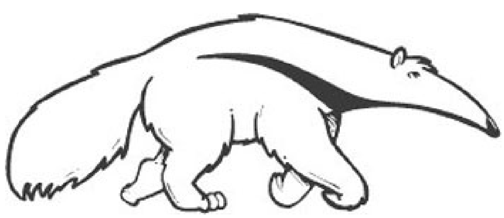
Probabilistic Reasoning Meets Heuristic Search

Rina Dechter

Collaborators:
Radu marinescu,
Alex Ihler,
Junkyu Lee



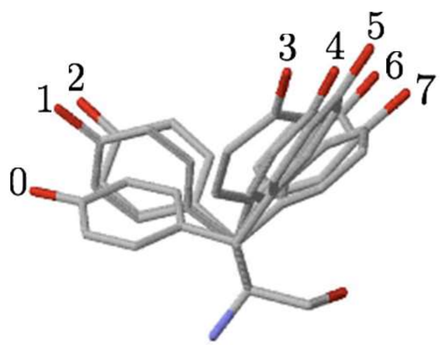
BREN:ICS
INFORMATION AND COMPUTER SCIENCES

UNIVERSITY of CALIFORNIA IRVINE

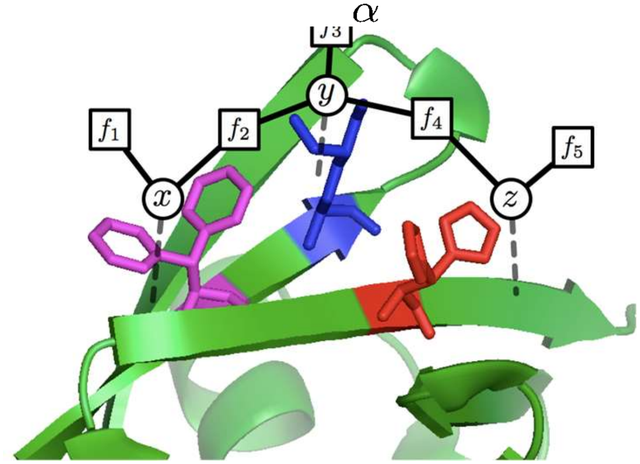
Probabilistic Graphical models

- Describe structure in large problems
  - Large complex system $F(X)$
  - Made of "smaller", "local" interactions $f_\alpha(x_\alpha)$
  - Complexity emerges through interdependence
- Examples & Tasks
  - Maximization (MAP): compute the most probable configuration

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



Phenylalanine



[Yanover & Weiss 2002]

Probabilistic Graphical models

- Describe structure in large problems
  - Large complex system $F(X)$
  - Made of "smaller", "local" interactions $f_\alpha(x_\alpha)$
  - Complexity emerges through interdependence
- Examples & Tasks
  - Summation & marginalization

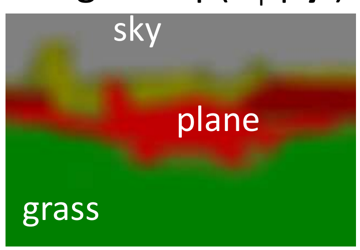
"partition function"

$$p(x_i) = \frac{1}{Z} \sum_{\mathbf{x} \setminus x_i} \prod_\alpha f_\alpha(\mathbf{x}_\alpha) \quad \text{and} \quad Z = \sum_{\mathbf{x}} \prod_\alpha f_\alpha(\mathbf{x}_\alpha)$$

Observation **y**

$\Rightarrow$

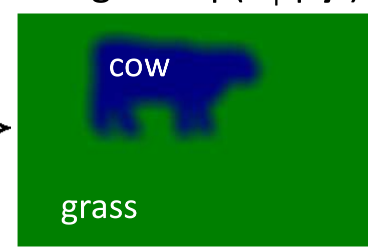
Marginals p(x_i | **y**)



Observation **y**

$\Rightarrow$

Marginals p(x_i | **y**)



e.g., [Plath et al. 2009]

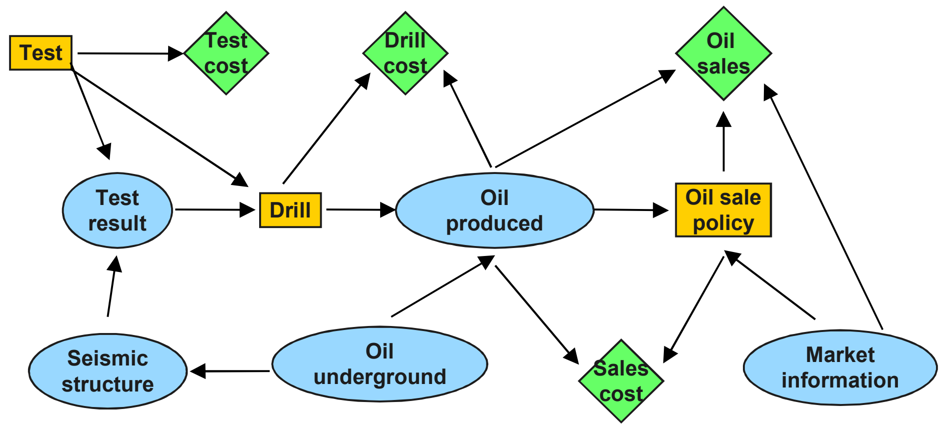
Graphical models

- Describe structure in large problems
  - Large complex system $F(X)$
  - Made of "smaller", "local" interactions $f_\alpha(x_\alpha)$
  - Complexity emerges through interdependence
- Examples & Tasks
  - Mixed inference (marginal MAP, MEU, …)

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

Influence diagrams &
optimal decision-making

(the "oil wildcatter" problem)



e.g., [Raiffa 1968; Shachter 1986]

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 or "factors"

Operators:

combination operator

(sum, product, join, …)

elimination operator

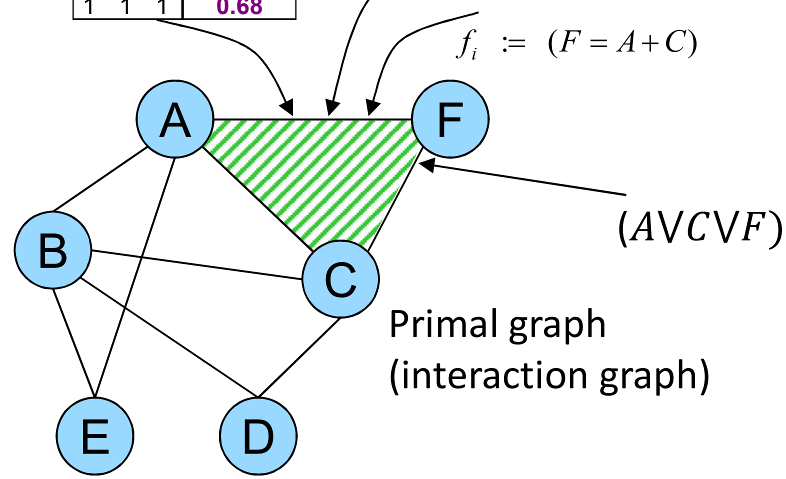
(projection, sum, max, min, ...)

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

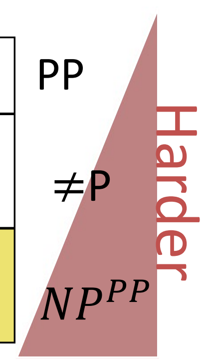
Relation

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



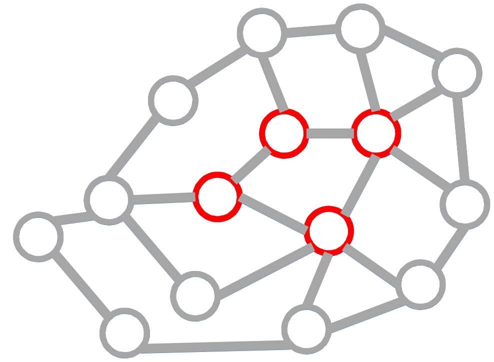
Types of queries:

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



- All these tasks are NP-hard
  - exploit problem structure
  - identify special cases
  - approximate

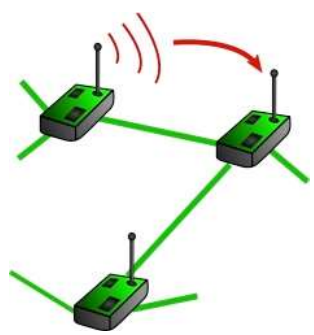
Why Marginal MAP?



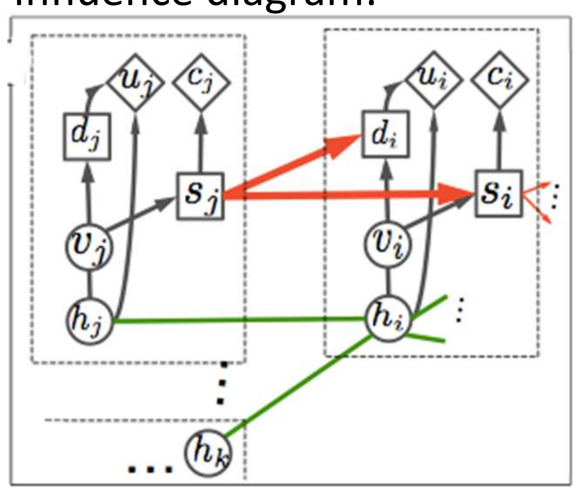
- Often, Marginal MAP is the "right" task:
  - We have a model describing a large system
  - We care about predicting the state of some part
- Example: decision making
  - Sum over random variables
  - Max over decision variables (specify action policies)

- Complexity: NP^{pp} complete
- Not necessarily easy on trees

Sensor network

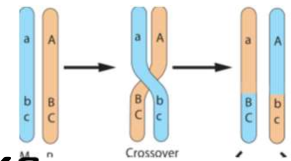


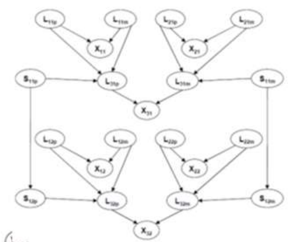
Influence diagram:

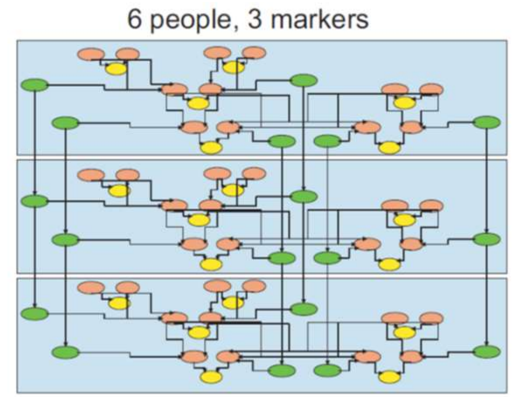


Example for MMAP Applications

- Haplotype in Family pedigrees







- Coding networks

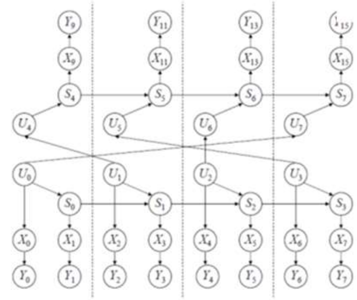
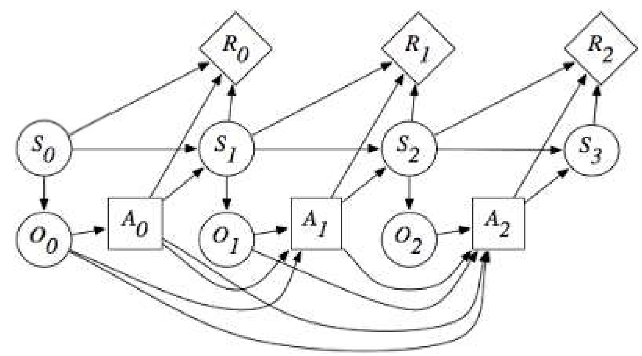
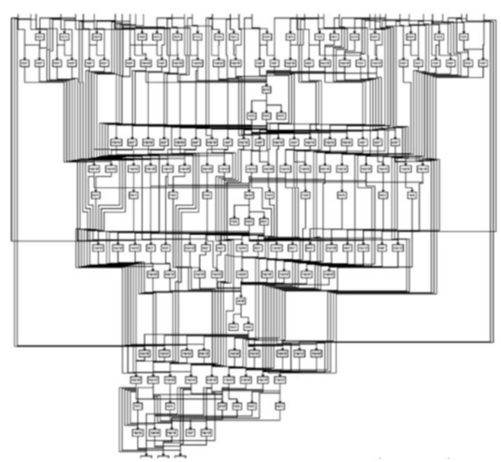


Figure 5.24: A Bayesian network for a turbo code.

- Probabilistic planning



- Diagnosis



Marginal map

- 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 $\mathbf{F} = \{f_1, \ldots, f_r\}$

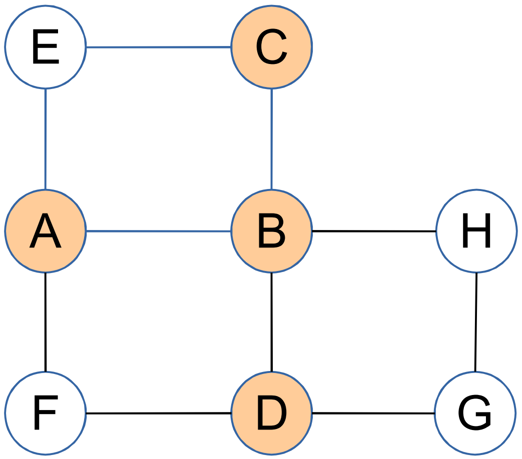
$$P(\mathbf{X}) = \frac{1}{Z} \prod_j f_j$$

- Marginal MAP task:

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

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

primal graph

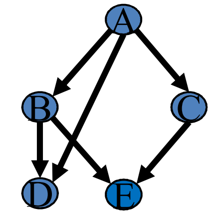


$$\mathbf{X}_M = \{A, B, C, D\}$$

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

Why is it harder? intuitively

Finding Marginals by Bucket elimination



Algorithm *BE-bel* (Dechter 1996)

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

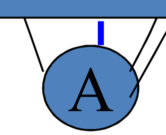
$\sum_b \prod$ ← Elimination operator

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)

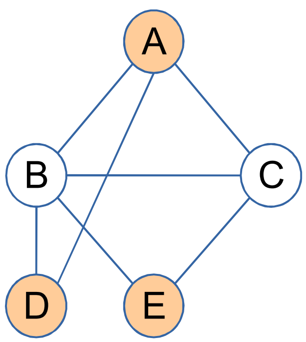


P(e=0)

P(a|e=0)

Why is MMAP harder?

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



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

$\mathbf{X}_S = \{B, C\}$

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

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

constrained elimination order

SUM

B: $f(A,B)\, f(B,C)\, f(B,D) f(B,E)$

Σ_B

C: $\lambda^B(A,C,D,E) f(A,C)\, f(C,E)$

Σ_C

MAX

D: $\lambda^C(A,D,E)\, f(A,D)$

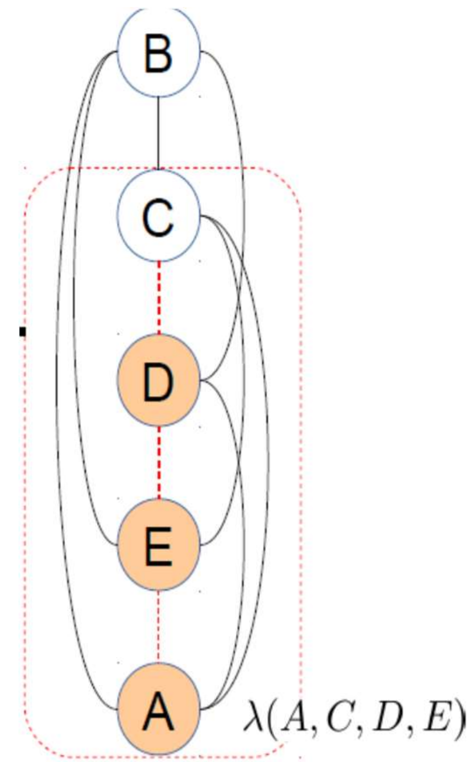
max_D

E: $\lambda^D(A,E)$

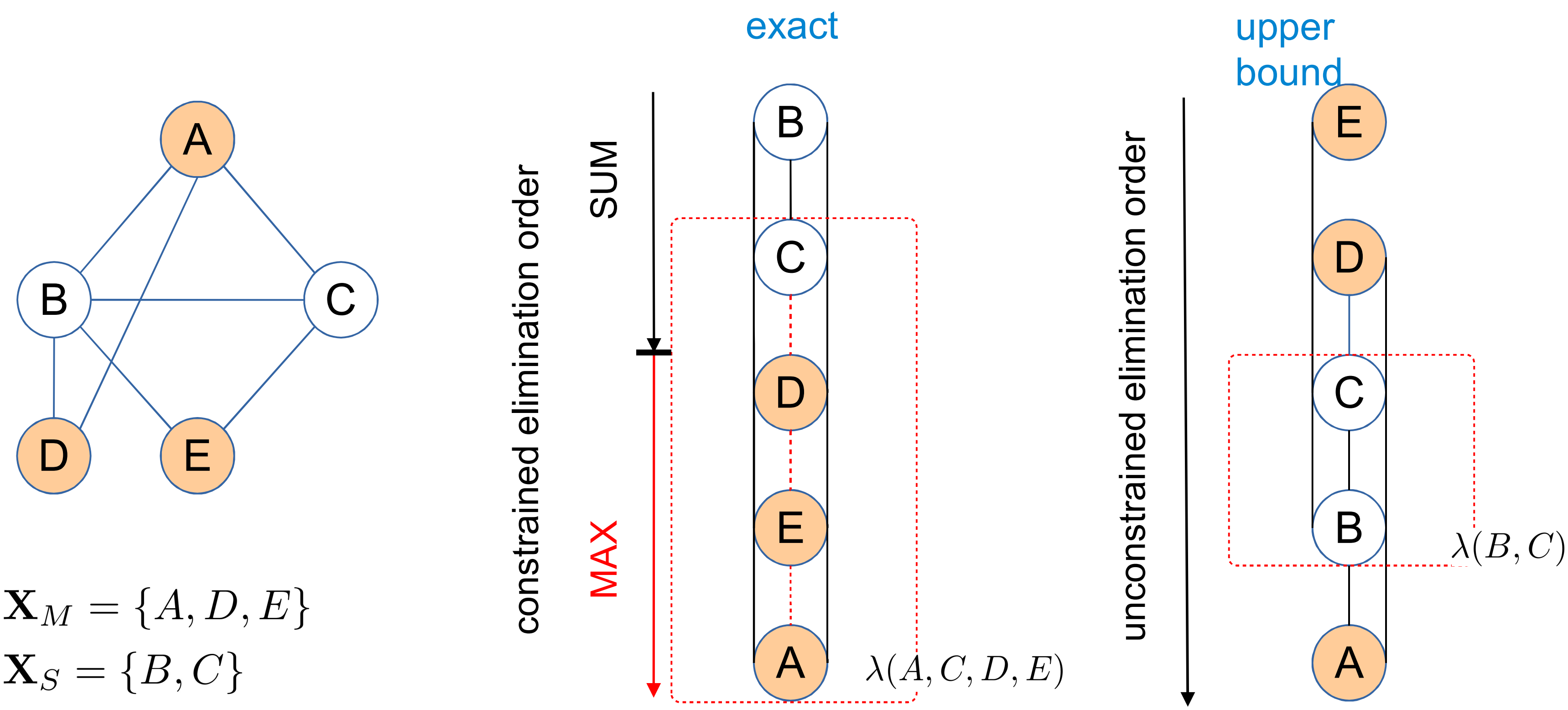
max_E

A: $\lambda^E(A)$

MAP* is the marginal MAP value



Why is MMAP harder?



