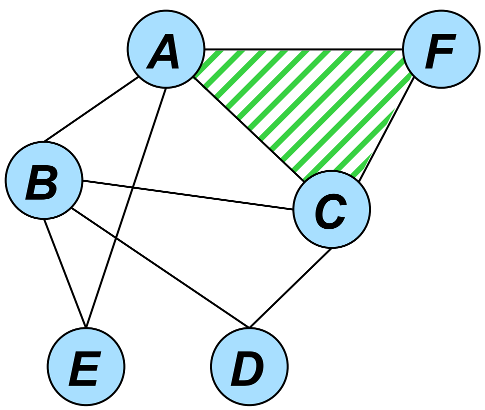
Inference and Search for Discrete Graphical Models;
A tutorial and recent work

Rina Dechter

Bren school of ICS, University of California, Irvine



Sample Applications for Graphical Models

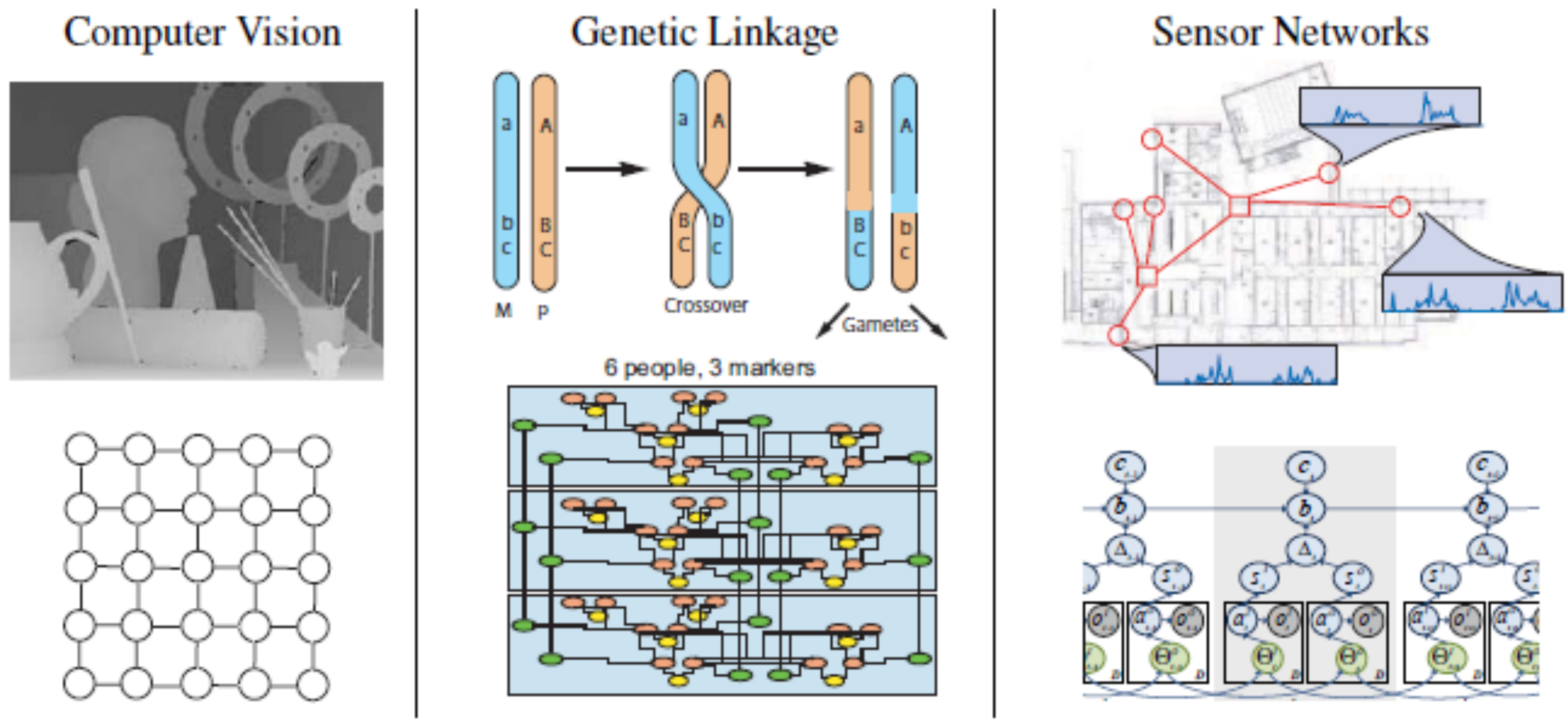
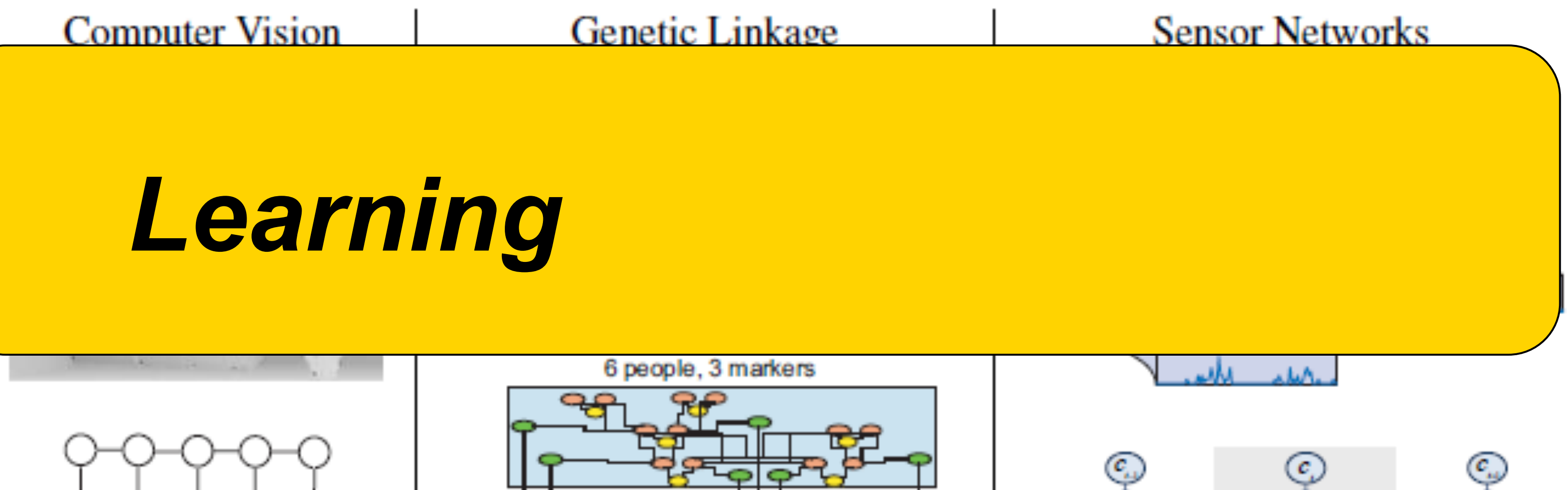


Figure 1: Application areas and graphical models used to represent their respective systems: (a) Finding correspondences between images, including depth estimation from stereo; (b) Genetic linkage analysis and pedigree data; (c) Understanding patterns of behavior in sensor measurements using spatio-temporal models.

