{C_1,\ldots,C_N} \prod_{E_{ij} \in E_{pw}} e^{-\frac{E_{ij}(C_i, C_j)}{\mathscr{R}T}}$$





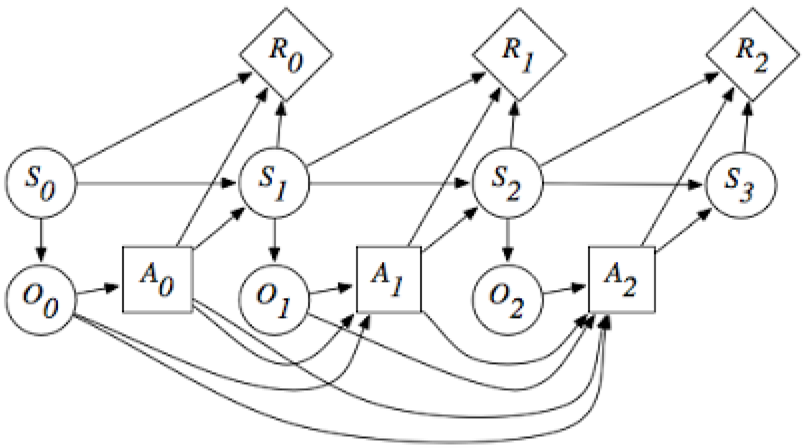
[Yanover & Weiss 2002]

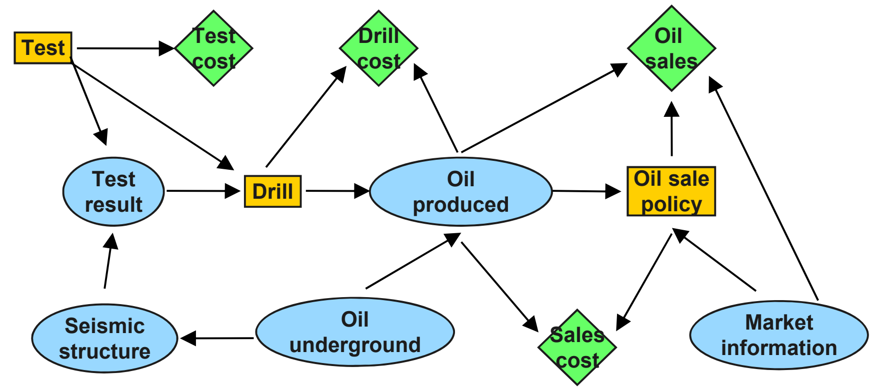
  - Mixed Max-sum (Marginal Map): compute the most likely marginal

$$X^* = \underset{R_1 \ldots R_N}{\operatorname{argmax}} \sum_{C_1,\ldots,C_N} \prod_{E_{ij} \in E_{pw}} e^{-\frac{E_{ij}(C_i, C_j)}{\mathscr{R}T}}$$

Planning as graphical models

Find a sequence of decision that maximize the expected utility/rewards





Expected utility (fixed policy)

$$\mathrm{EU} = \sum_x \prod f_\alpha(x_\alpha) \sum_a u_a(x_a)$$

Maximum expected utility

$$\mathrm{MEU} = \max_\delta \mathbb{E}(u(x)|\delta)$$

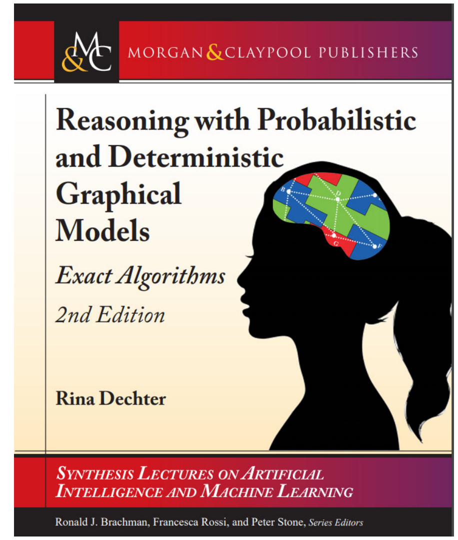
$$= \max_\delta \sum_x u(x) \prod_{i \in C} p_i(x_i | x_{\mathrm{pa}(i)}) \prod_{i \in D} p_i^\delta(x_i | x_{\mathrm{pa}(i)})$$

Influence diagrams & optimal decision-making

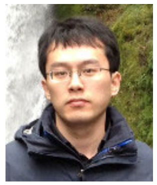
(the "oil wildcatter" problem)

Thank You !

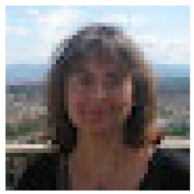
For publication see:
http://www.ics.uci.edu/~dechter/publications.html





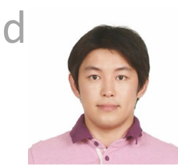
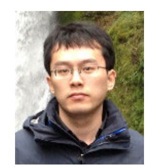







Alex Ihler

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


UNIVERSITY of CALIFORNIA IRVINE