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

$\mathbf{X}_S = \{B, C\}$

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

In practice, constrained induced is much larger!

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

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

Complexity of Bucket Elimination

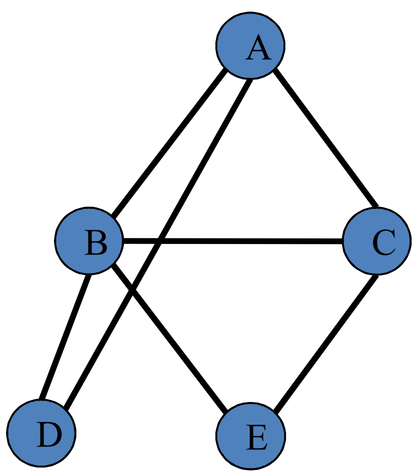
Bucket Elimination is **time** and **space**

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

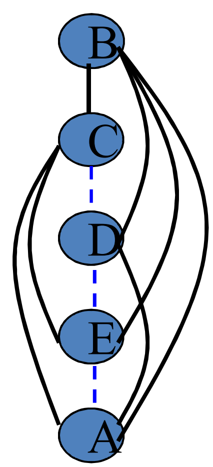
$w^*(d)$ – the induced width of graph along ordering d

r = number of functions

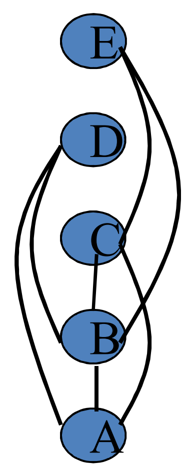
The effect of the ordering:



"Moral" graph



$w^*(d_1) = 4$



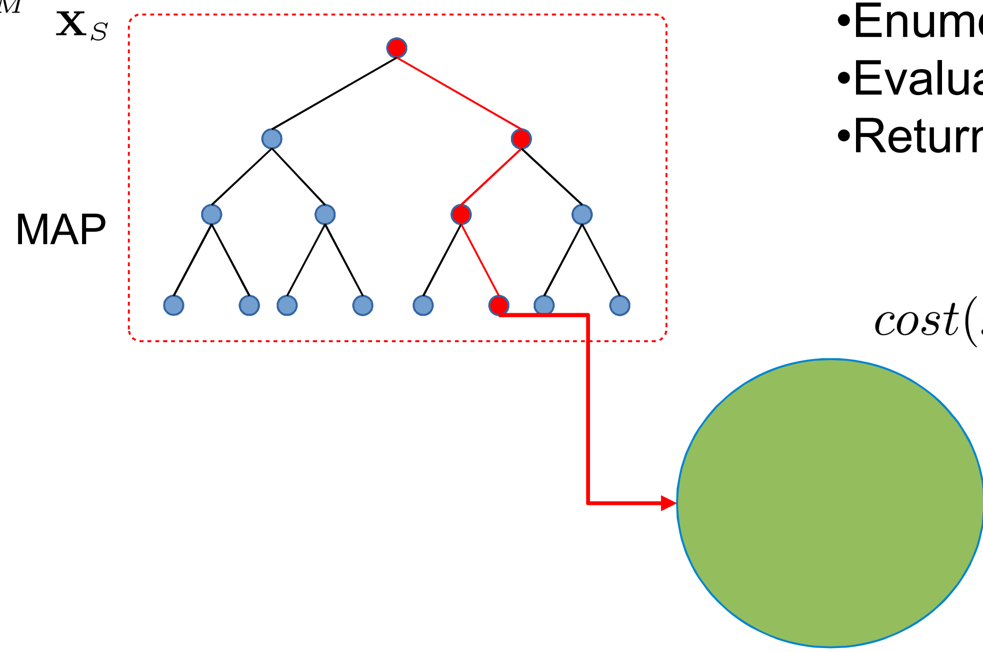
$w^*(d_2) = 2$

Finding the smallest induced width is hard!

Why is MMAP harder?

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

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

$$\#P - complete$$

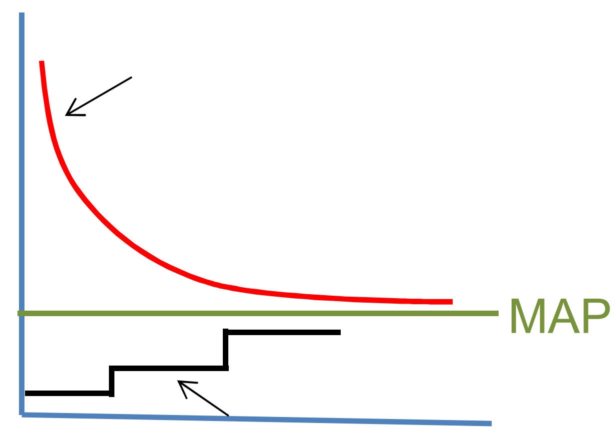
Evaluating a MAP assignment is hard!

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

Harder relative to summation: higher induced-width

Outline

- Graphical models, marginal map and planning
- Heuristic search meets probabilistic reasoning:
  - AND/OR search spaces
  - Decomposition bounds
- MMAP AND/OR search with WMB heuristics
  - Exact search
  - Anytime search
- Marginal Map for planning
- Challenges and future plans



AND/OR Search Spaces for Graphical Models

Potential search spaces

$$F^* = \min_x \sum_\alpha f_\alpha(x_\alpha)$$

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

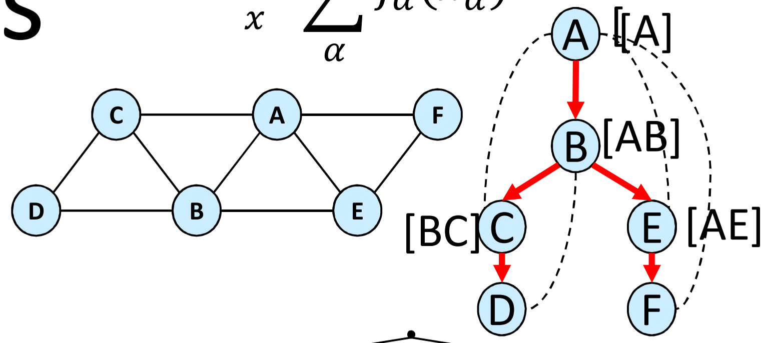
B	C	f_5
0	0	0
0	1	1
1	0	2
1	1	4

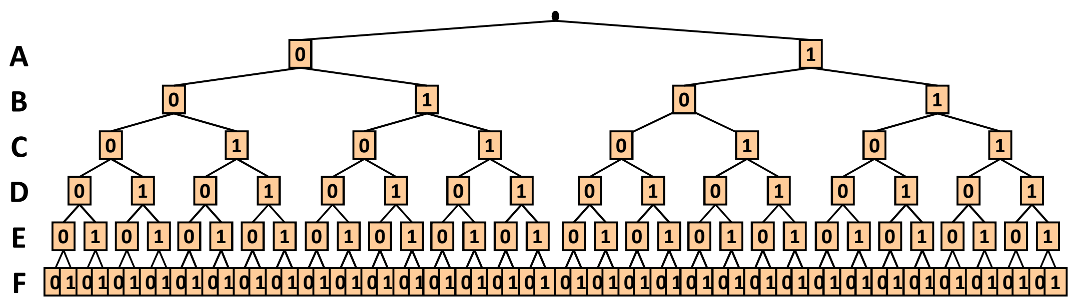
B	D	f_6
0	0	4
0	1	2
1	0	1
1	1	0

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

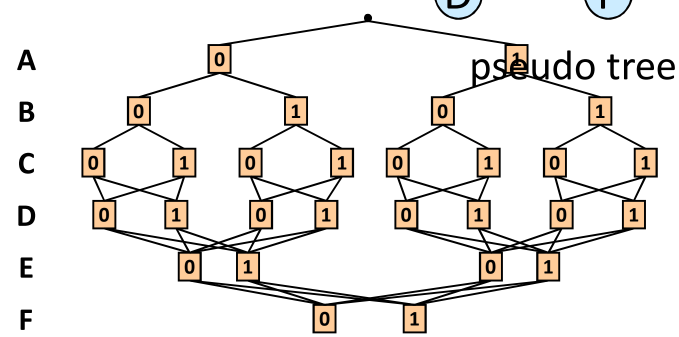
E	F	f_9
0	0	1
0	1	0
1	0	0
1	1	2





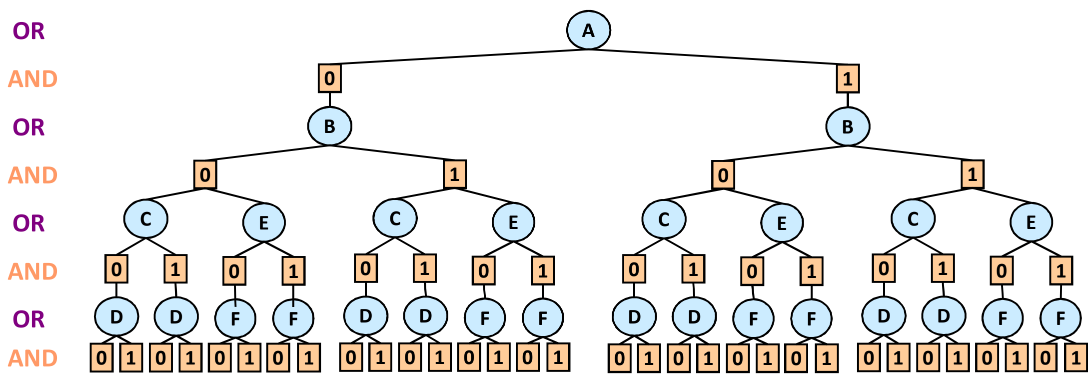
Full OR search tree

126 nodes



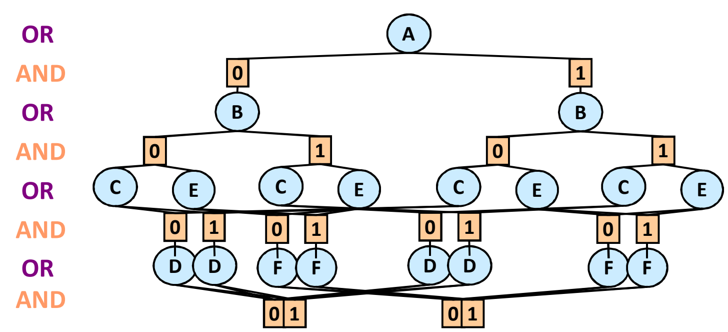
Context minimal OR search graph

28 nodes



Full AND/OR search tree

54 AND nodes



Context minimal AND/OR search graph

18 AND nodes

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

Potential search spaces

$$F^* = \min_x \sum_\alpha f_\alpha(x_\alpha)$$

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

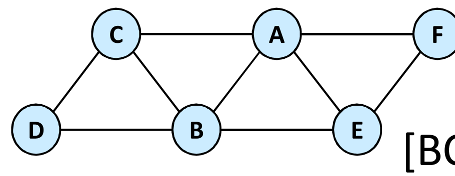
B	C	f_5
0	0	0
0	1	1
1	0	2
1	1	4

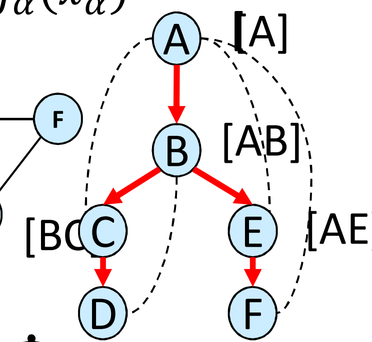
B	D	f_6
0	0	4
0	1	2
1	0	1
1	1	0

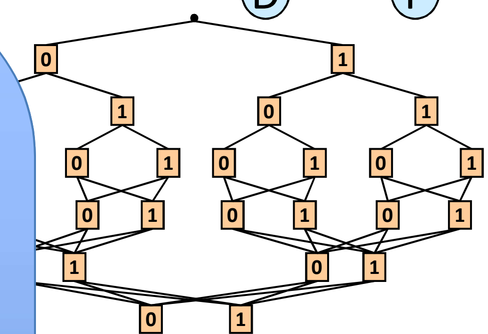
B	E	f_7
0	0	3
0	1	2
1	0	1
1	1	0

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

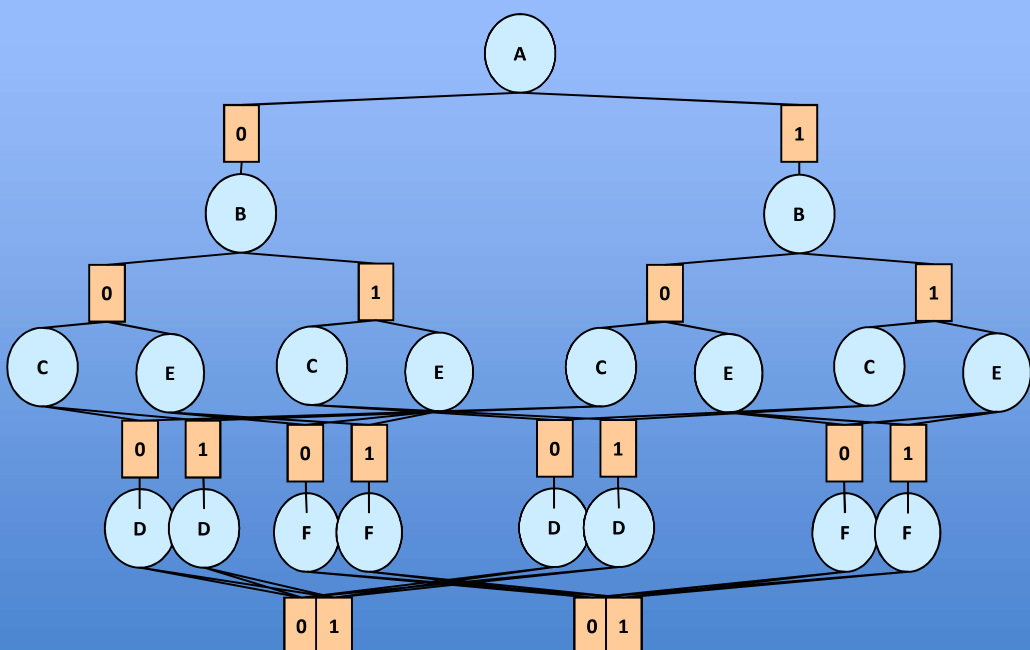






ext minimal OR search graph

28 nodes



Context minimal AND/OR search graph

18 AND nodes

A

s best computed

Over the c-minimal AO search space

Potential search spaces

$$F^* = \min_x \sum_\alpha f_\alpha(x_\alpha)$$

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

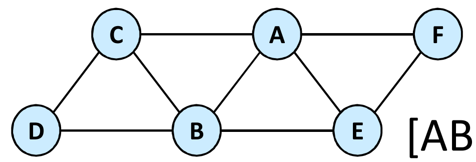
B	C	f_5
0	0	0
0	1	1
1	0	2
1	1	4

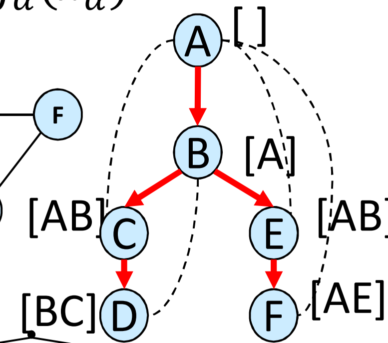
B	D	f_6
0	0	4
0	1	2
1	0	1
1	1	0

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

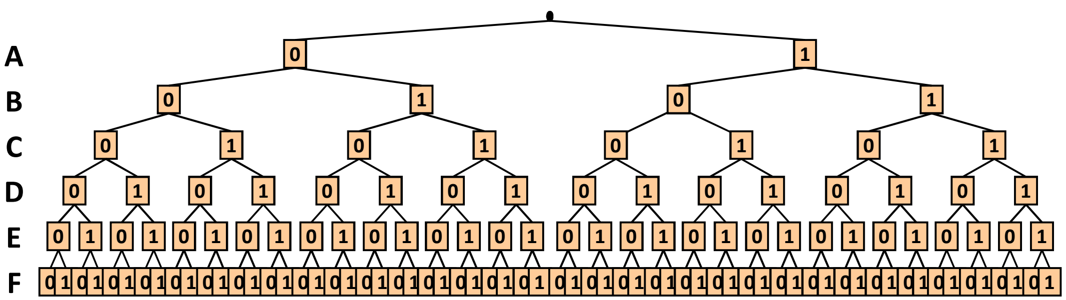
C	D	f_8
0	0	1
0	1	4
1	0	0
1	1	0

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



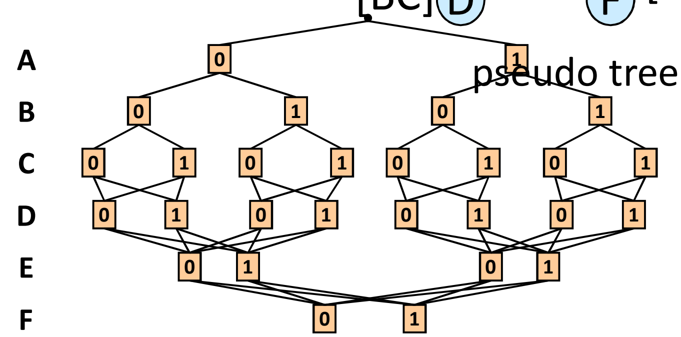


pseudo tree



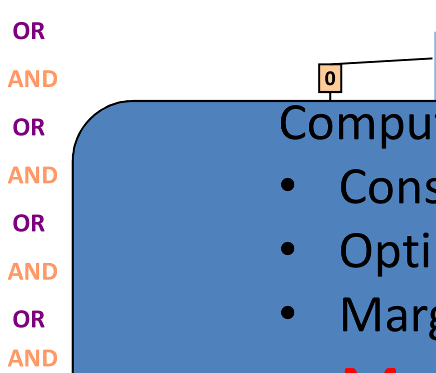
Full OR search tree

126 nodes

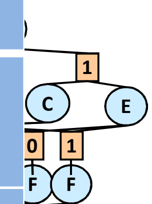


Context minimal OR search graph

28 nodes



	OR tree	AND/OR	OR graph	AND/OR graph
[illegible]	[illegible]	[illegible]	$O(n\ k^{pw^*})$	$O(n\ k^{w^*})$
[illegible]	[illegible]	[illegible]	$O(n\ k^{pw^*})$	$O(n\ k^{w^*})$



Computes any query:

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

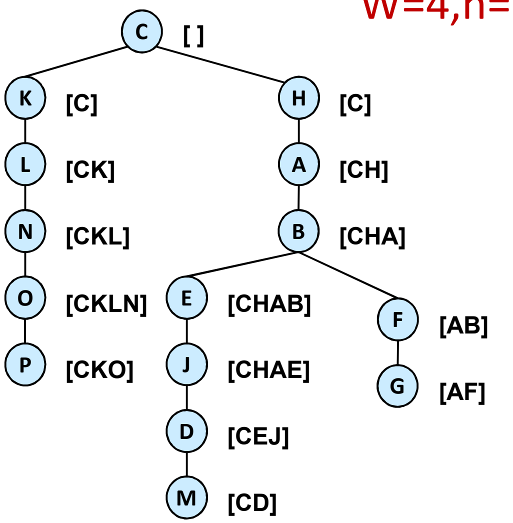
18 AND nodes

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

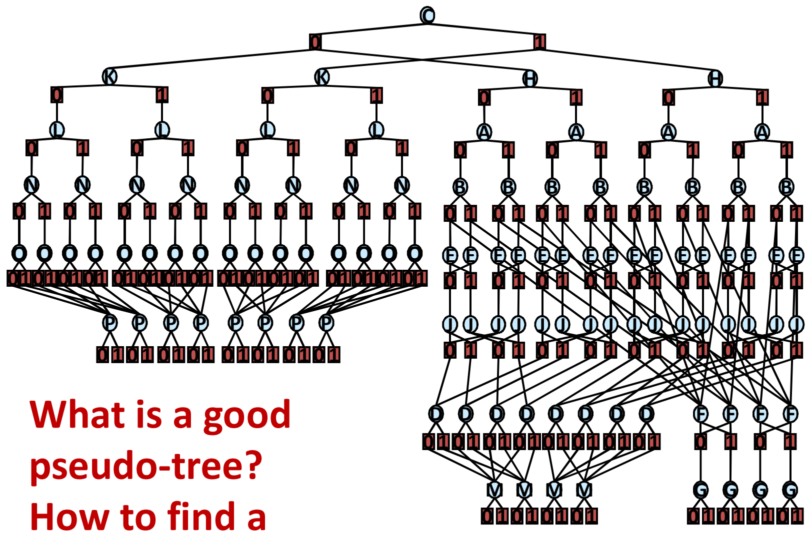
The Impact of the Pseudo-Tree

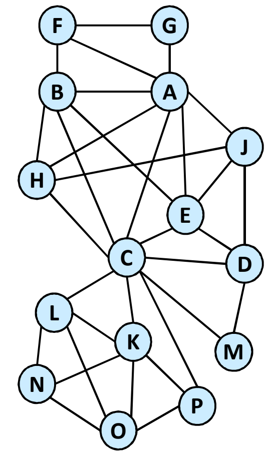
N=15

W=4,h=8



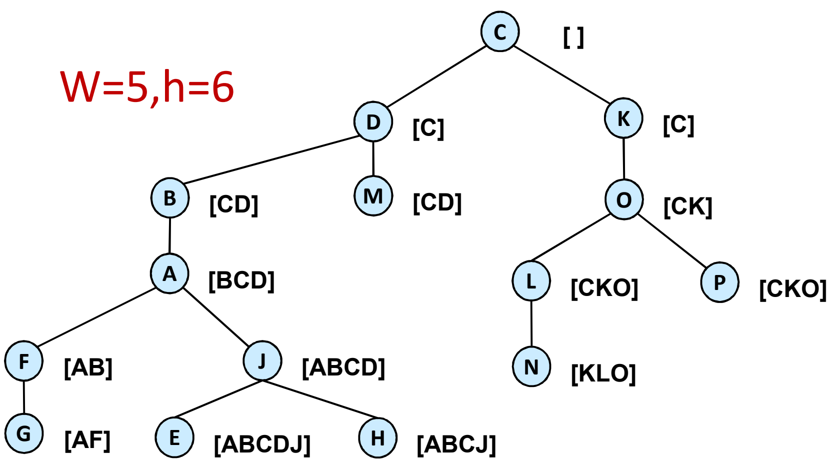
(C K H A B E J L N O D P M F G)

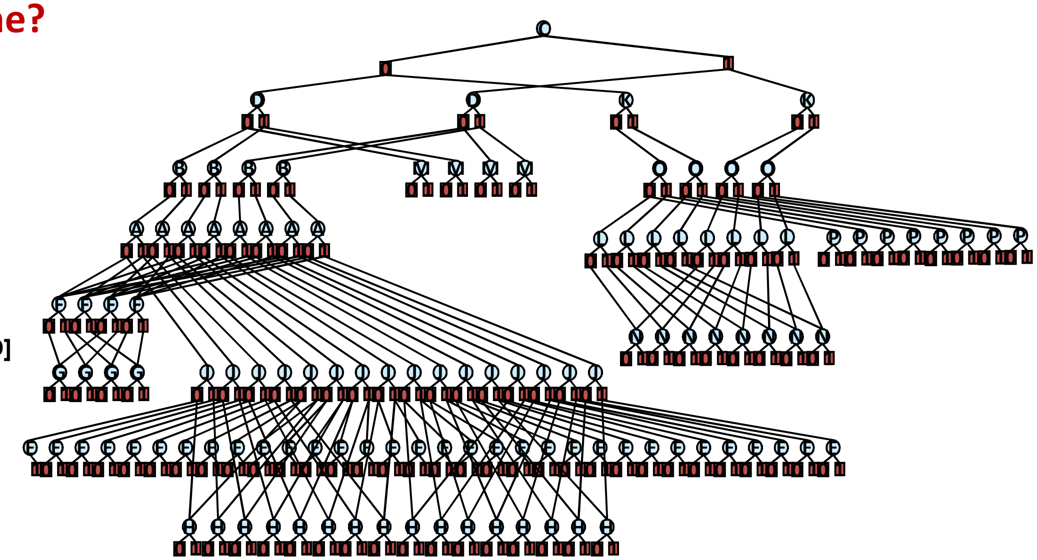




What is a good pseudo-tree? How to find a good one?

W=5,h=6





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

In more detail:
Arc-weights and 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

$P(B \mid A)$

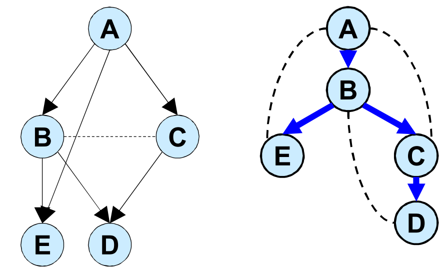
A	B=0	B=1
0	.4	.6
1	.1	.9

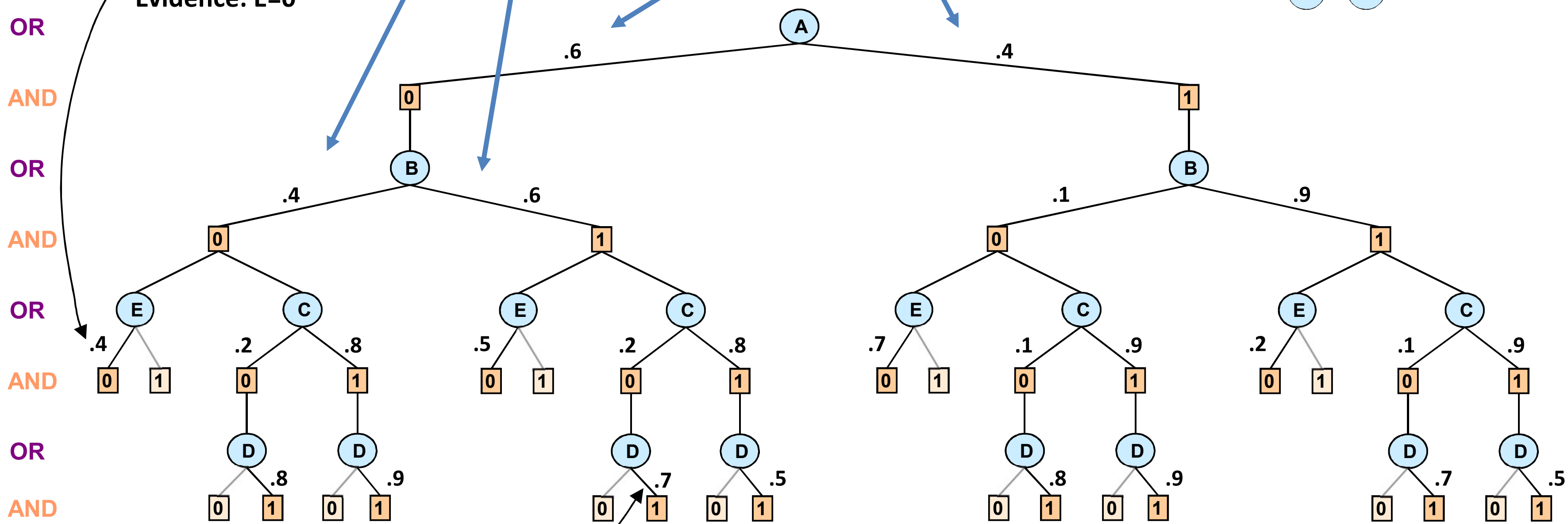
$P(C \mid A)$

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

In more detail:
Arc-weights and 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

$P(B \mid A)$

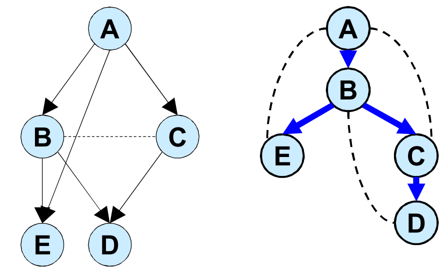
A	B=0	B=1
0	.4	.6
1	.1	.9

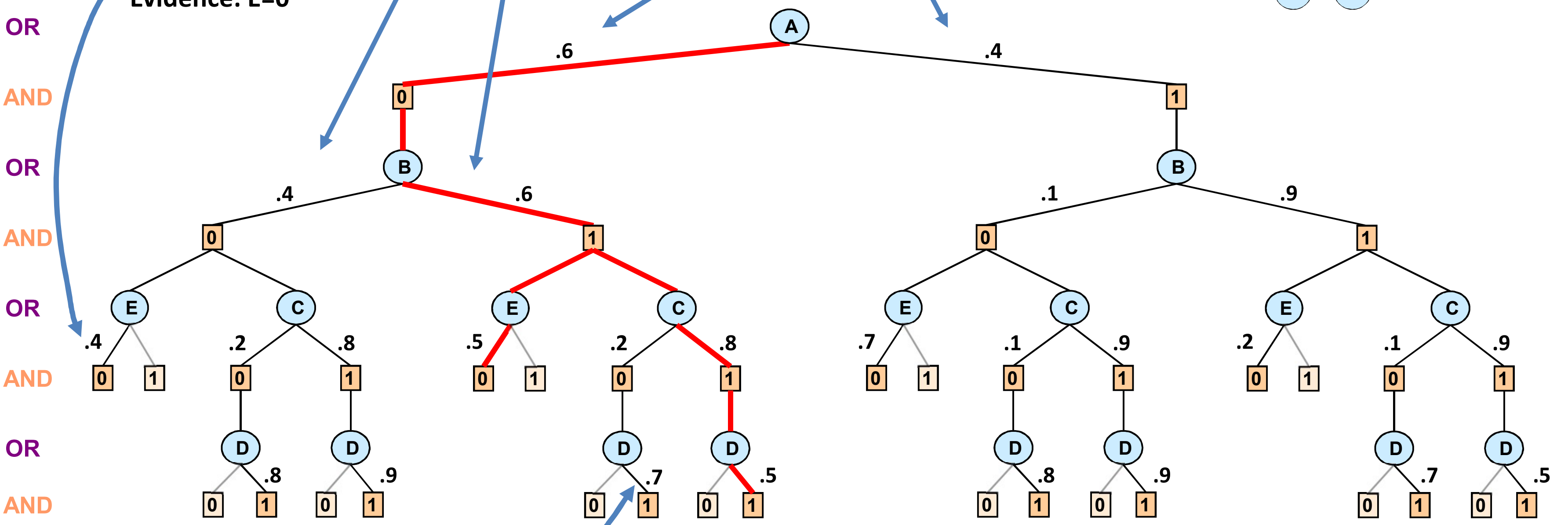
$P(C \mid A)$

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

In more detail:
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

$P(B \mid A)$

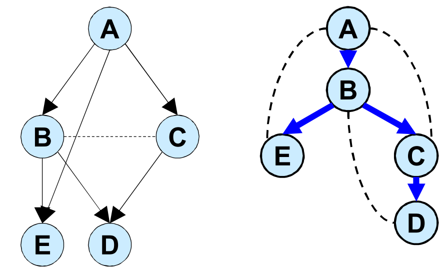
A	B=0	B=1
0	.4	.6
1	.1	.9

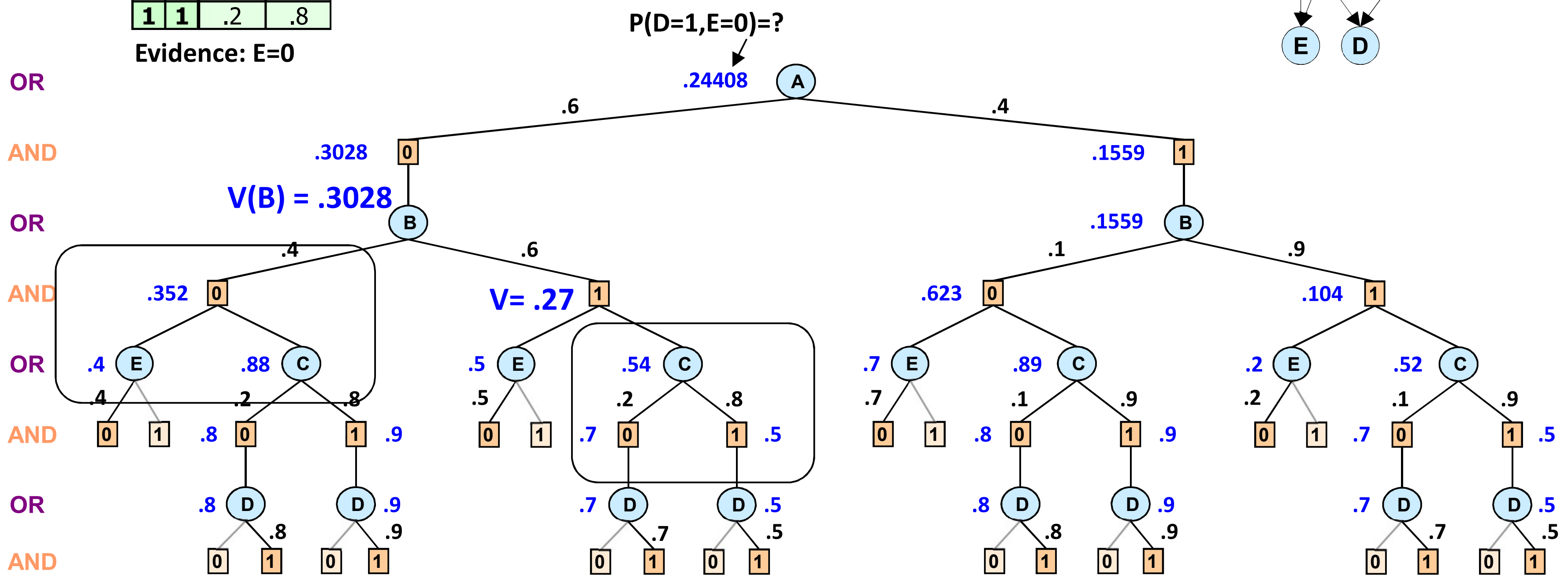
$P(C \mid A)$

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 using value for sub-problem below