Sample Applications for Graphical Models



Learning

Reasoning

Figure 1: Application areas and graphical models used to represent their respective systems: (a) Finding correspondences between images, including depth estimation from stereo; (b) Genetic linkage analysis and pedigree data; (c) Understanding patterns of behavior in sensor measurements using spatio-temporal models.

Outline

- What are graphical models? Queries
- Inference
- Search
- Time vs space, search vs inference
- Bounding inference (Variational: BP, GBP, weighted mini-bucket, cost-shifting)
- Bounding Search (Sampling)
- Anytime algorithms
- Optimization: Tailoring solvers to instance
- Recent algorithmic development
- Summary

Exact algorithms

Approximations Anytime

New work

Graphical Models

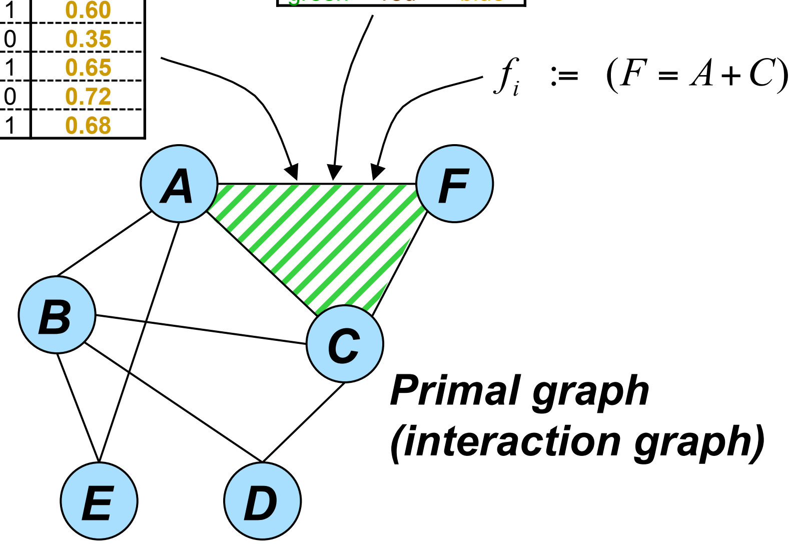
- A graphical model **(X,D,F)**:
  - **X** = $\{X_1,\ldots X_n\}$ variables
  - **D** = $\{D_1, \ldots D_n\}$ domains
  - **F** = $\{f_1,\ldots,f_r\}$ functions
    (constraints, CPTS, CNFs …)
- Operators:
  - combination : Sum, product, join
  - Elimination: projection, sum, max/min
- Tasks:
  - **Belief updating**: $\Sigma_{X-y} \prod_j P_i$
  - **MPE**: $\max_X \prod_j P_j$
  - **CSP**: $\prod_X \times_j C_j$
  - **Max-CSP**: $\min_X \Sigma_j F_j$

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

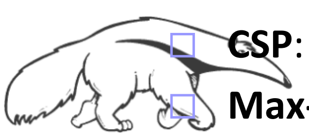
Relation

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



Primal graph (interaction graph)

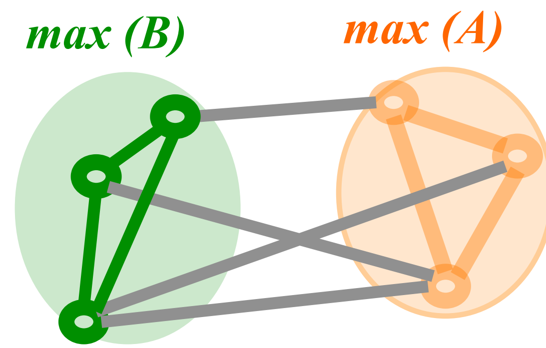
- ***All these tasks are NP-hard***
  - ***exploit problem structure***
  - ***identify special cases***
  - ***approximate***



Queries

- **Optimization Queries: MAP/MPE queries:**

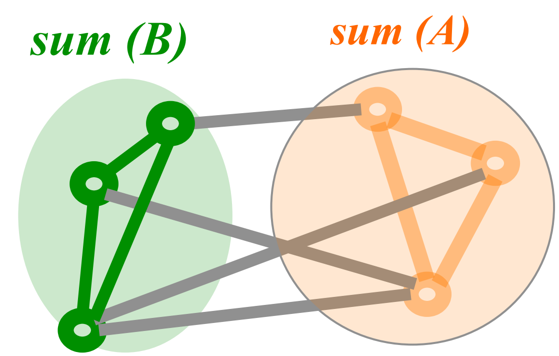
$$x^*_{AB} = \arg\min_{x_A, x_B} \sum_{x_\alpha} \varphi_\alpha \qquad x^*_{AB} = \arg\max_{x_A, x_B} \prod_{x_\alpha} \varphi_\alpha$$



$X = A \cup B$

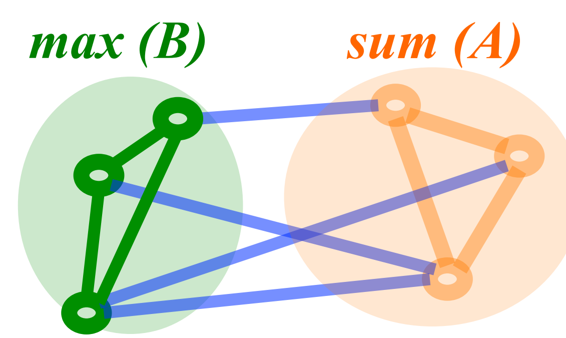
- **Likelihood queries: (counting, partition function, marginal, probability of evidence)**

$$Z = \sum_{x_A, x_B} \prod_{x_\alpha} \varphi_\alpha$$



- **Marginal MAP:**
  - ***Marginalize (sum) away variables A, then find optimal configuration of variables B***

$$\boldsymbol{x}_B^* = \arg\max_{\boldsymbol{x}_B} \sum_{\boldsymbol{x}_A} \prod_{\alpha} \psi(\boldsymbol{x}_\alpha)$$

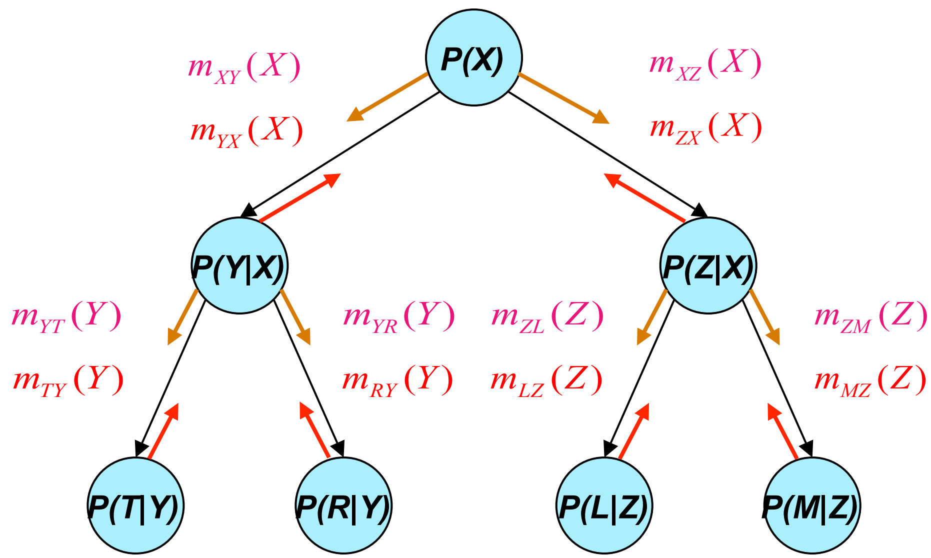


Also satisfiability and expected utility

Tree-solving is Easy

Belief updating (sum-prod)

CSP – consistency (projection-join)



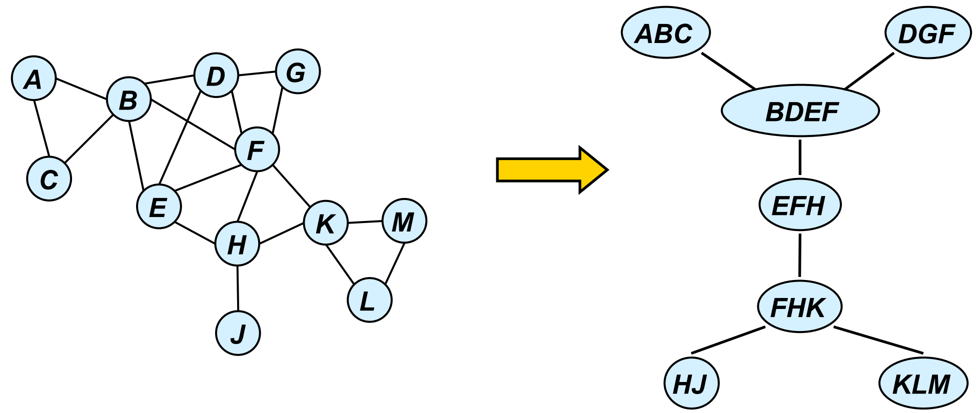
Dynamic Programming, Inference

MPE (max-prod)

#CSP (sum-prod)

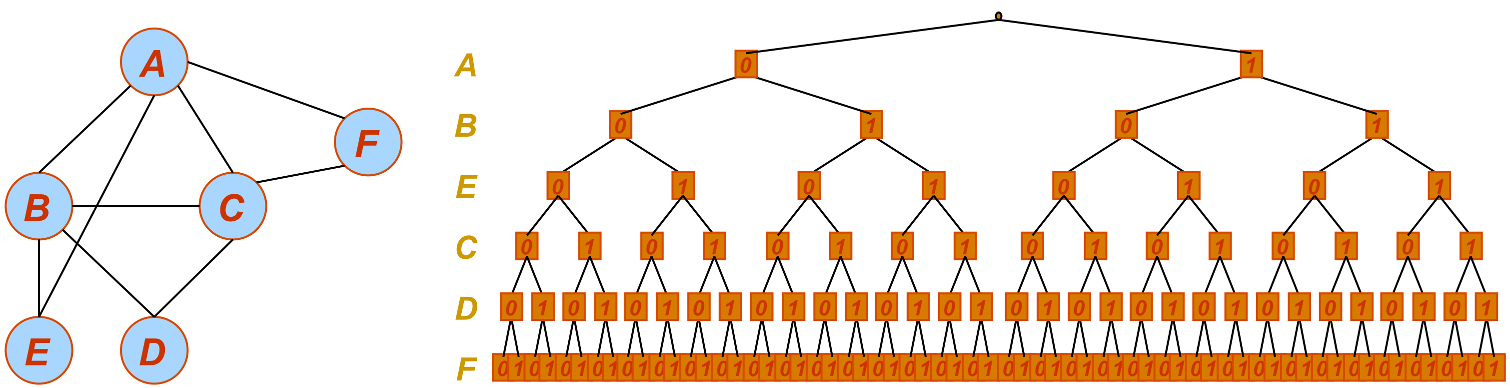
Trees are processed in linear time and memory
Message-passing