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

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

In more detail:
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

$P(B \mid A)$

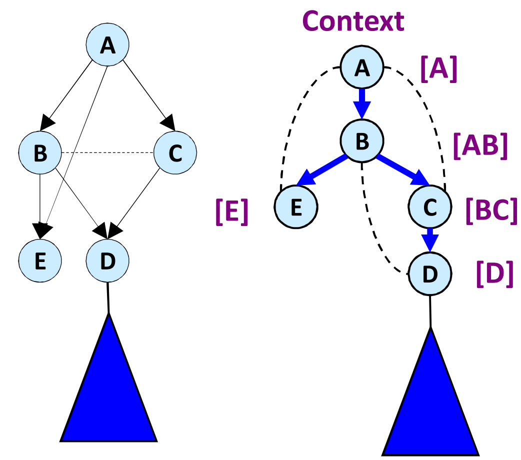
A	B=0	B=1
0	.4	.6
1	.1	.9

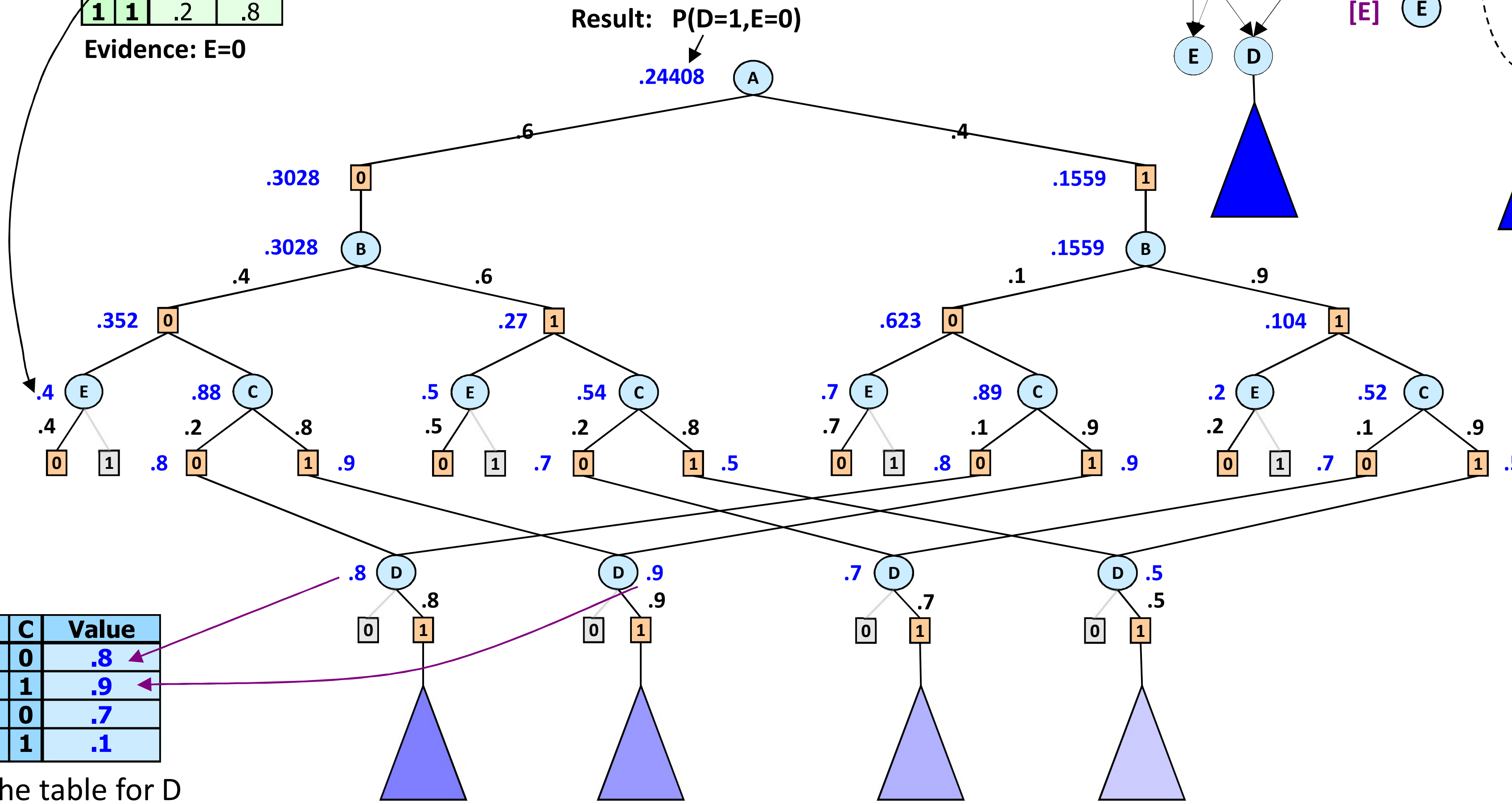
$P(C \mid A)$

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

$P(A)$

A	P(A)
0	.6
1	.4

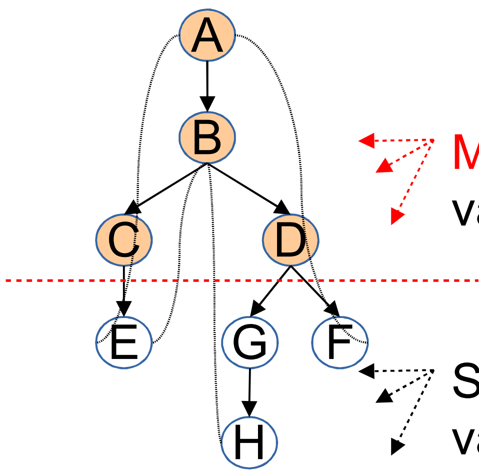




B	C	Value
0	0	.8
0	1	.9
1	0	.7
1	1	.1

Cache table for D

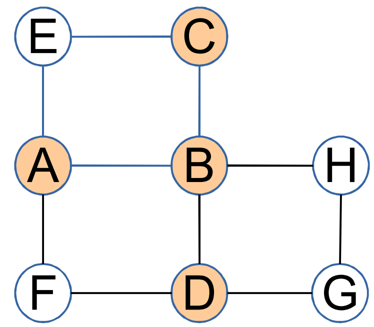
AND/OR search for Marginal MAP



MAP variables

SUM variables

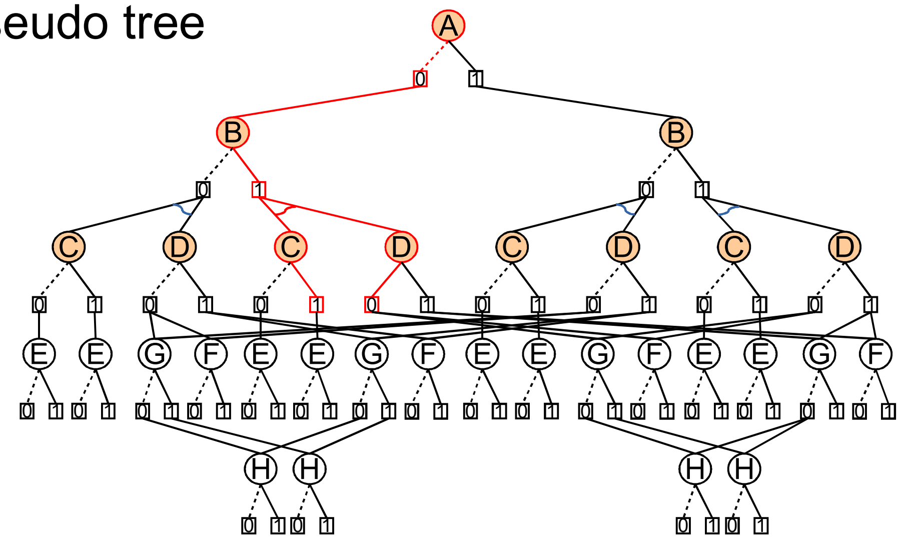
constrained pseudo tree



primal graph

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

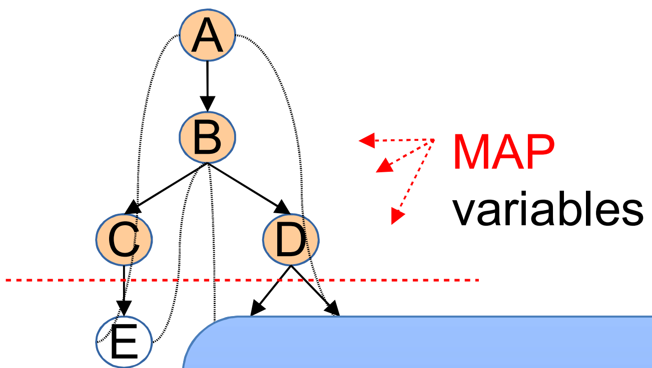
$X_S = \{E, F, G, H\}$

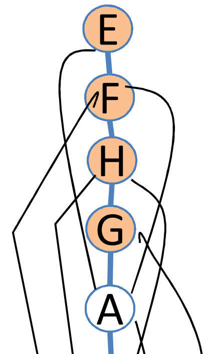


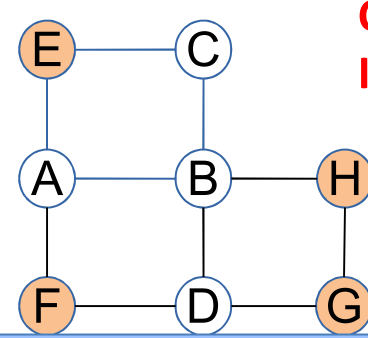
Node types
- OR (MAP): max
- OR (SUM): sum
- AND: multiplication

[Marinescu, Dechter and Ihler, 2014]

AND/OR Search for Marginal MAP



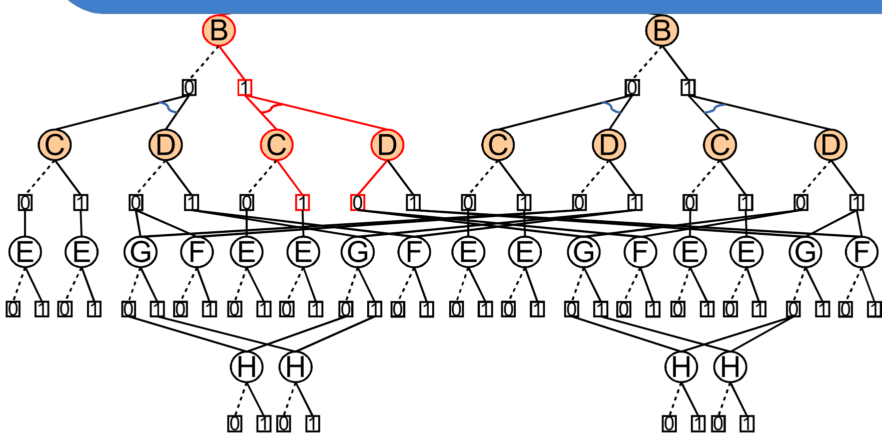


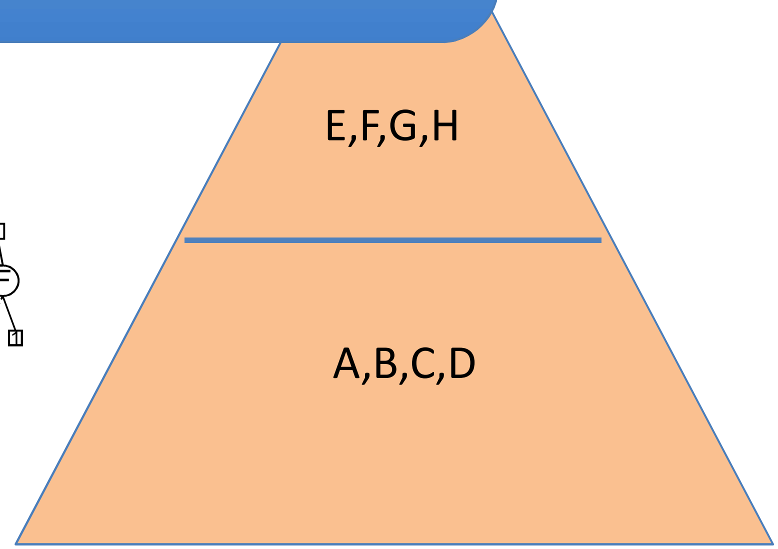


C-induced-width = 6
Induced-width = 2

const
pseu

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





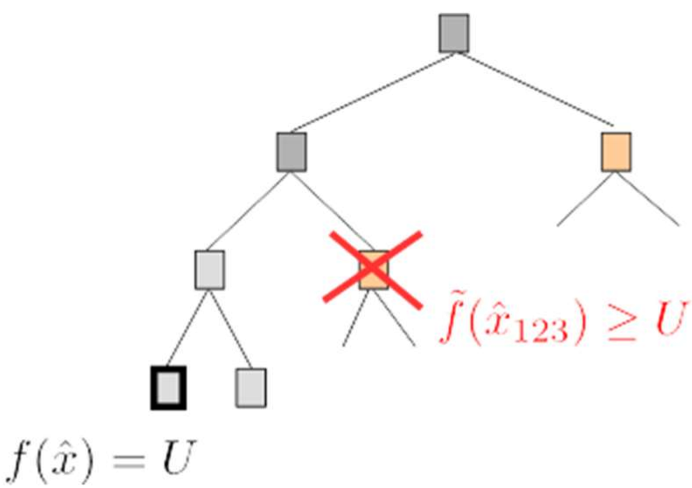
[Marinescu, Dechter and Ihler, 2014]

Basic Heuristic Search Schemes

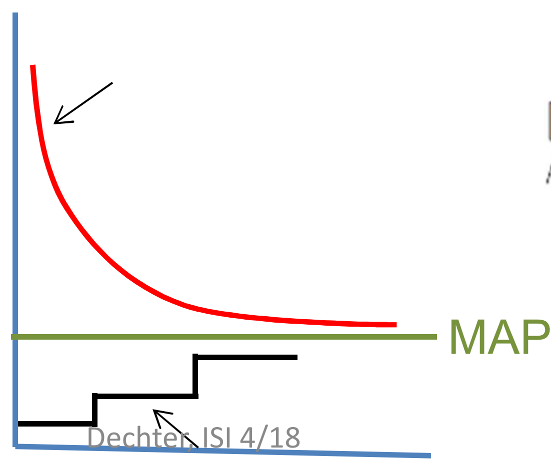
Heuristic function $\tilde{f}(\hat{x}_p)$ computes a lower bound on the best extension of partial configuration $\hat{x}_p$ and can be used to guide heuristic search.
We focus on:

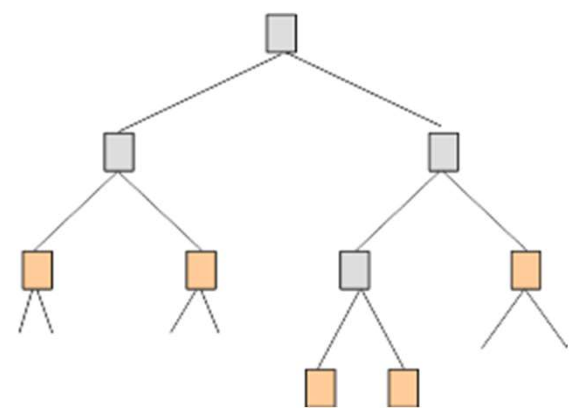
1. Branch-and-Bound
Use heuristic function $\tilde{f}(\hat{x}_p)$ to prune the depth-first search tree
Linear space

2. Best-First Search
Always expand the node with the lowest heuristic value $\tilde{f}(\hat{x}_p)$
Needs lots of memory



BnB is upper-bound anytime





Outline

- Background: Marginal Map and planning
- **Heuristic search meets probabilistic reasoning:**
  - AND/OR search
  - **Decomposition heuristics**
- Heuristic search schemes for Marginal Map:
  - Exact and anytime schemes
  - Anytime solvers
- Applying Marginal Map to planning
- Challenges and future plans

Decomposition-bound heuristics:

Mini-bucket

Tightening by Cost-shifting

Weighted min-bucket

Mini-Bucket Approximation

For optimization

Split a bucket into mini-buckets —> bound complexity

bucket (X) =

$$\{\ f_1,\ f_2,\ \dots\ f_r,\ f_{r+1},\ \dots\ f_n\ \}$$

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

$$\{\ f_1,\ \dots\ f_r\ \}$$

$$\lambda_{X,1}(\cdot) = \max_x \prod_{i=1}^{r} f_i(x,\dots)$$

$$\{\ f_{r+1},\ \dots\ f_n\ \}$$

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

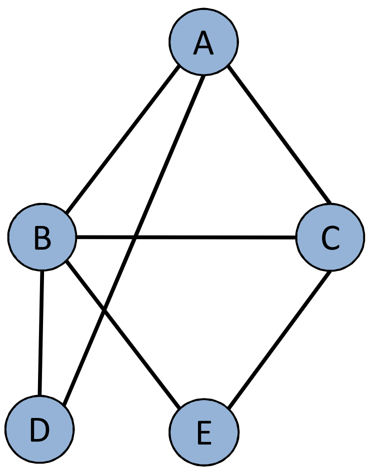
$$\lambda_X(\cdot)\ \le\ \lambda_{X,1}(\cdot)\ \lambda_{X,2}(\cdot)$$

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

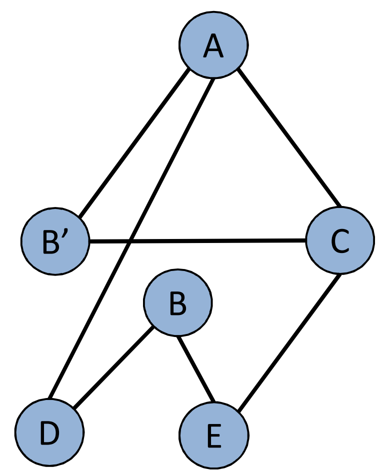
bucket C: $f(c,a)\ f(c,e)\ \lambda_{B\to C}(a,c)$

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

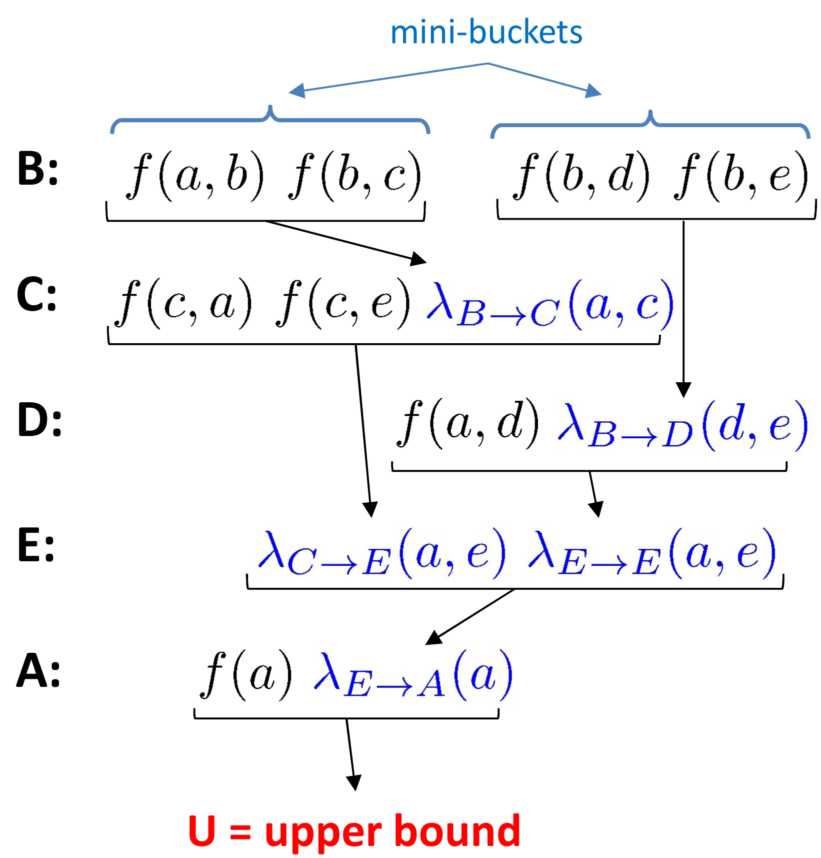
$$\mathbf{b}^* = \arg\max_b f(a^*, b) \cdot f(b, c^*) \cdot f(b, d^*) \cdot f(b, e^*)$$

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

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

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

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

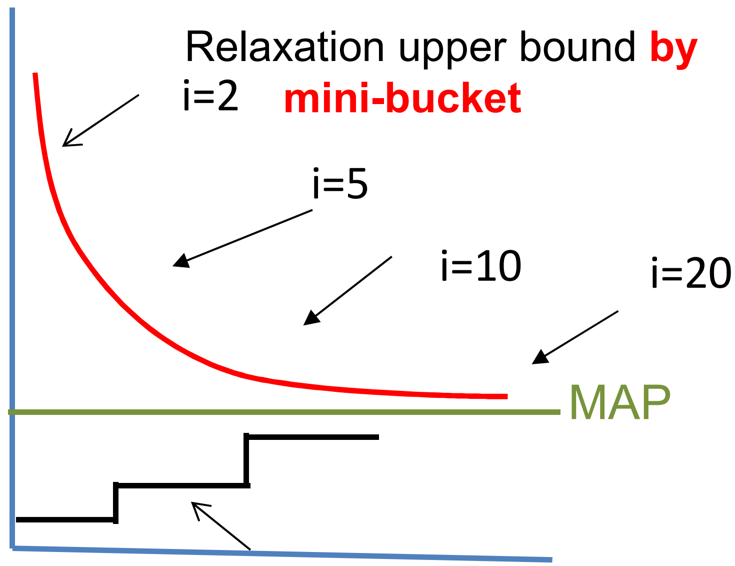


Greedy configuration = lower bound

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

(Reparameterization, or cost-shifting)

$+\lambda(B)$

A	B	f(A,B)
b	b	6 + 3
b	g	0 - 1
g	b	0 + 3
g	g	6 - 1

+

$-\lambda(B)$

B	C	f(B,C)
b	b	6 - 3
b	g	0 - 3
g	b	0 + 1
g	g	6 + 1

A	B	C	f(A,B,C)
b	b	b	12
b	b	g	6
b	g	b	0
b	g	g	6
g	b	b	6
g	b	g	0
g	g	b	6
g	g	g	12

= 0 + 6

B	λ(B)
b	3
g	-1

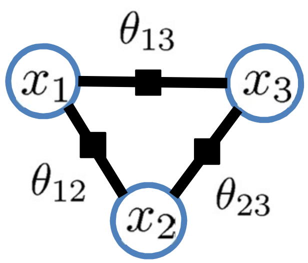
Modify the individual functions

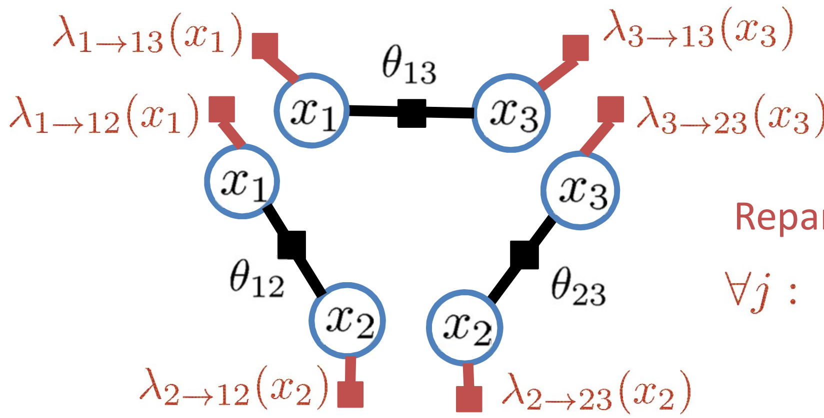
- but -

keep the sum or product of functions unchanged

Tightening the Bound

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





Reparameterization:

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

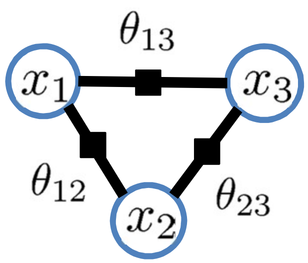
$$\log f(\mathbf{x}^*) = \max_{\mathbf{x}} \sum_{\alpha} \theta_\alpha(\mathbf{x}_\alpha) \quad \leq \min_{\{\lambda_{i\to\alpha}\}} \sum_{\alpha} \max_{\mathbf{x}_\alpha} \Big[\theta_\alpha(\mathbf{x}_\alpha) + \sum_{i\in\alpha} \lambda_{i\to\alpha}(x_i)\Big]$$

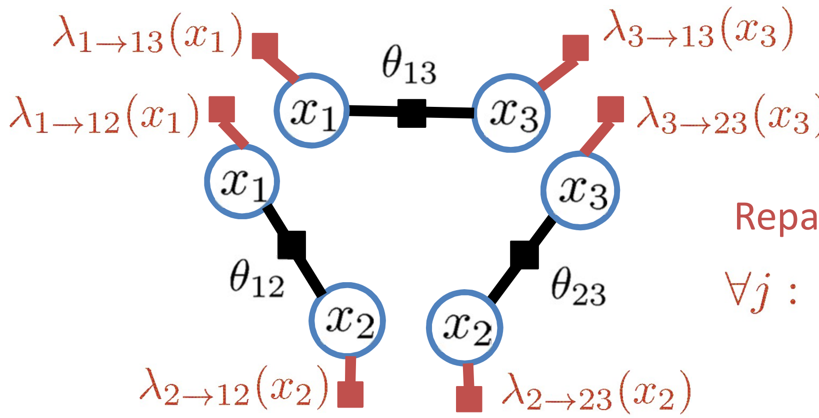
- Bound solution using decomposed optimization
- Solve independently: optimistic bound

- 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:

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

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

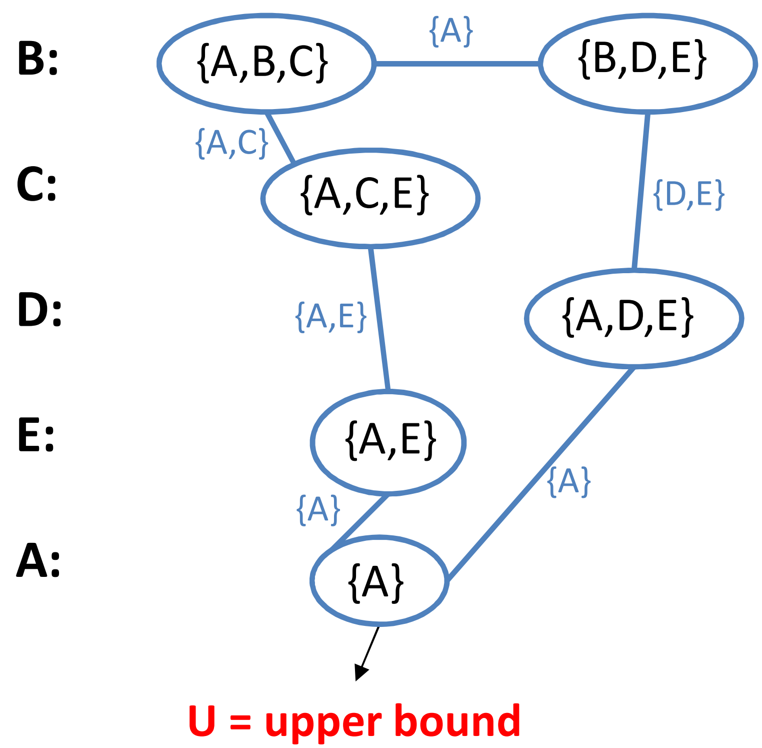
- Many names for the same class of bounds
  - Dual decomposition [Komodakis et al. 2007]
  - TRW, MPLP [Wainwright et al. 2005; Globerson & Jaakkola 2007]
  - Soft arc consistency [Cooper & Schiex 2004]
  - Max-sum diffusion [Warner 2007]

Mini-Bucket with Moment-Matching

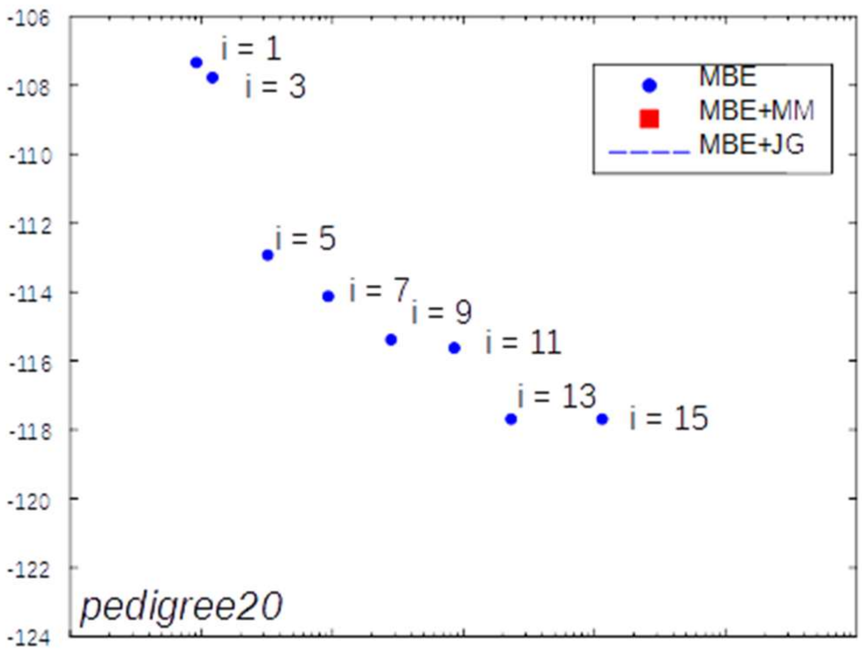
[Ihler et al. 2012]

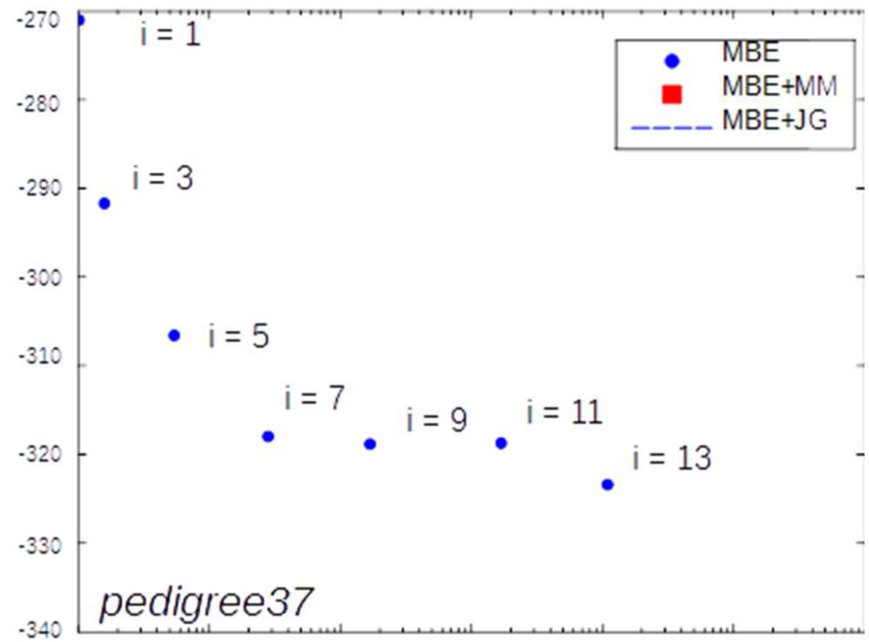
- Downward pass as cost shifting
- Can also do cost shifting within mini-buckets:
  "Join graph" message passing
- "Moment-matching" version:
  One message exchange within each bucket, during downward sweep
- Optimal bound defined by cliques ("regions") and cost-shifting f'n scopes ("coordinates")

Join graph:



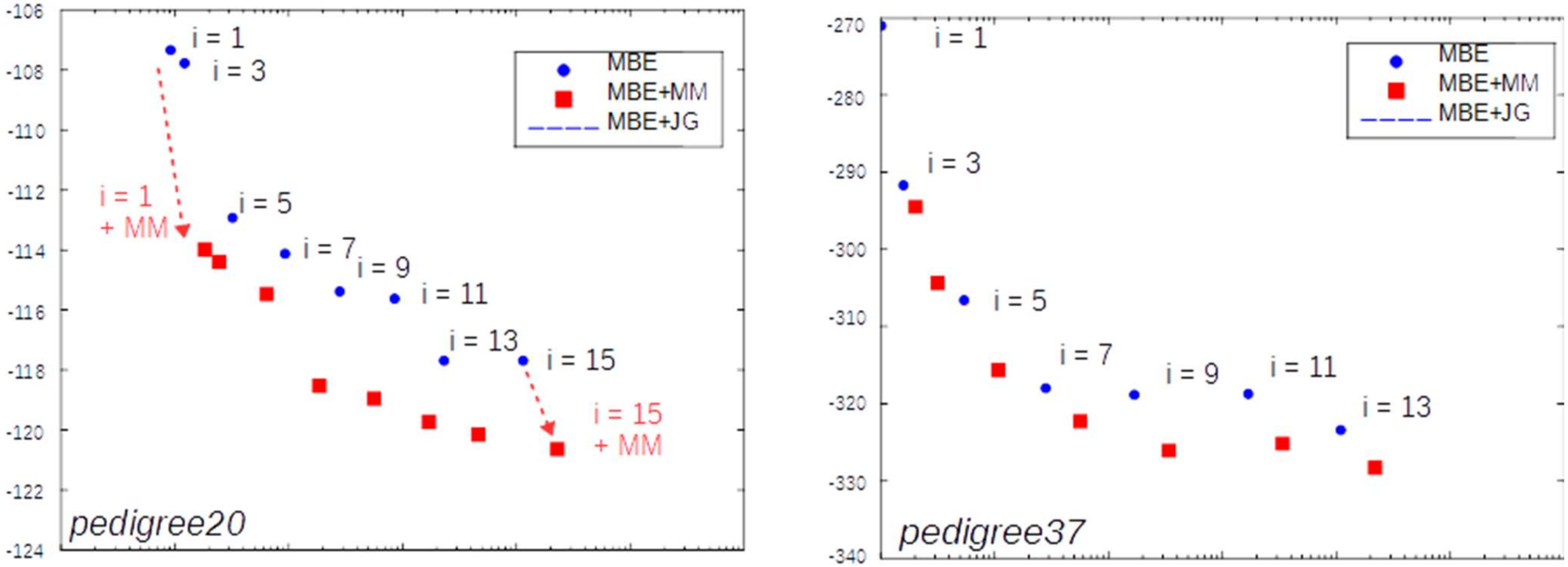
Anytime Approximation





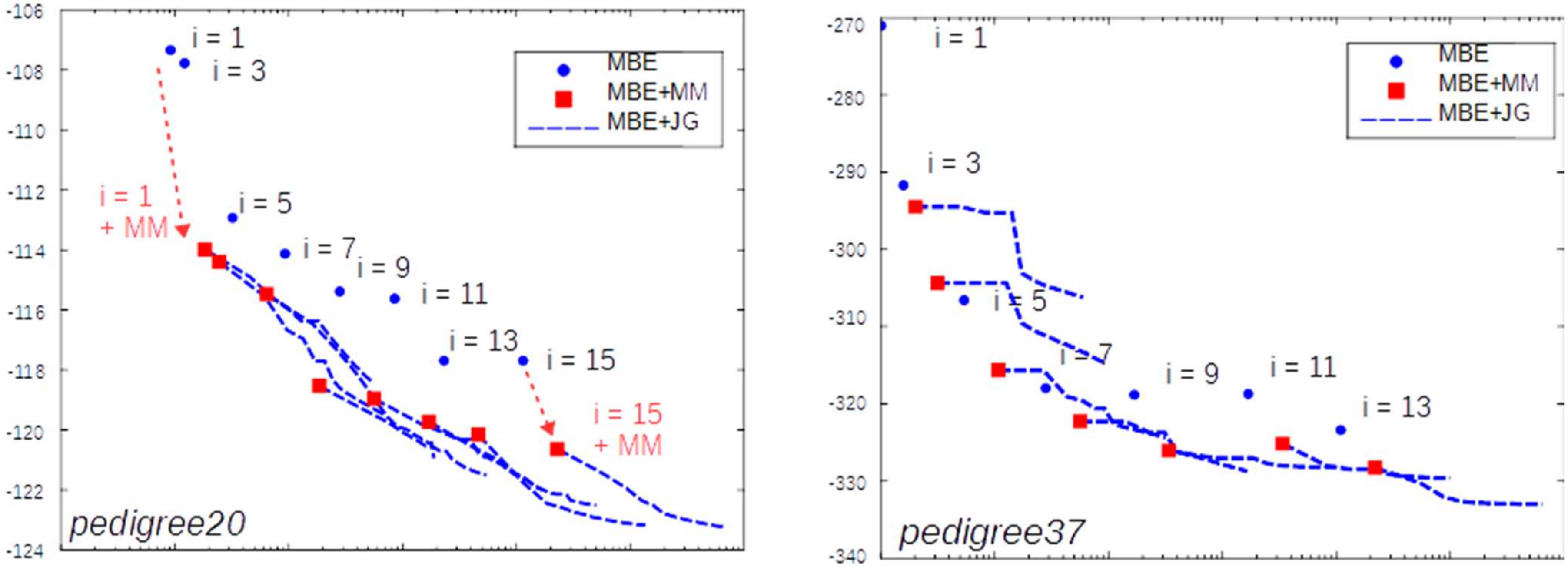
- Can tighten the bound in various ways
  - Cost-shifting (improve consistency between cliques)
  - Increase i-bound (higher order consistency)
- Simple moment-matching step improves bound significantly

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

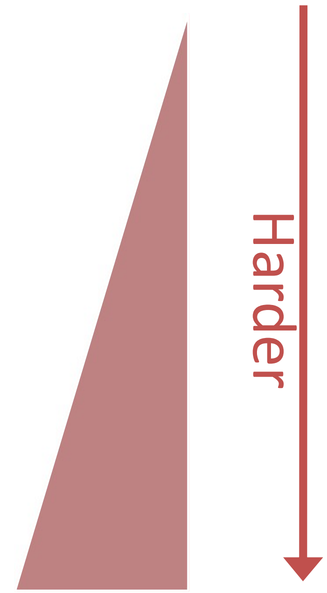
Anytime Approximation



- 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

Types of Queries

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



- **Pure-mini-bucket is extremely weak for summation**

Mini-Bucket for Summation

(Liu &Ihler, 2011)

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

- Generalize technique to sum via Holder's inequality:

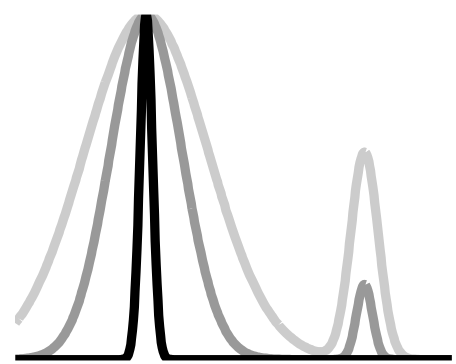
$$\sum_x f_1(x) \cdot f_2(x) \quad \leq \quad \Big[\sum_x f_1(x)^{\frac{1}{w_1}}\Big]^{w_1} \cdot \Big[\sum_x f_2(x)^{\frac{1}{w_2}}\Big]^{w_2}$$

$$w_1 + w_2 = 1$$

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

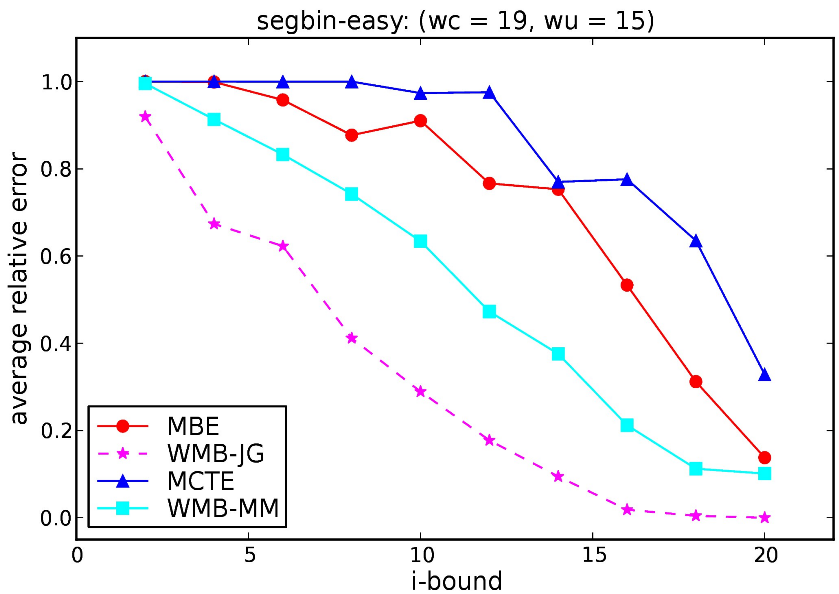
$\max_A$ bucket A: $f(a)$ $\lambda_{E\to A}(a)$

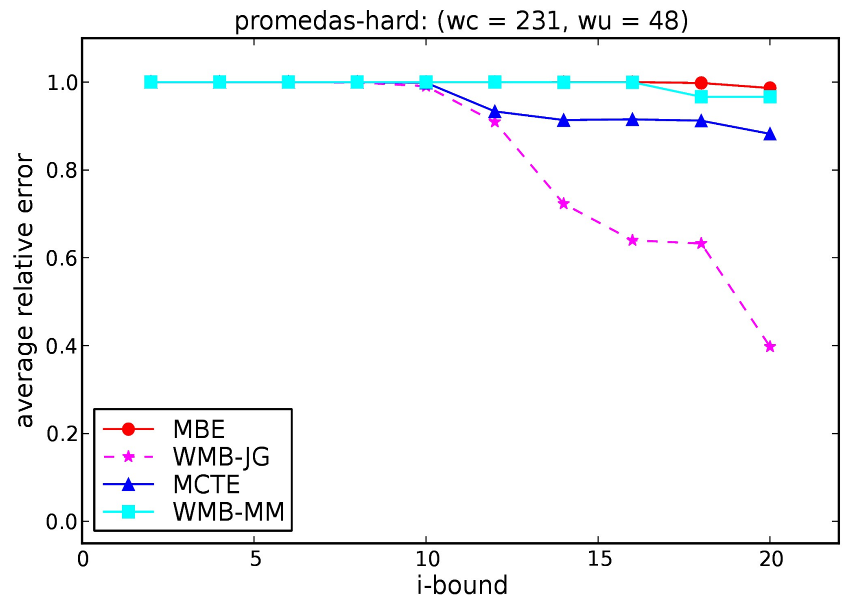
U = upper bound

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

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

MMAP: Quality of Upper Bounds





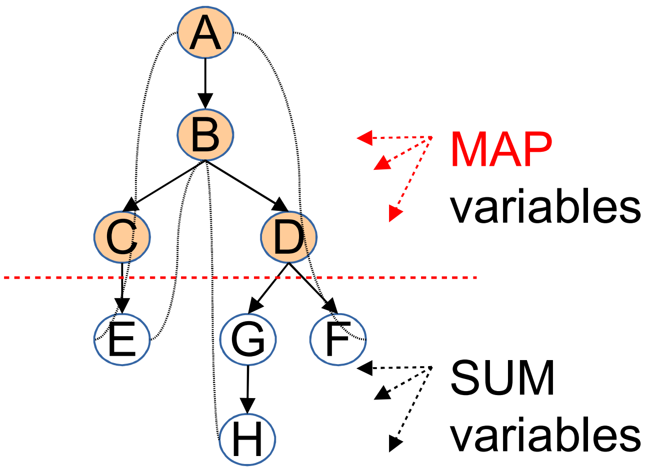
Average relative error wrt tightest upper bound. 10 iterations for WMB-JG(i).