Inference vs conditioning-search



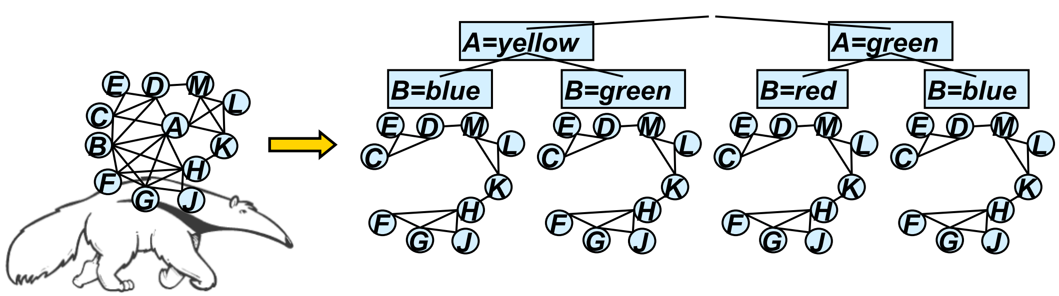
Inference

exp(w) time/space*



Search

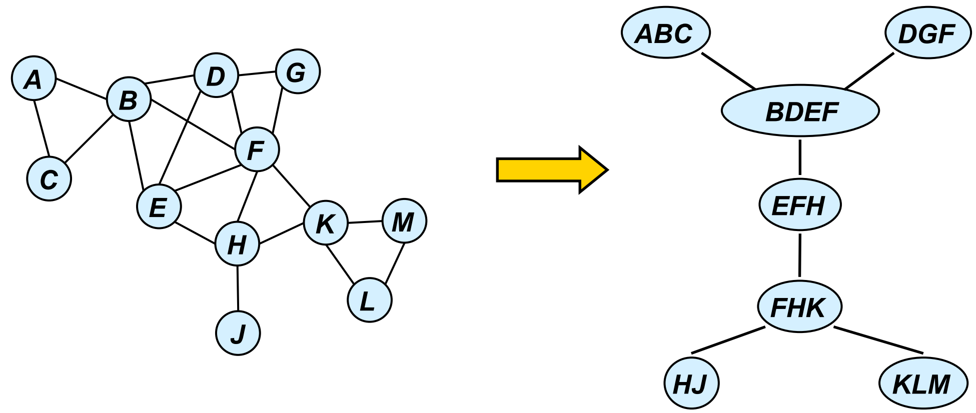
Exp(n) time
O(n) space



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

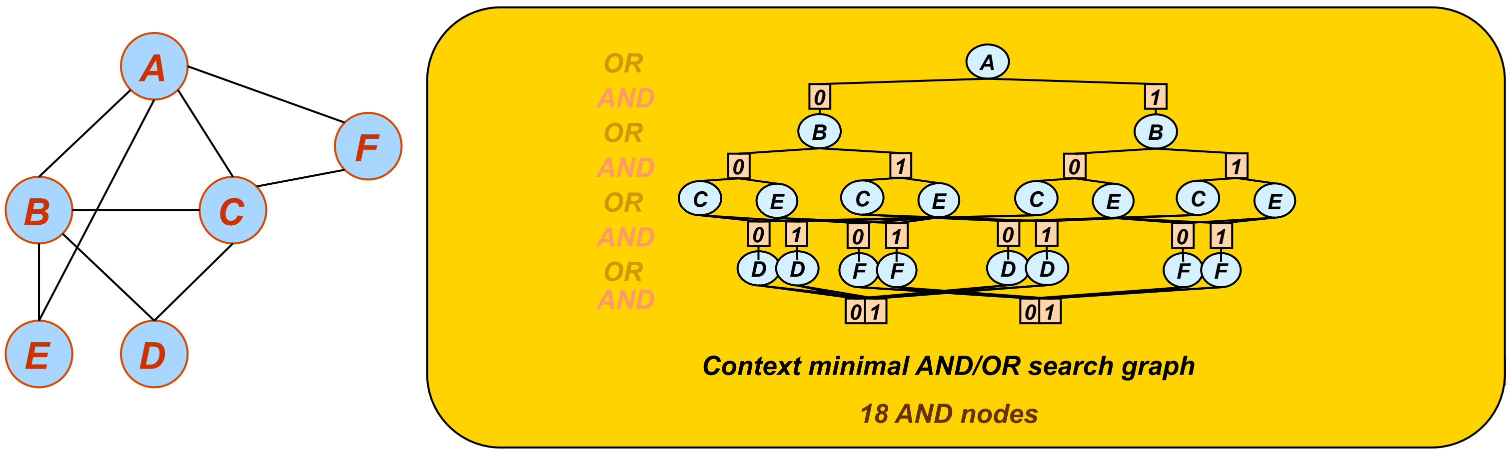
w: user controlled

Inference vs conditioning-search



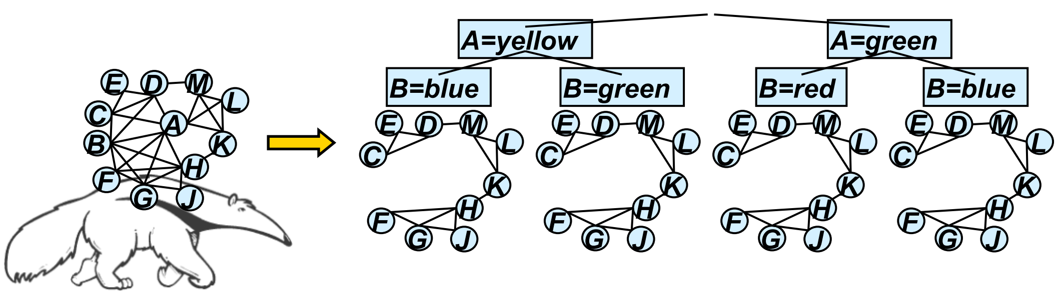
Inference

exp(w) time/space*



Search

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

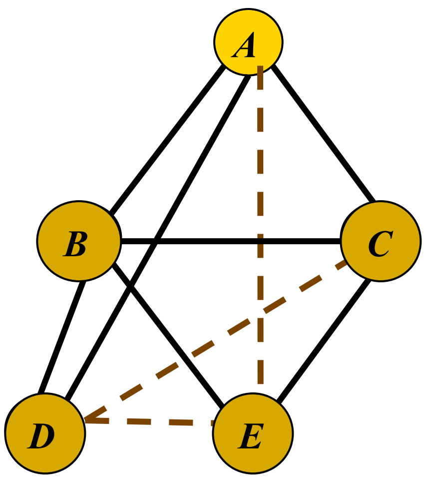


Search+inference:
Space: exp(q)
Time: exp(q+c(q))

q: user controlled

Inference

Query 1: Belief updating: P(X|evidence)=?



“Moral” graph

$$P(a|e=0) \propto P(a,e=0) =$$

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

$$P(a)\sum_{e=0}\sum_{d}\sum_{c} P(c|a) \underbrace{\sum_{b} P(b|a)P(d|b,a)P(e|b,c)}_{h^{B}(a,d,c,e)}$$

Variable Elimination

Query 2: Finding MPE by Bucket Elimination

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

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

$$\underbrace{\qquad\qquad}_{}\;\max_X \prod$$

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

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

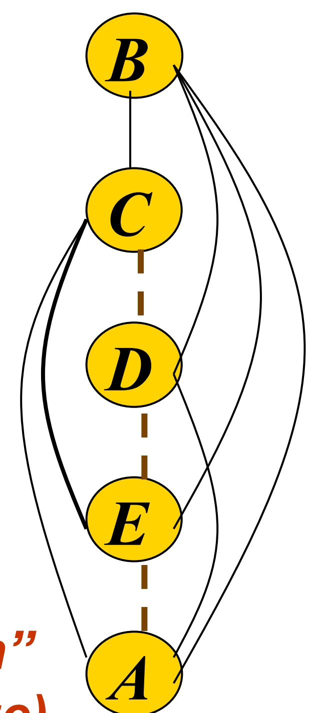
bucket D: $h_{C\to D}(a,d,e)$

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)



Generating the MPE-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. $a' = \arg\max_a P(a) \cdot h^E(a)$

$B{:}\quad P(b|a) \quad P(d|b,a) \quad P(e|b,c)$

$C{:}\quad P(c|a) \quad h^B(a,d,c,e)$

$D{:}\quad h^C(a,d,e)$

$E{:}\quad e=0 \quad h^D(a,e)$

$A{:}\quad P(a) \quad h^E(a)$

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

Generating the MPE-tuple

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

$B: \quad P(b|a) \quad P(d|b,a) \quad P(e|b,c)$

Time and space exponential in the induced-width / treewidth

$$O(n_{k \uparrow w^* + 1})$$

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

$A: \quad P(a) \qquad h^E(a)$

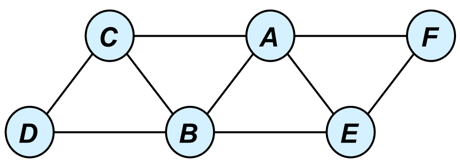
Return (a', b', c', d', e')

Exact Inference solvers at UCI

- BE (Bucket Elimination)
- BEEM BE with External Memory, (UAI 2010)
- IGVO (Iterative Greedy Variable Ordering, AAAI 2011)

Search

OR Search Tree

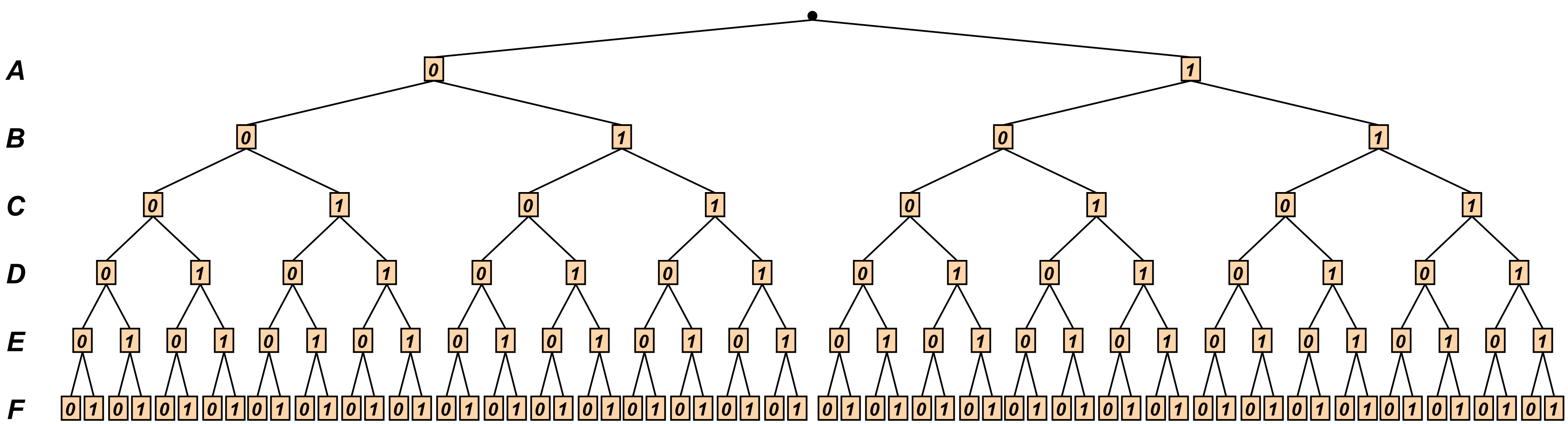


A	B	C	R_{ABC}
0	0	0	**1**
0	0	1	**1**
0	1	0	**0**
0	1	1	**1**
1	0	0	**1**
1	0	1	**1**
1	1	0	**1**
1	1	1	**0**

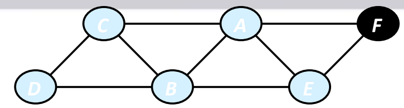
B	C	D	R_{BCD}
0	0	0	**1**
0	0	1	**1**
0	1	0	**1**
0	1	1	**0**
1	0	0	**1**
1	0	1	**0**
1	1	0	**1**
1	1	1	**1**

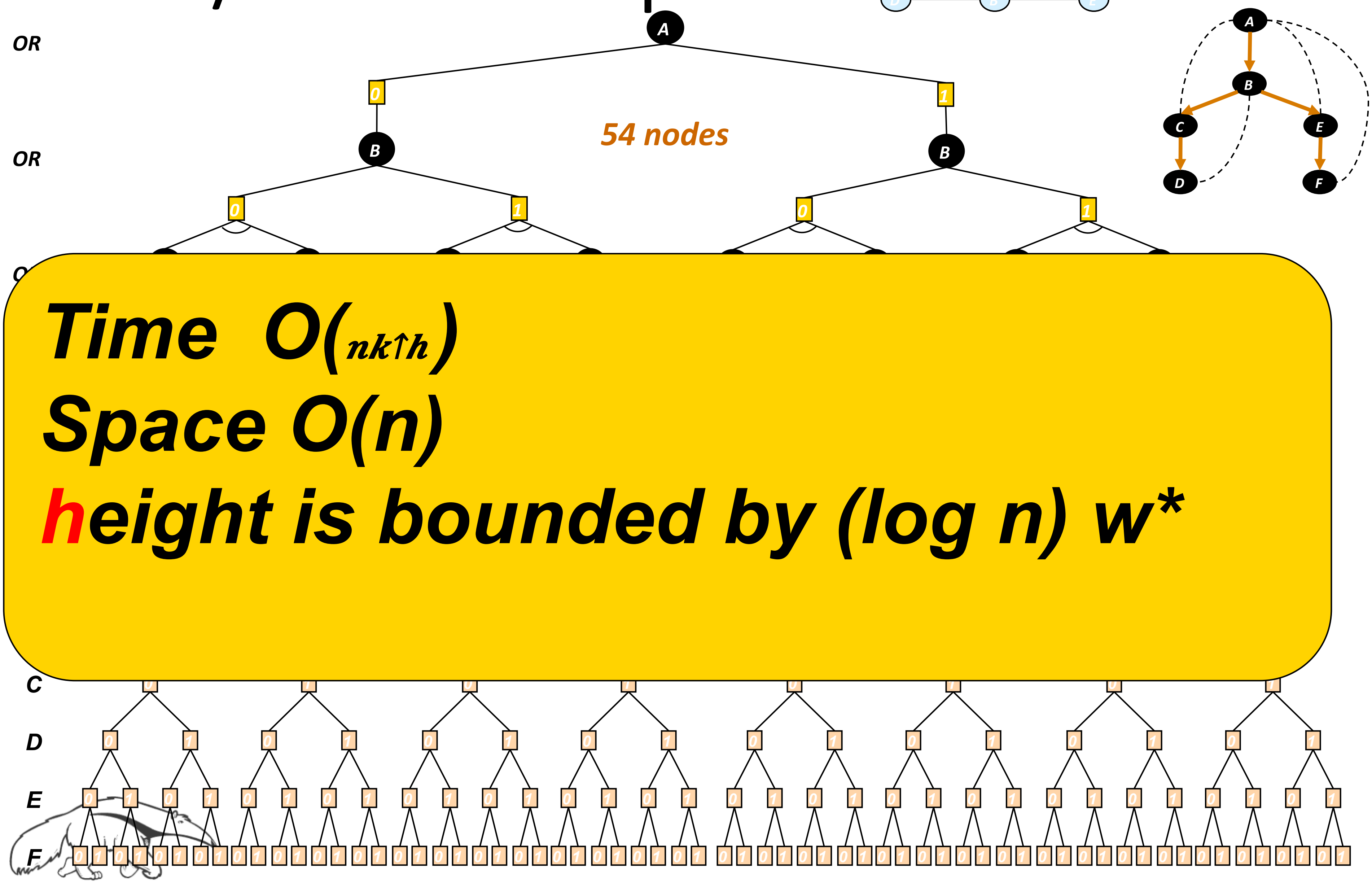
A	B	E	R_{ABE}
0	0	0	**1**
0	0	1	**0**
0	1	0	**1**
0	1	1	**1**
1	0	0	**0**
1	0	1	**1**
1	1	0	**1**
1	1	1	**0**