Outline

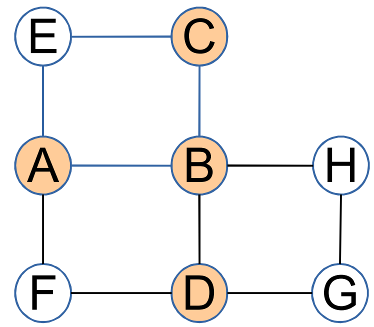
- Graphical models, marginal map and planning
- Heuristic search meets probabilistic reasoning:
  - AND/OR search spaces
  - Decomposition bounds
- MMAP AND/OR search with WMB heuristics
  - Exact search
  - Anytime search
- Marginal Map for planning
- Challenges and future plans

Exact search

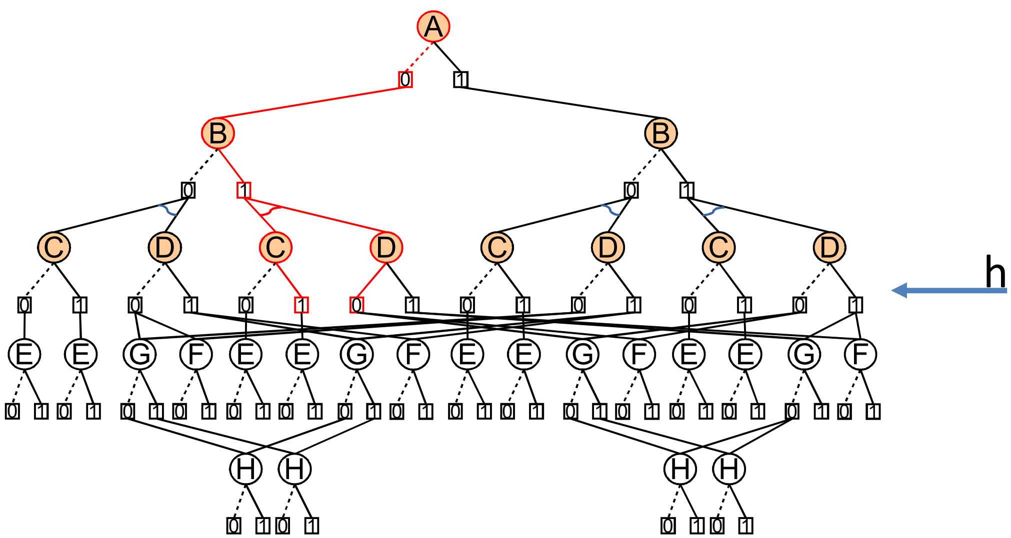
AND/OR Search for Marginal MAP

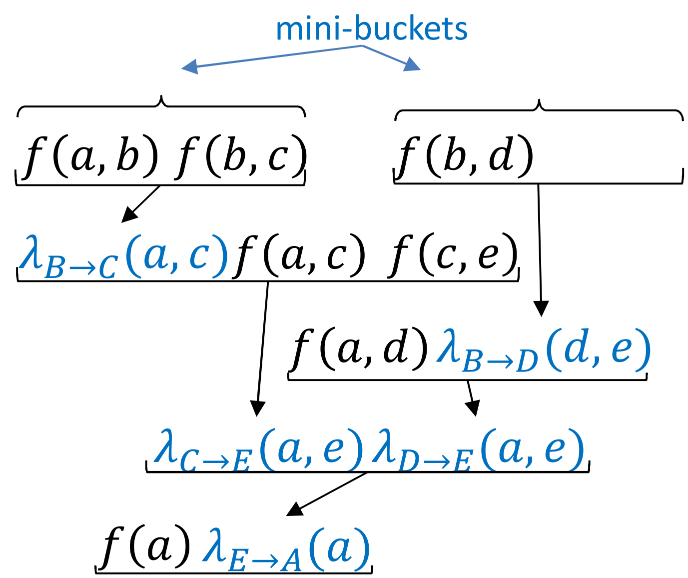


constrained pseudo tree



primal graph

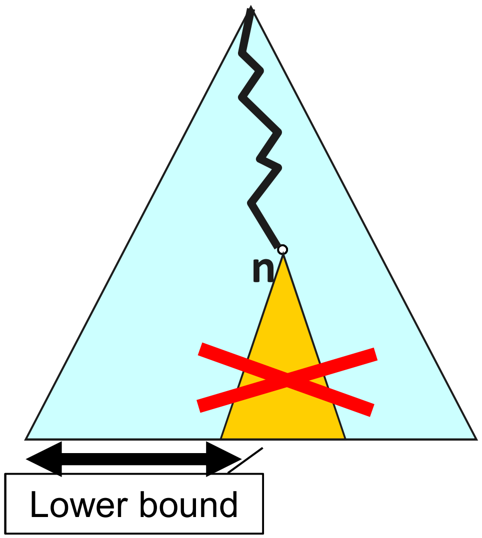




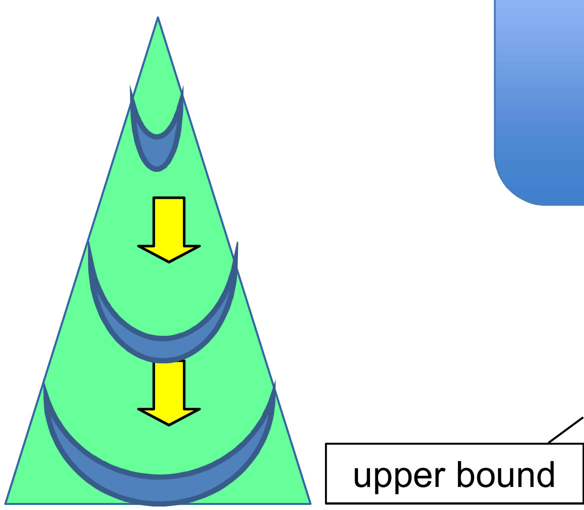
[Marinescu, Dechter and Ihler, 2014]

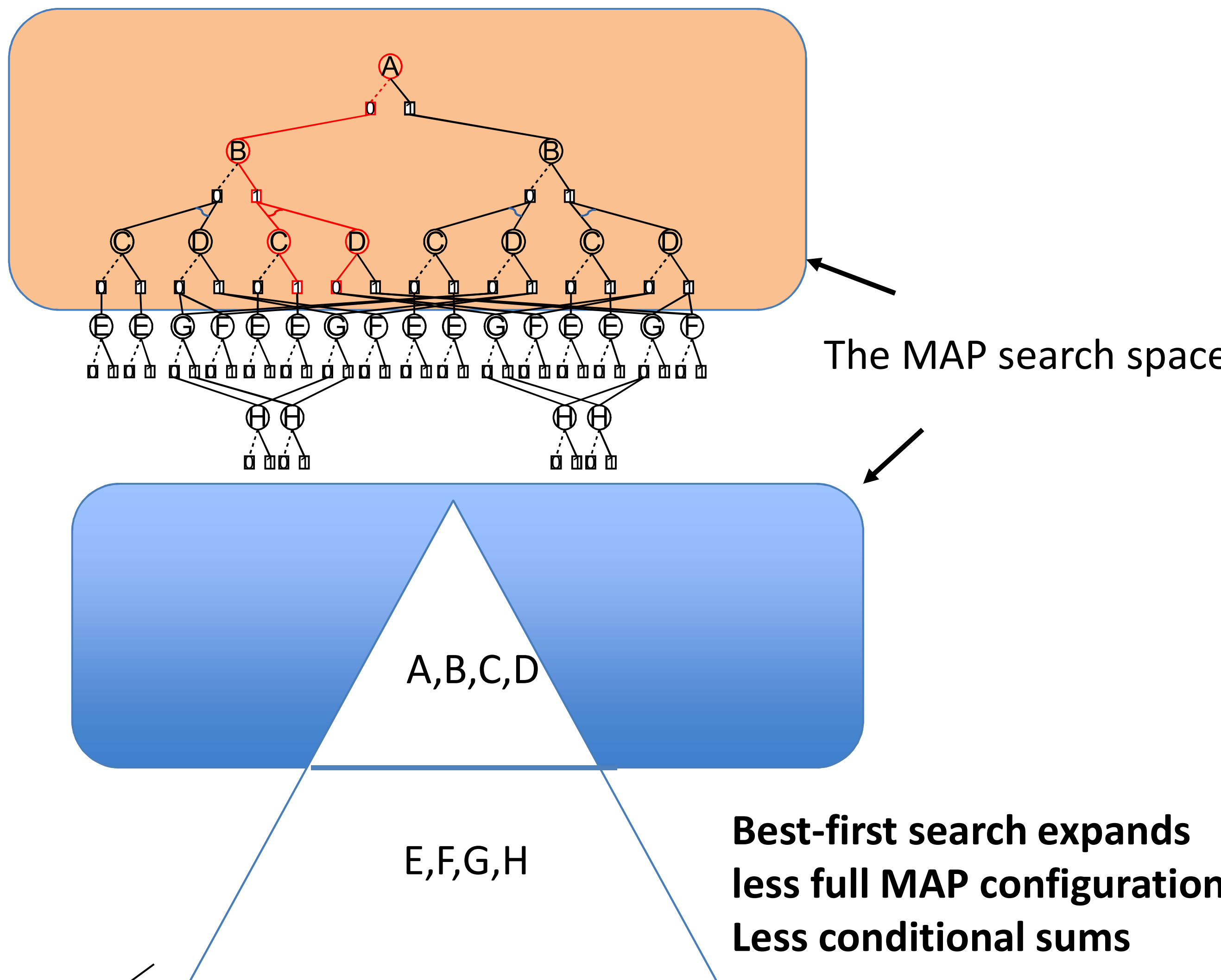
Exact Solvers: Best or Depth-First Search?

Depth-First search



Best-First search

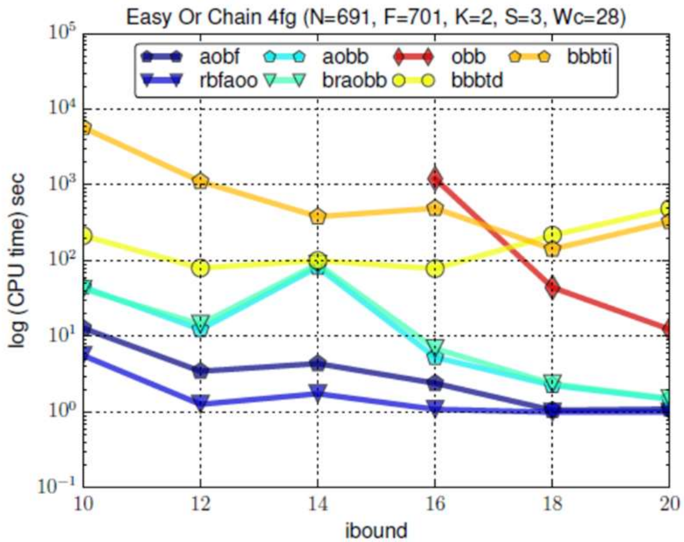


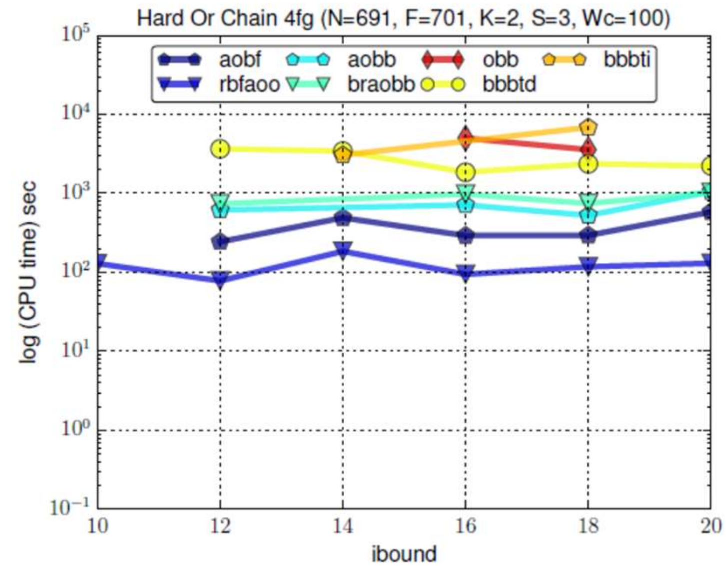


Best-first search expands less full MAP configurations Less conditional sums

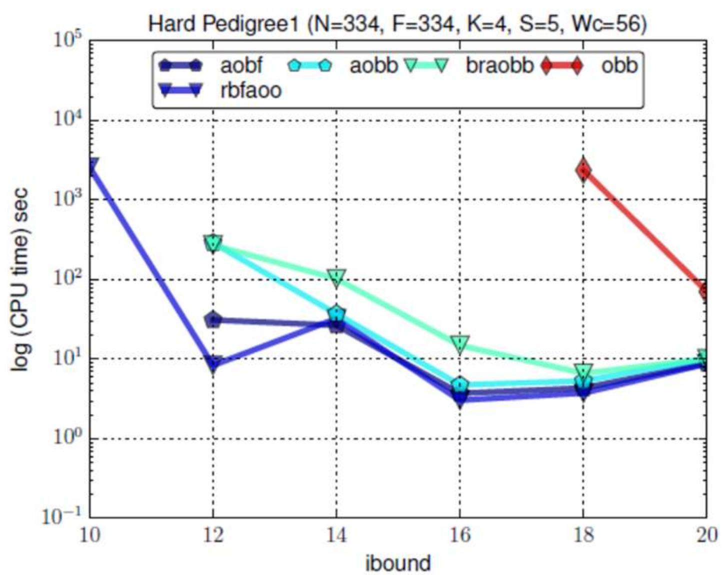
Results: Exact AND/OR solvers

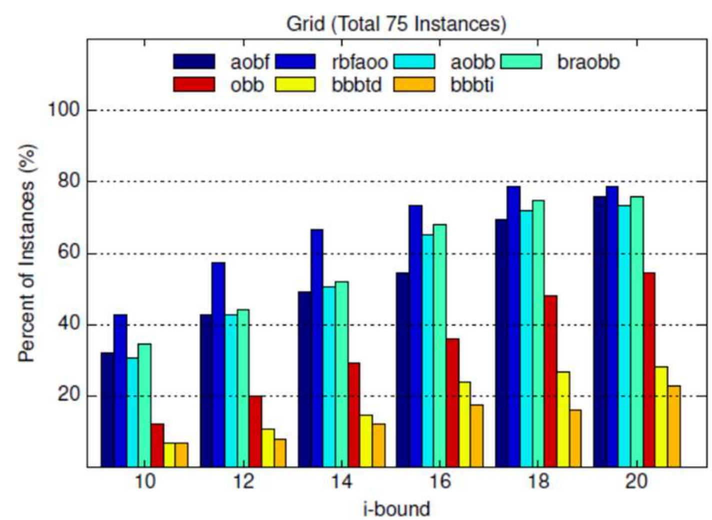
Benchmarks:
Grids (128)
Pedigrees (88)
Promedas (100)





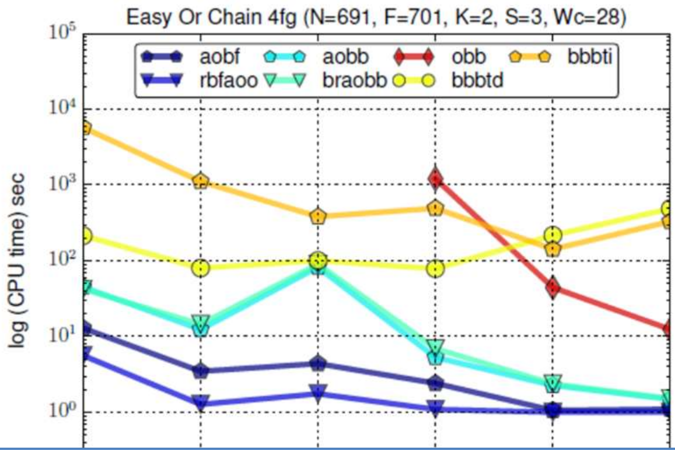
AOBF
RBFAOO - recursive
BRAOBB
Yuan, Park BBTDi, BBBTD
Time-bound 2 hours

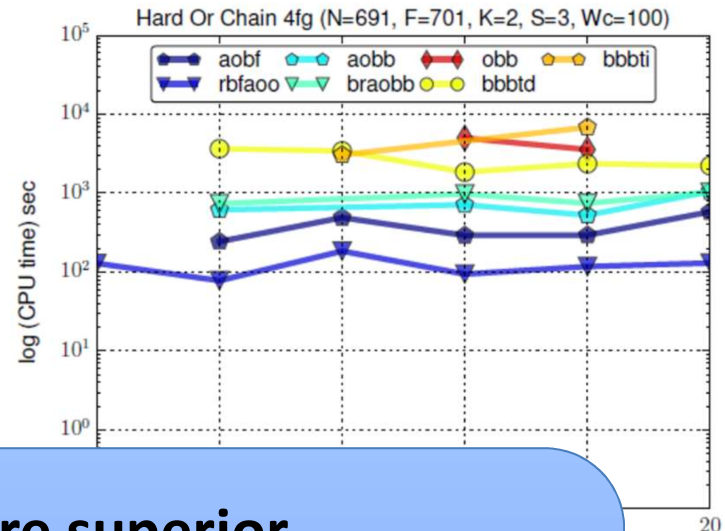




Results: Exact AND/OR solvers

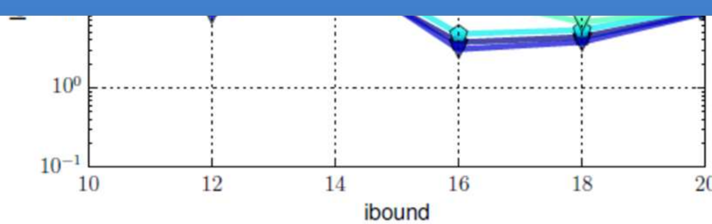
Benchmarks:
Grids (128)
Pedigrees (88)
Promedas (100)

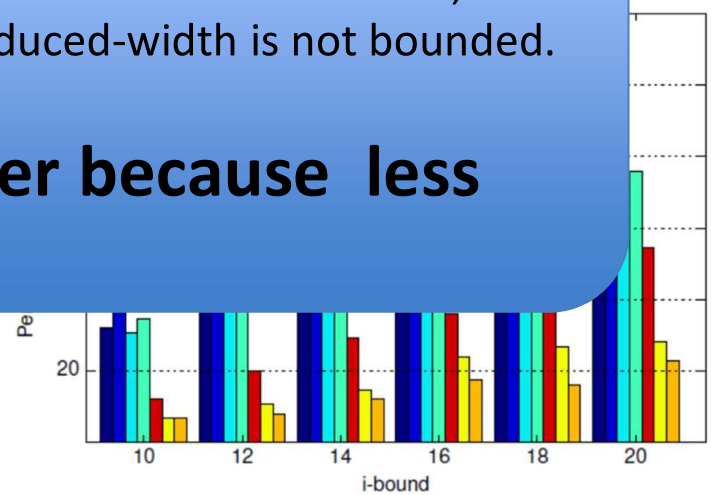




AOBF
RBFA
BRAO
Yuan,
Time-

- **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 search yielding bounds

Ideas:

1. Weighted Heuristics
2. Alternate Best and Depth search

To yield upper and lower bound in an anytime way

Anytime AND/OR solvers

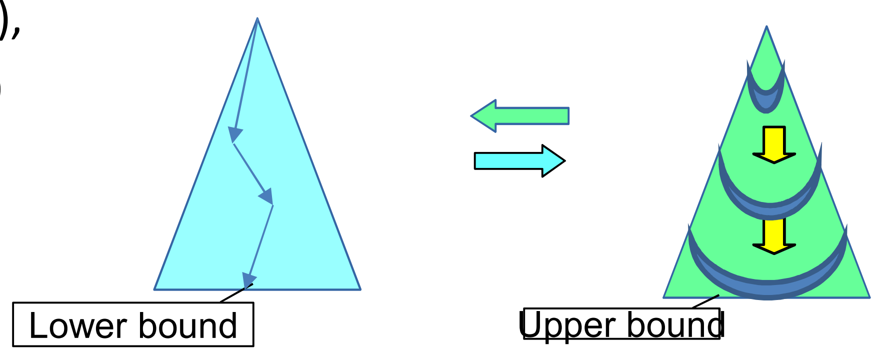
- **Weighted Heuristic:** [Lee et. al. AAAI-2016]
  - Weighted Restarting AOBF (WAOBF)
  - Weighted Restarting RBFAOO (WRBFAOO)
  - Weighted Repairing AOBF (WRAOBF)

Weighted A* search [Pohl 1970]
- non-admissible heuristic
- Evaluation function:

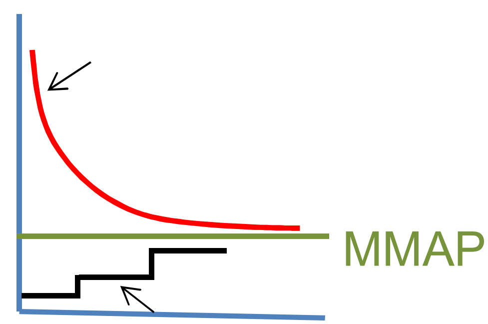
$$f(n)=g(n)+w\cdot h(n)$$

- **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)

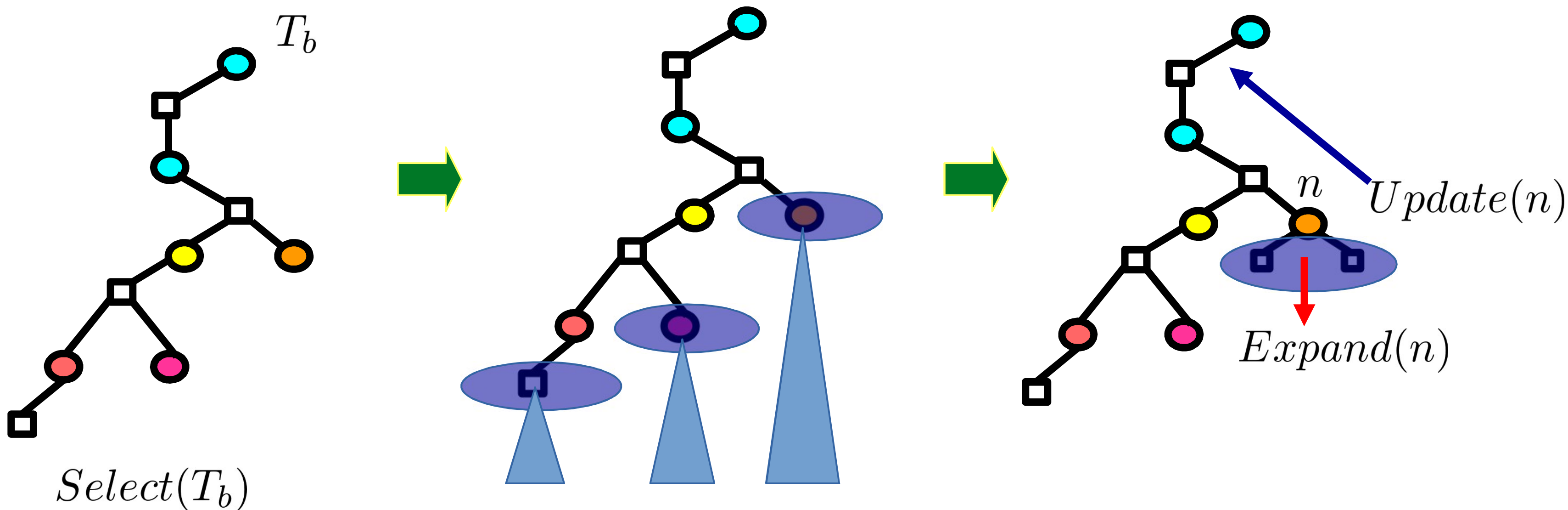


Goal: anytime bounds
And anytime solution



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

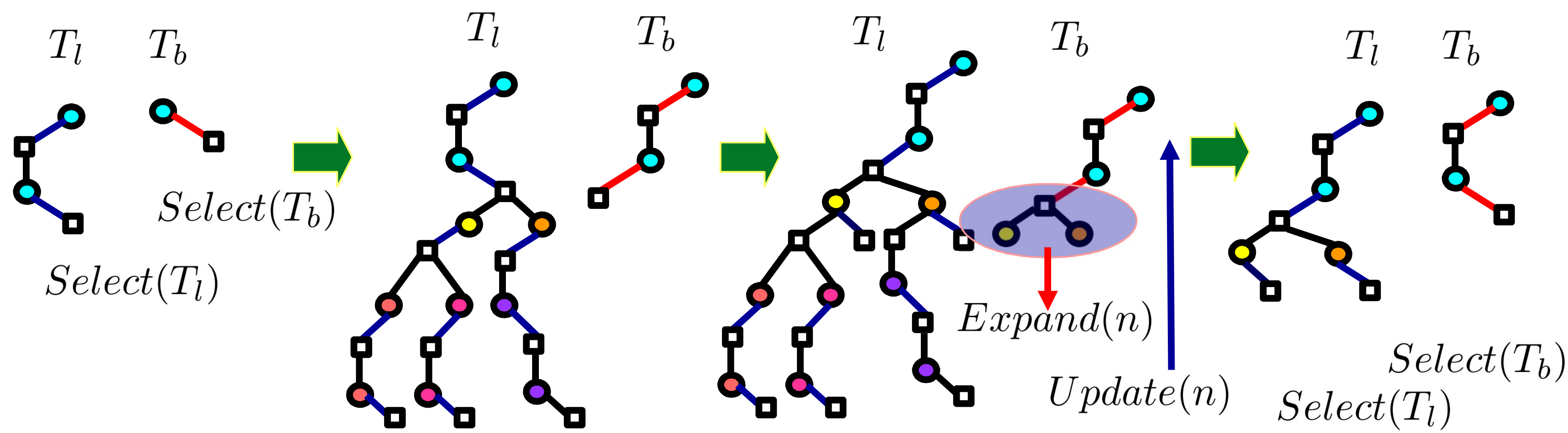
cutoff parameter: perform depth-first dive at every θ number of node expansions.
best partial solution tree: T_b

AAOBF (alternating best- and depth-first)

Depth-first selection
Best-first selection

Depth-first greedy expansion
Best-first re-direct T_b

Depth-first re-direct T_l
Best-first expansion & update



- Expand(n) and Update(n)
  $n \in T_l$
- depth-first greedy search
  $\max w(n,m)q(m)$
- redirect T_l from explicated search graph with updated q and l $\min \frac{l(n)}{w(n,m)q(m)}$
- select T_b
  Expand and Update a tip node

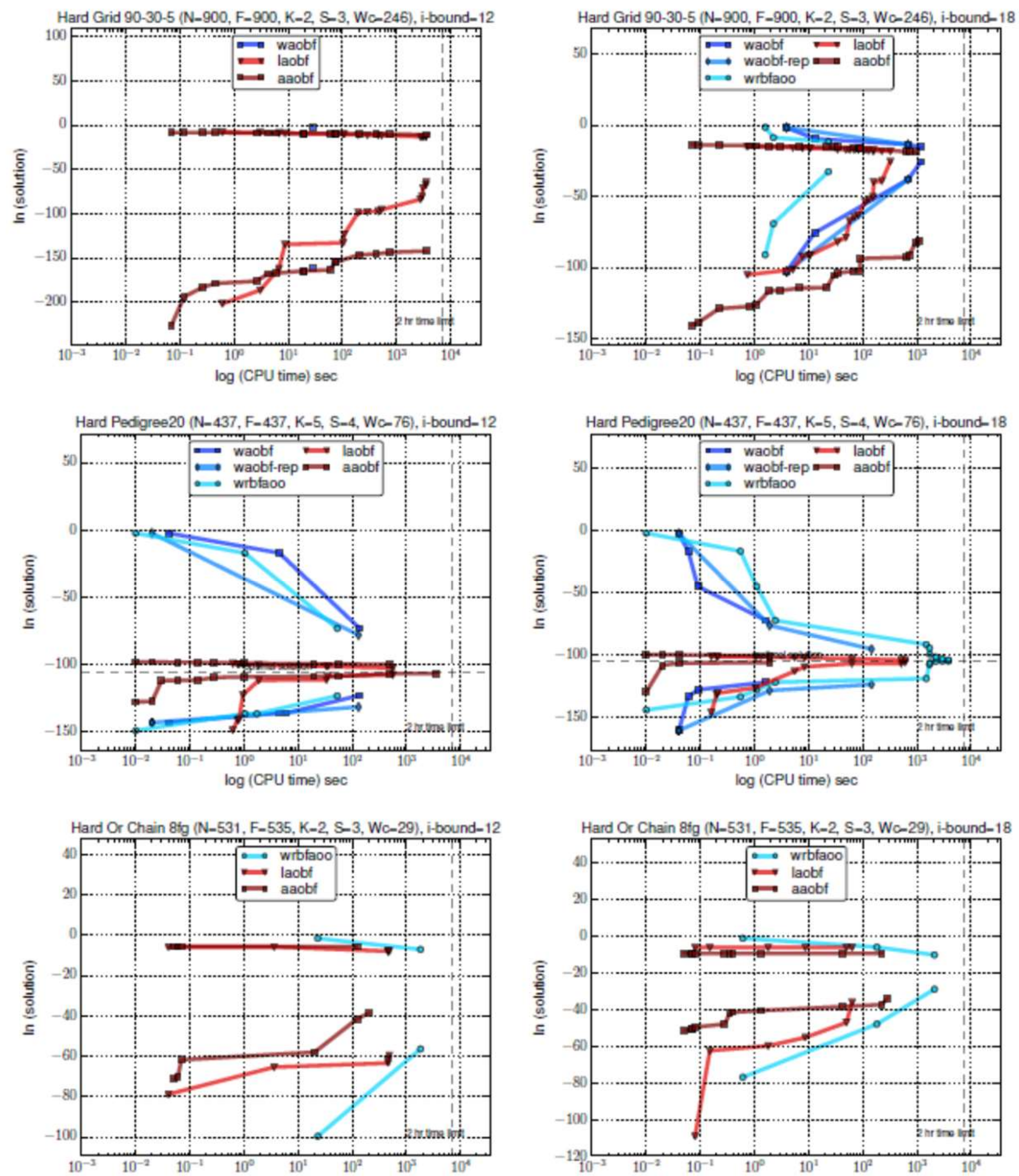
Anytime Bounding of Marginal MAP

(UAI'14, IJCAI'15, AAAI'16, AAAI'17, (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.



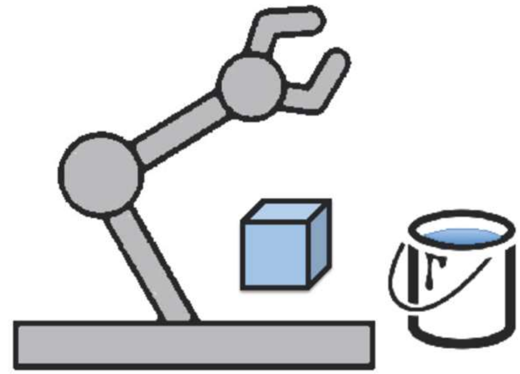
Outline

- Graphical models, marginal map and planning
- Heuristic search meets probabilistic reasoning:
  - AND/OR search spaces
  - Decomposition bounds
- MMAP AND/OR search with WMB heuristics
  - Exact search
  - Anytime search
- Marginal Map for planning
- Challenges and future plans

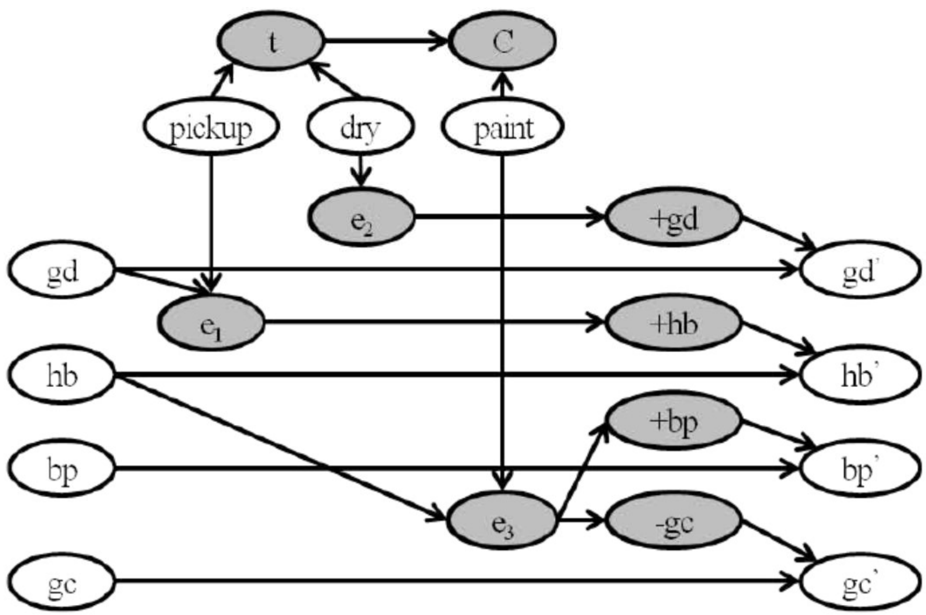
Compiling PPDDL into 2TDBN

[Slippery Gripper Domain Example

[Lee, Marinescu, and Dechter. *ISAIM 2016].*
[Lee 2014, (master thesis)]



Express it in the UAI format



pickup	f(pickup)
0	1
1	1

pickup	gd	e_1	Pr(e_1|)
0	0	noop	1
0	0	hb	0
0	0	null	0
0	1	noop	1
0	1	hb	0
0	1	null	0
1	0	noop	0
1	0	hb	0.5
1	0	null	0.5
1	1	noop	0
1	1	hb	0.95
1	1	null	005

e_1	+hb	f(+hb)
noop	0	1
noop	1	0
hb	0	0
hb	1	1
null	0	1
null	1	0

+hb	hb	hb'	f(hb')
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

PPDDL vs. FDR(SAS+) translation

instance blocks, horizon	ppddl to dbn n, a, w_c, h_c	i_{best}	braobb-mmap time (sec)	braobb-mmap pr(G)	sas+ to dbn n, a, w_c, h_c	i_{best}	braobb-mmap time (sec)	braobb-mmap pr(G)
2, 5	299, 40, 48, 76	10	1.56	0.703125	406, 5, 22, 64	2	1.65	0.703125
2, 8	473, 64, 72, 112	10	2990.73	0.91626	646, 8, 24, 76	14	1857.33	0.91626
2, 11	647, 88, 96, 149	16	oot	0.966007	886, 11, 24, 86	6	oot	0.943176
2, 14	821, 112, 120, 169	2	oot	0.91626	1126, 14, 28, 100	8	oot	0.91626
2, 17	995, 136, 144, 199	10	oot	0.91626	1366, 17, 28, 108	10	oot	0.91626
2, 20	1169, 160, 163, 237	2	oot	0.870117	1606, 20, 25, 103	2	oot	0.870117
3, 5	741, 90, 132, 182	6	2.53	0.079102	833, 5, 44, 85	4	0.96	0.079102
3, 8	1176, 144, 159, 251	6	5767.69	0.494385	1328, 8, 45, 125	4	4382.65	0.494385
3, 11	1611, 198, 213, 328	10	oot	0.494385	1823, 11, 45, 132	2	oot	0.494385
3, 14	2046, 252, 267, 401	10	oot	0.454834	2318, 14, 45, 145	2	oot	0.494385
3, 17	2481, 306, 326, 474	2	oot	0.395508	2813, 17, 44, 183	4	oot	0.494385
3, 20	2916, 360, 380, 545	2	oot	0.395508	3308, 20, 44, 178	6	oot	0.494385
4, 8	2185, 256, 370, 477	10	108.7	0.177979	2266, 8, 67, 164	6	55.04	0.177979
4, 9	2455, 288, 415, 520	12	5717.1	0.222473	2548, 9, 68, 188	2	2291.27	0.222473
4, 10	2725, 320, 397, 556	2	oot	0.222473	2830, 10, 68, 179	2	oot	0.222473
4, 11	2995, 352, 491, 624	2	oot	0.222473	3112, 11, 68, 214	2	oot	0.222473
4, 13	3535, 416, 541, 716	2	oot	0.222473	3676, 13, 68, 222	2	oot	0.222473
4, 15	4075, 480, 672, 841	10	oot	0.222473	4240, 15, 82, 263	2	oot	0.222473

- Translation from FDR(SAS+)
  - 1.3 ~ 2.6 times speed up
  - constrained induced width of problem is much less

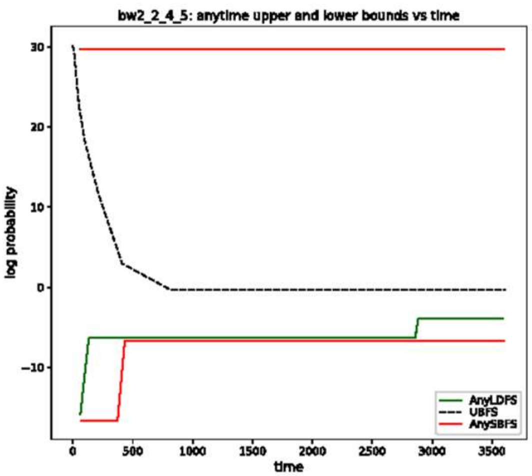
New Generation Algorithms (Approximate Summation)

[Lou, Dechter, Ihler, AAAI-2018: "Anytime Anyspace AND/OR Best-first Search for Bounding Marginal MAP"]

[Marinescu, Ihler, Dechter: (under review): "Stochastic Anytime Search for Bounding Marginal MAP"]

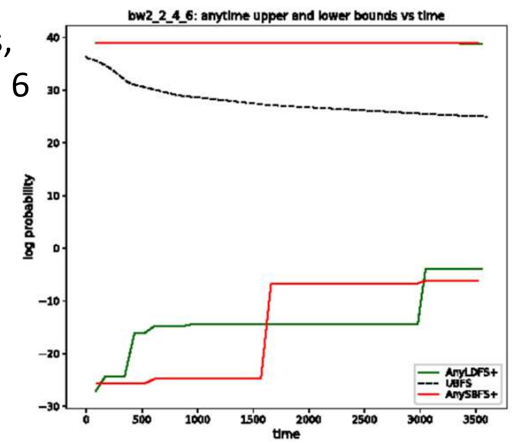
(n,k,c,w*,h)

(314,3,317,56,248)