A	E	F	R_{AEF}
0	0	0	**0**
0	0	1	**1**
0	1	0	**1**
0	1	1	**1**
1	0	0	**1**
1	0	1	**1**
1	1	0	**1**
1	1	1	**0**



AND/OR vs. OR Spaces





Time $O(nk^h)$

Space $O(n)$

height is bounded by (log n) w*

AND/OR Tree DFS Algorithm (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

$P(B \mid A)$

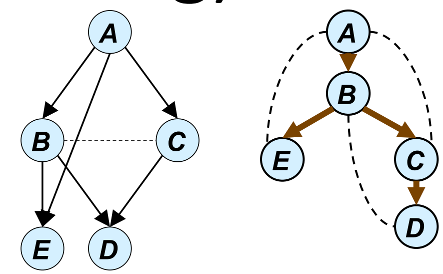
A	B=0	B=1
0	.4	.6
1	.1	.9

$P(C \mid A)$

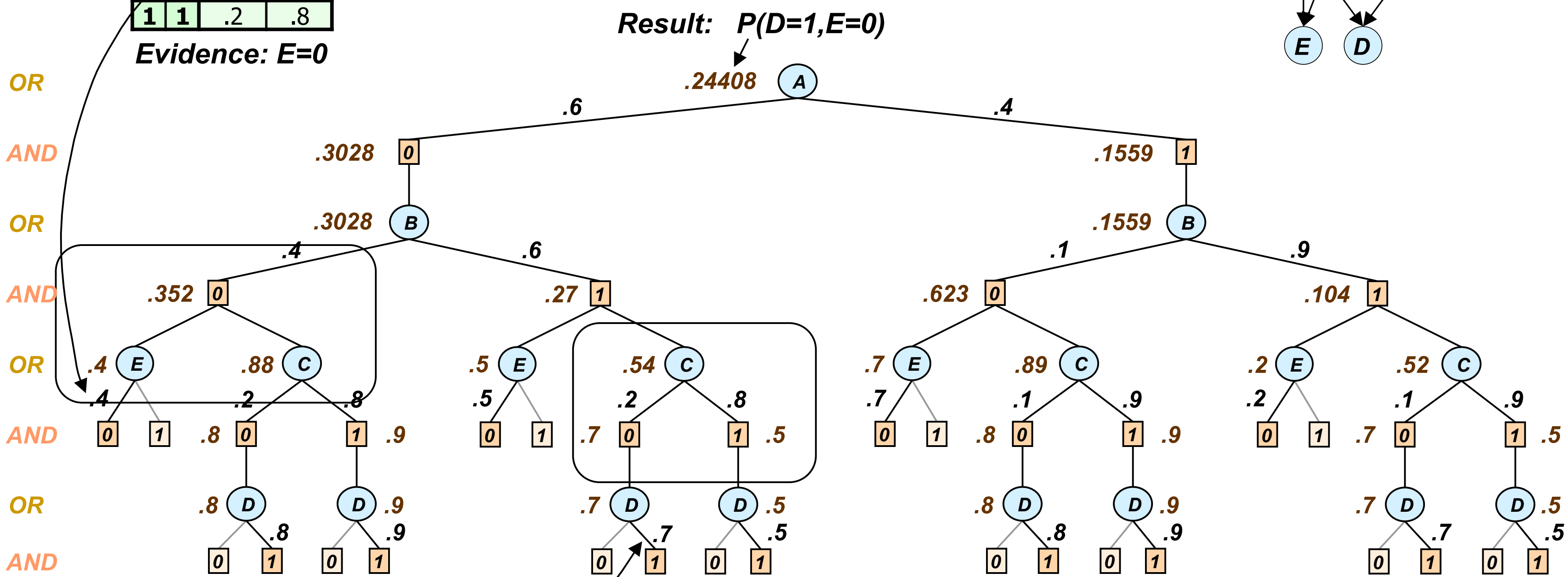
A	C=0	C=1
0	.2	.8
1	.7	.3

$P(A)$

A	P(A)
0	.6
1	.4



Evidence: E=0



$P(D \mid B,C)$

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

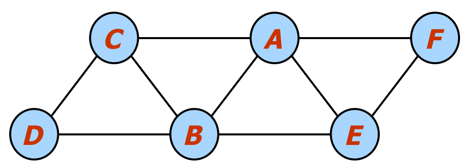
Evidence: D=1

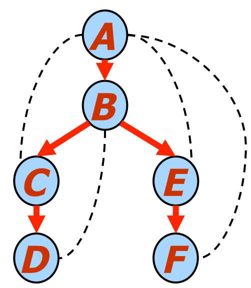
OR node: Marginalization by summation

AND node: product

Value of node = updated belief for sub-problem below

AND/OR Tree Search for Optimization





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**

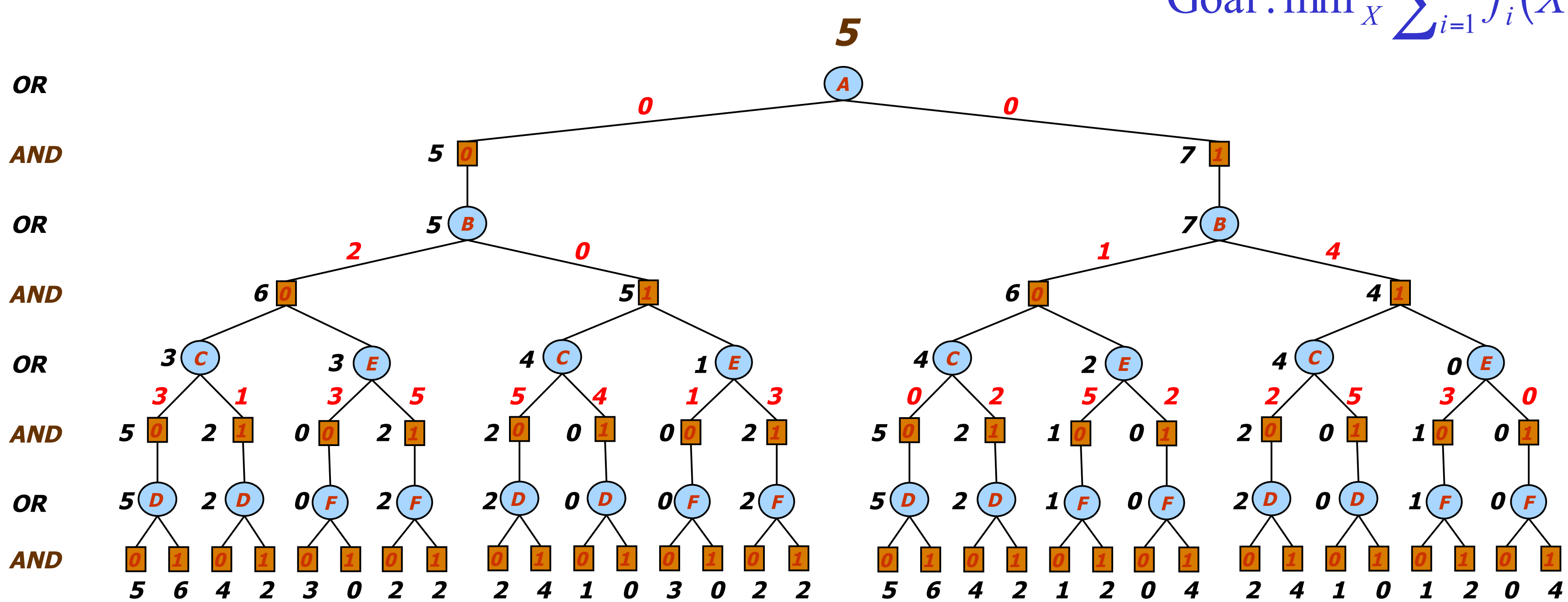
B	D	f_6
0	0	**4**
0	1	**2**
1	0	**1**
1	1	**0**

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**

$$\text{Goal}: \min_X \sum_{i=1}^{9} f_i(X)$$

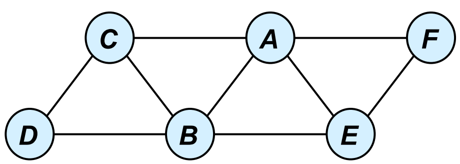


***AND* node = Combination operator (summation)**

***OR* node = Marginalization operator (minimization)**

AND/OR Search Graph

Constraint Satisfaction – Counting Solutions



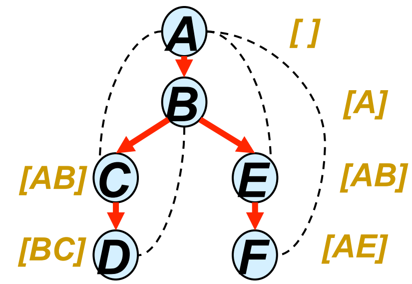
A	B	C	R_{ABC}
0	0	0	**1**
0	0	1	**1**
0	1	0	**0**
0	1	1	**1**
1	0	0	**1**
1	0	1	**1**
1	1	0	**1**
1	1	1	**0**