2 blocks,
T=5 and 6

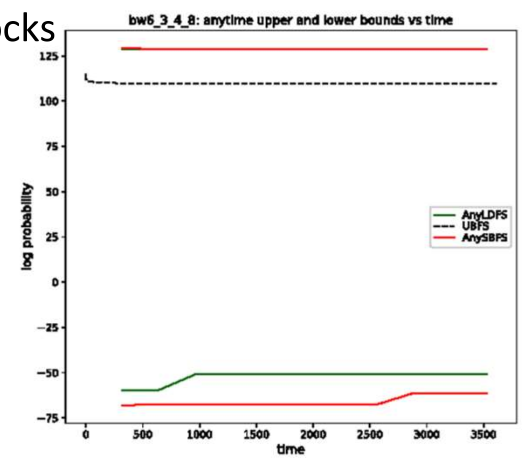
(375,3,378,64,302)



Algorithms:
UBFS
ANYLDFS
AnySBFS

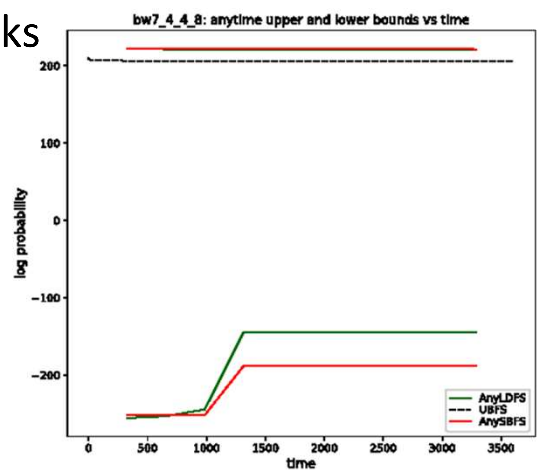
(1134,3,1044,173,908)

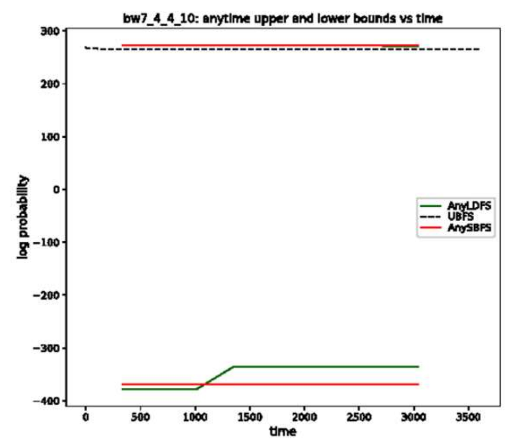
6 blocks
T=8



(2161,3,2168,302,1484

7 blocks
T=8

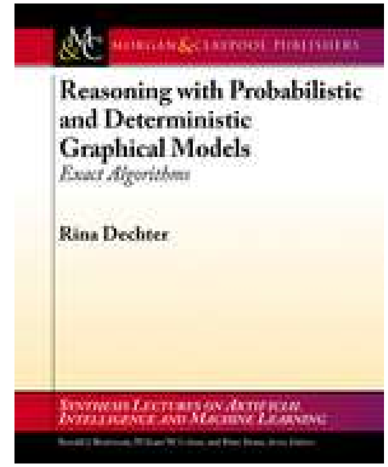


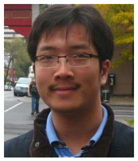


Conclusion

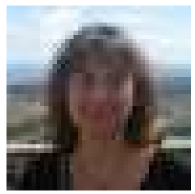
- Reasoning with graphical models using heuristic AND/OR search guided by decomposition-based heuristics
- Applied this approach to MMAP producing anytime upper and lower-bounds
- Empirical evaluation including some panning instances.
- Challenge for planning
  - Avoid generate the full multi-horizon model explicitly
  - Avoid generating a grounded model
  - Avoid the UAI format, by working directly with a generative planning generative model like PPDDL or RRDDL
- Move to influence diagrams

Thank you



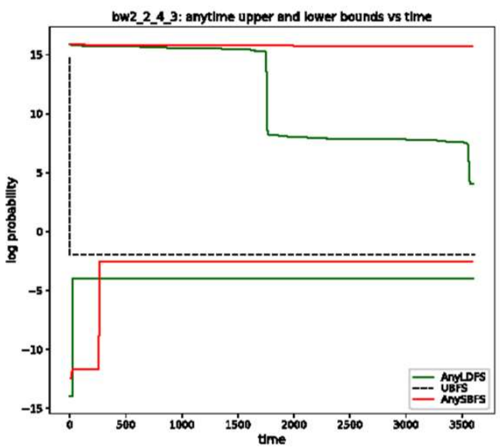




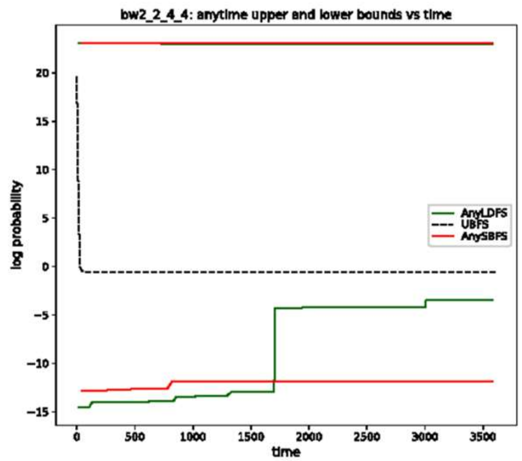



Alex Ihler
Kalev Kask
Irina Rish
Bozhena Bidyuk
Robert Mateescu
Radu Marinescu
Vibhav Gogate
Emma Rollon
Lars Otten
Natalia Flerova
Andrew Gelfand
William Lam
Junkyu Lee

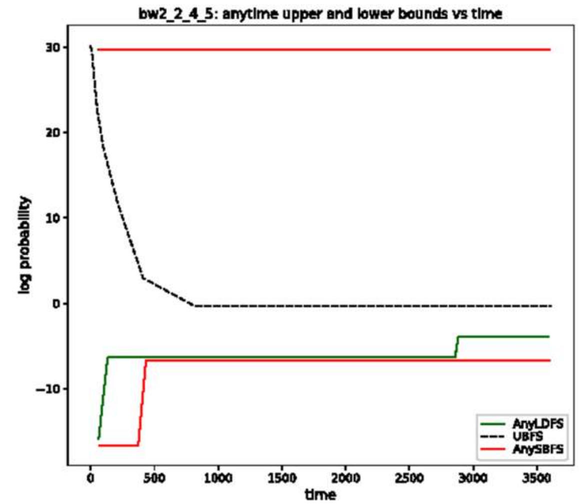
2 blocks



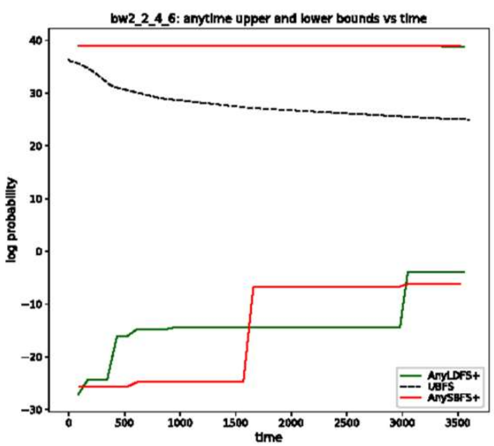
bw2_2_4_3: anytime upper and lower bounds vs time
log probability
time
AnyLDFS
UBFS
AnySBFS



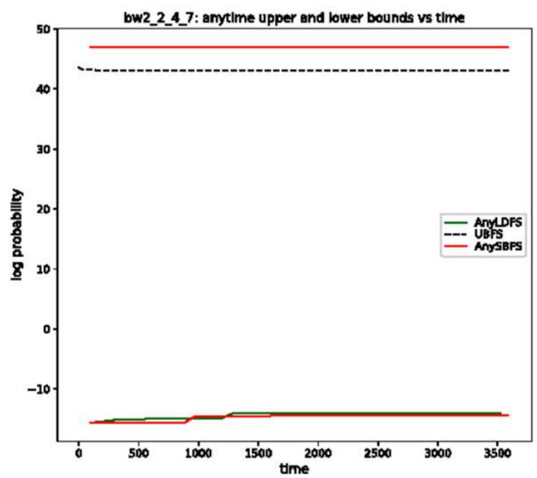
bw2_2_4_4: anytime upper and lower bounds vs time
log probability
time
AnyLDFS
UBFS
AnySBFS



bw2_2_4_5: anytime upper and lower bounds vs time
log probability
time
AnyLDFS
UBFS
AnySBFS

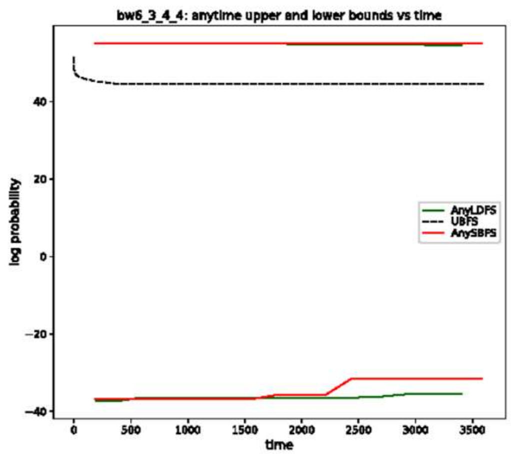


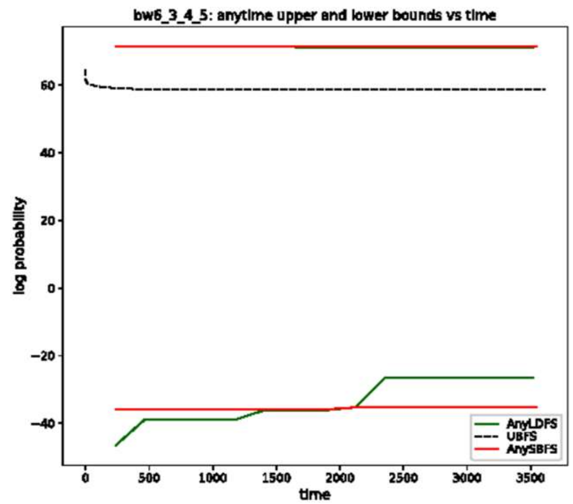
bw2_2_4_6: anytime upper and lower bounds vs time
log probability
time
AnyLDFS+
UBFS
AnySBFS+

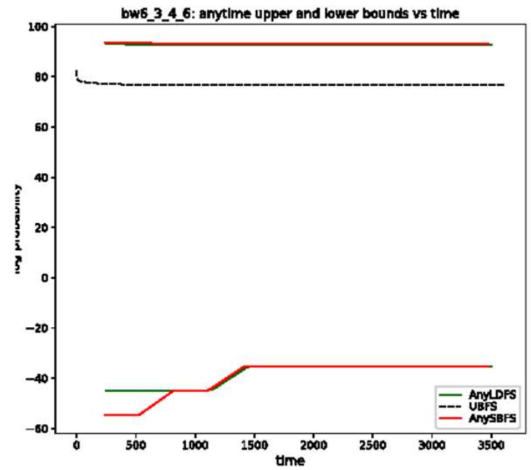


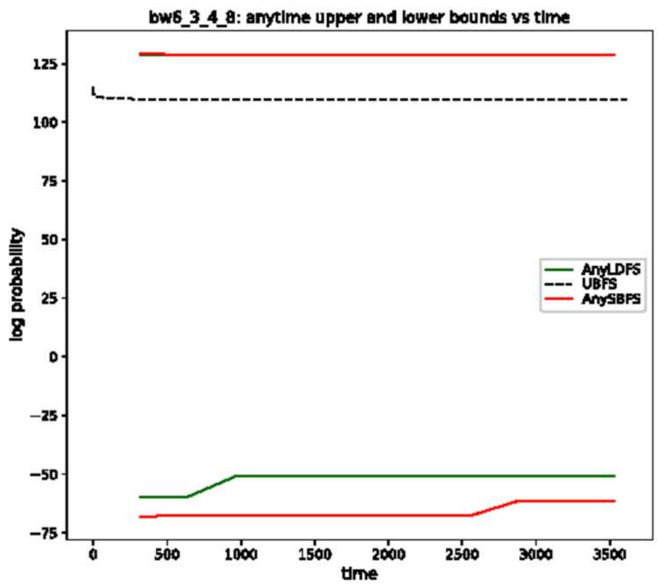
bw2_2_4_7: anytime upper and lower bounds vs time
log probability
time
AnyLDFS
UBFS
AnySBFS

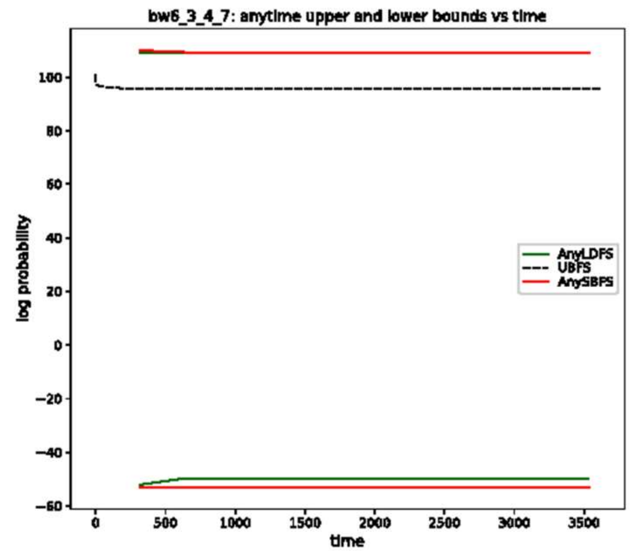
6 blocks


bw6_3_4_4: anytime upper and lower bounds vs time
log probability
time
AnyLDFS
UBFS
AnySBFS

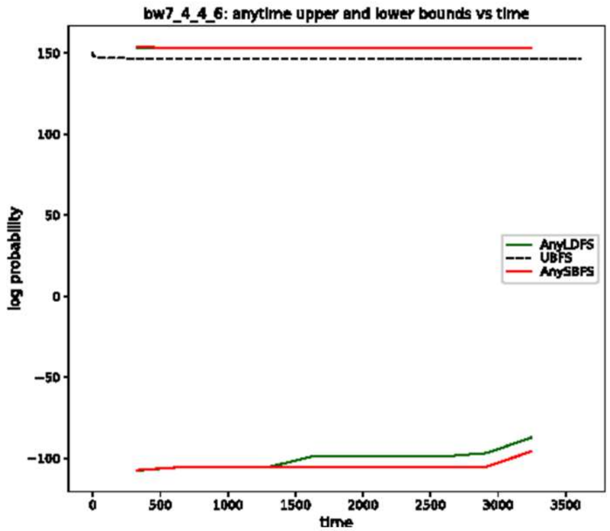

bw6_3_4_5: anytime upper and lower bounds vs time
log probability
time
AnyLDFS
UBFS
AnySBFS

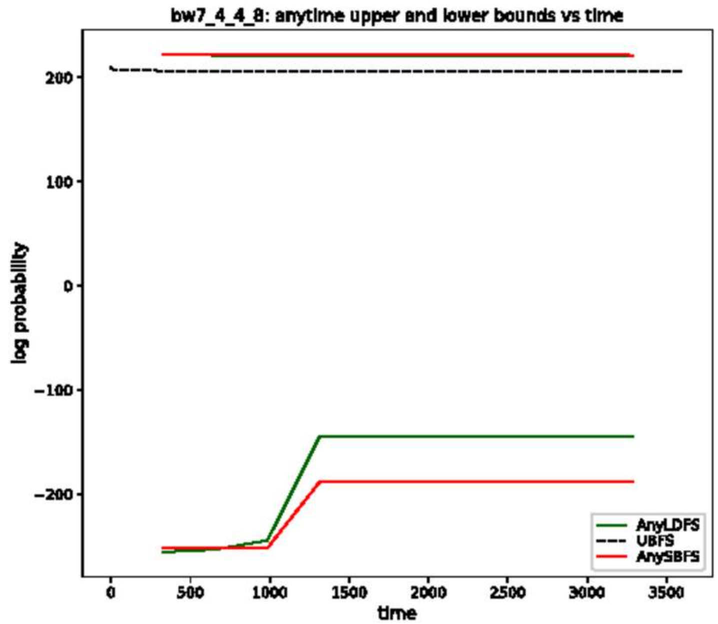

bw6_3_4_6: anytime upper and lower bounds vs time
time
AnyLDFS
UBFS
AnySBFS

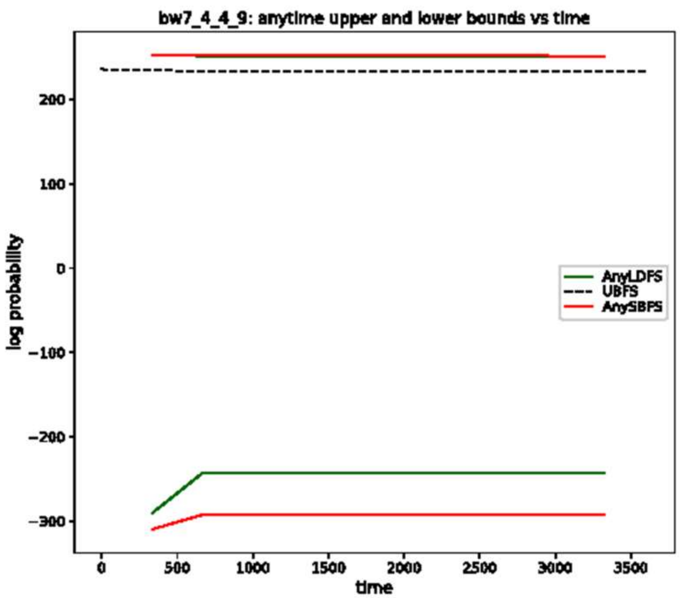

bw6_3_4_8: anytime upper and lower bounds vs time
log probability
time
AnyLDFS
UBFS
AnySBFS

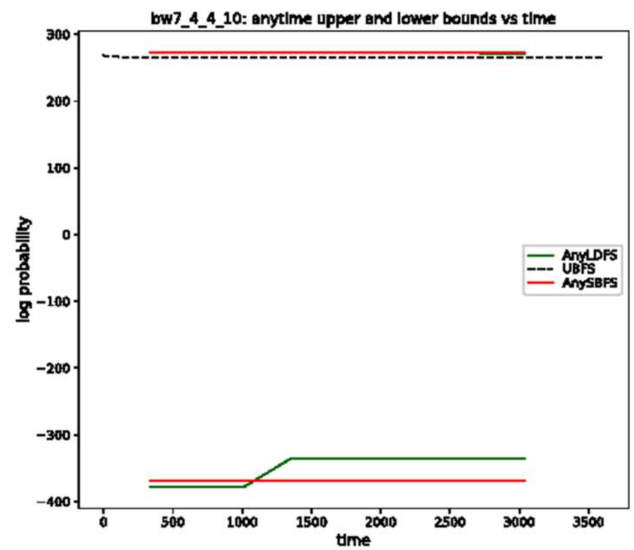

bw6_3_4_7: anytime upper and lower bounds vs time
log probability
time
AnyLDFS
UBFS
AnySBFS

7 blocks


bw7_4_4_6: anytime upper and lower bounds vs time
log probability
time
AnyLDFS
UBFS
AnySBFS


bw7_4_4_8: anytime upper and lower bounds vs time
log probability
time
AnyLDFS
UBFS
AnySBFS


bw7_4_4_9: anytime upper and lower bounds vs time
log probability
time
AnyLDFS
UBFS
AnySBFS


bw7_4_4_10: anytime upper and lower bounds vs time
log probability
time
AnyLDFS
UBFS
AnySBFS