B	C	D	R_{BCD}
0	0	0	**1**
0	0	1	**1**
0	1	0	**1**
0	1	1	**0**
1	0	0	**1**
1	0	1	**0**
1	1	0	**1**
1	1	1	**1**

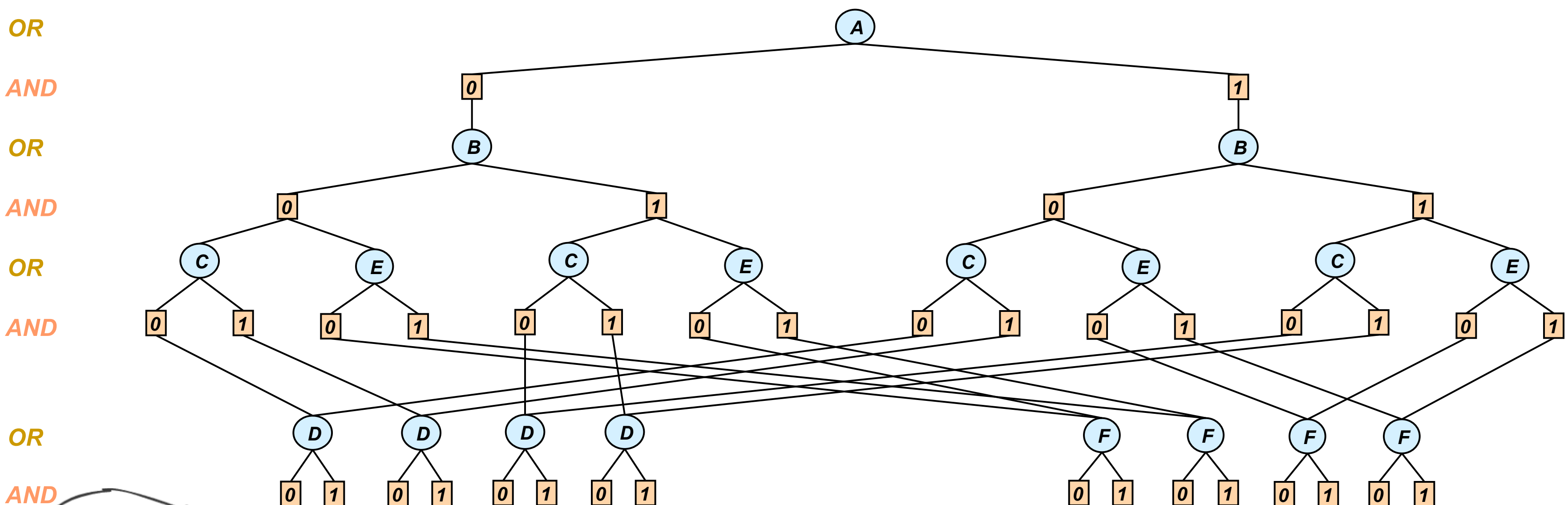
A	B	E	R_{ABE}
0	0	0	**1**
0	0	1	**0**
0	1	0	**1**
0	1	1	**1**
1	0	0	**0**
1	0	1	**1**
1	1	0	**1**
1	1	1	**0**

A	E	F	R_{AEF}
0	0	0	**0**
0	0	1	**1**
0	1	0	**1**
0	1	1	**1**
1	0	0	**1**
1	0	1	**1**
1	1	0	**1**
1	1	1	**0**

Context

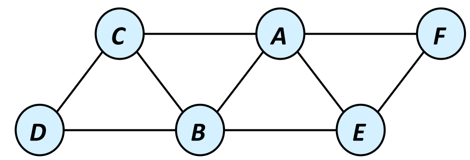


pseudo tree

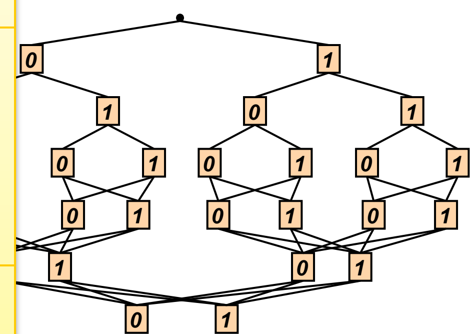


context minimal graph

All Four Search Spaces

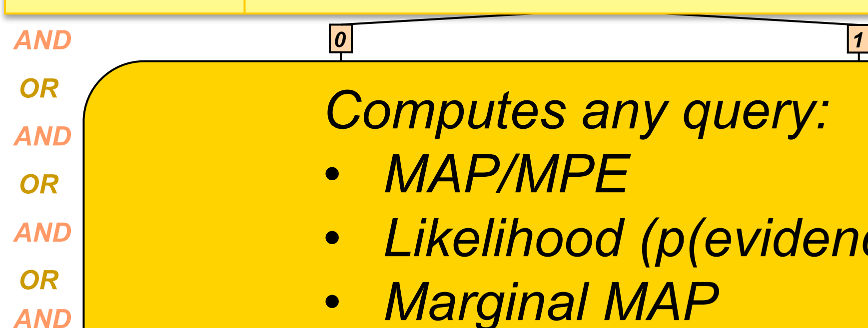


	AND/OR graph	OR graph
Space	$O(n\ k^{w^*})$	$O(n\ k^{pw^*})$
Time	$O(n\ k^{w^*})$	$O(n\ k^{pw^*})$



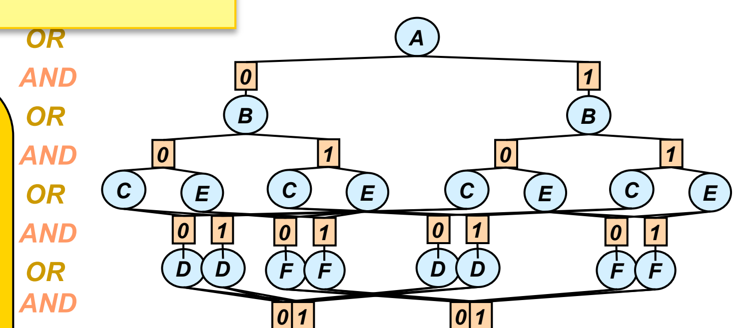
xt minimal OR search graph

28 nodes



Computes any query:

- *MAP/MPE*
- *Likelihood (p(evidence)*
- *Marginal MAP*

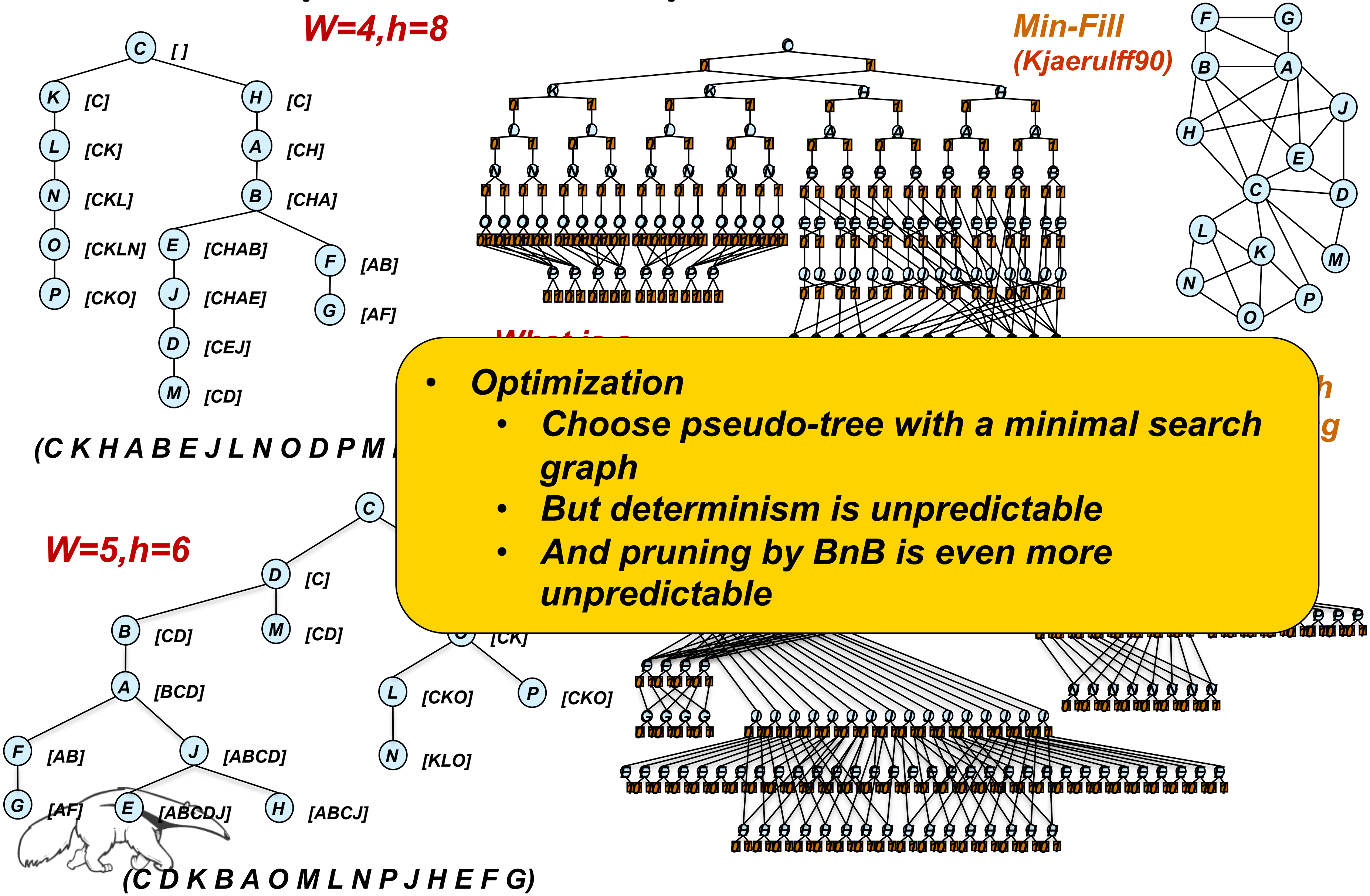


Context minimal AND/OR search graph

18 AND nodes

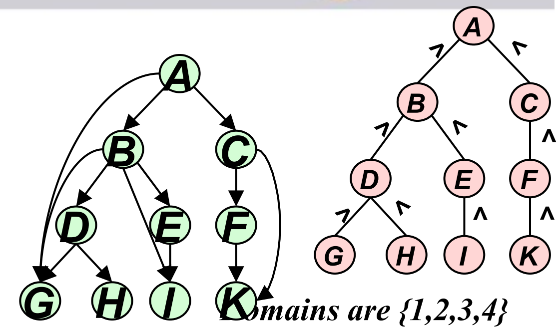
Any query is best computed Over the c-minimal AO space

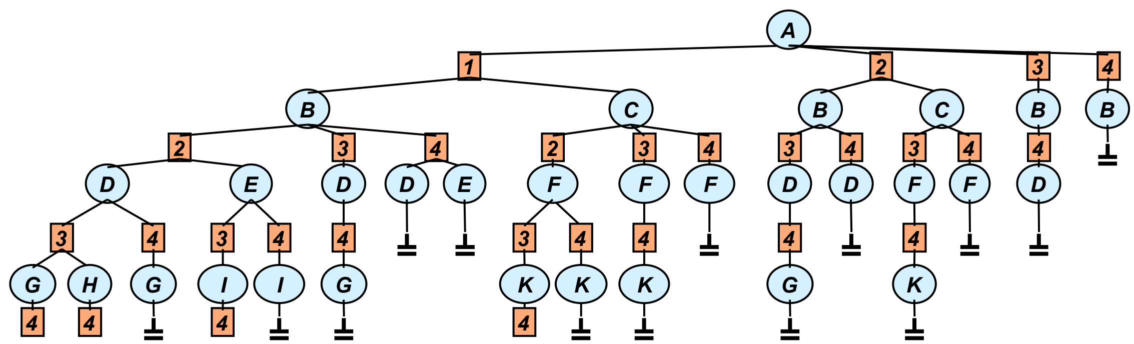
The impact of the pseudo-tree



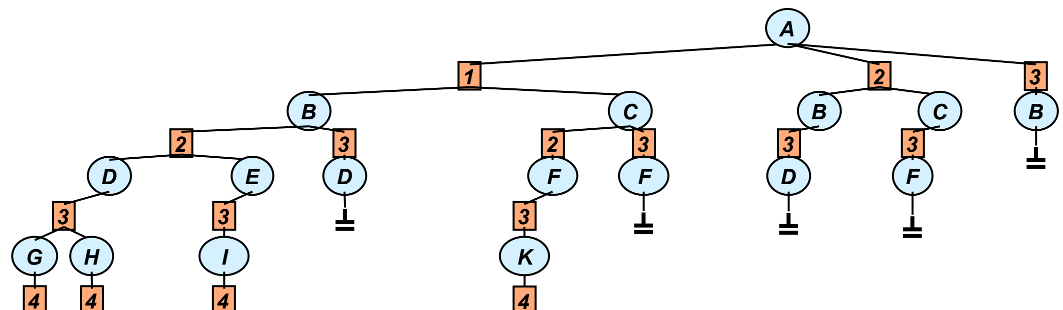
- *Optimization*
  - *Choose pseudo-tree with a minimal search graph*
  - *But determinism is unpredictable*
  - *And pruning by BnB is even more unpredictable*

The Effect of Constraint Propagation

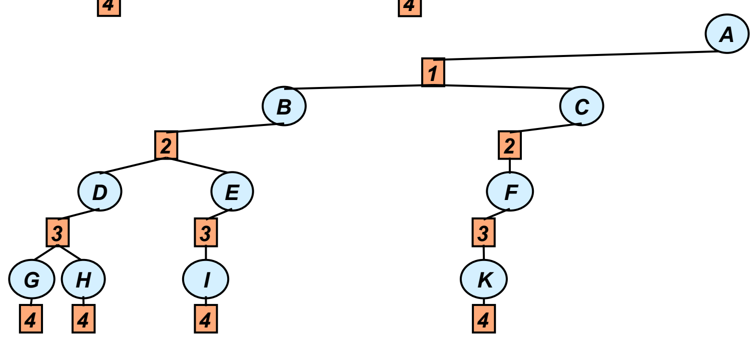




CONSTRAINTS ONLY



FORWARD CHECKING



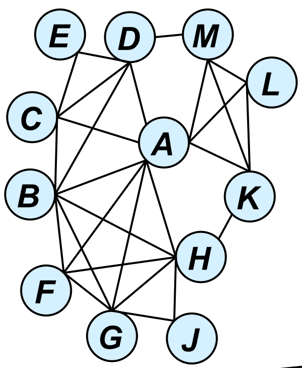
MAINTAINING ARC CONSISTENCY

Search + Inference

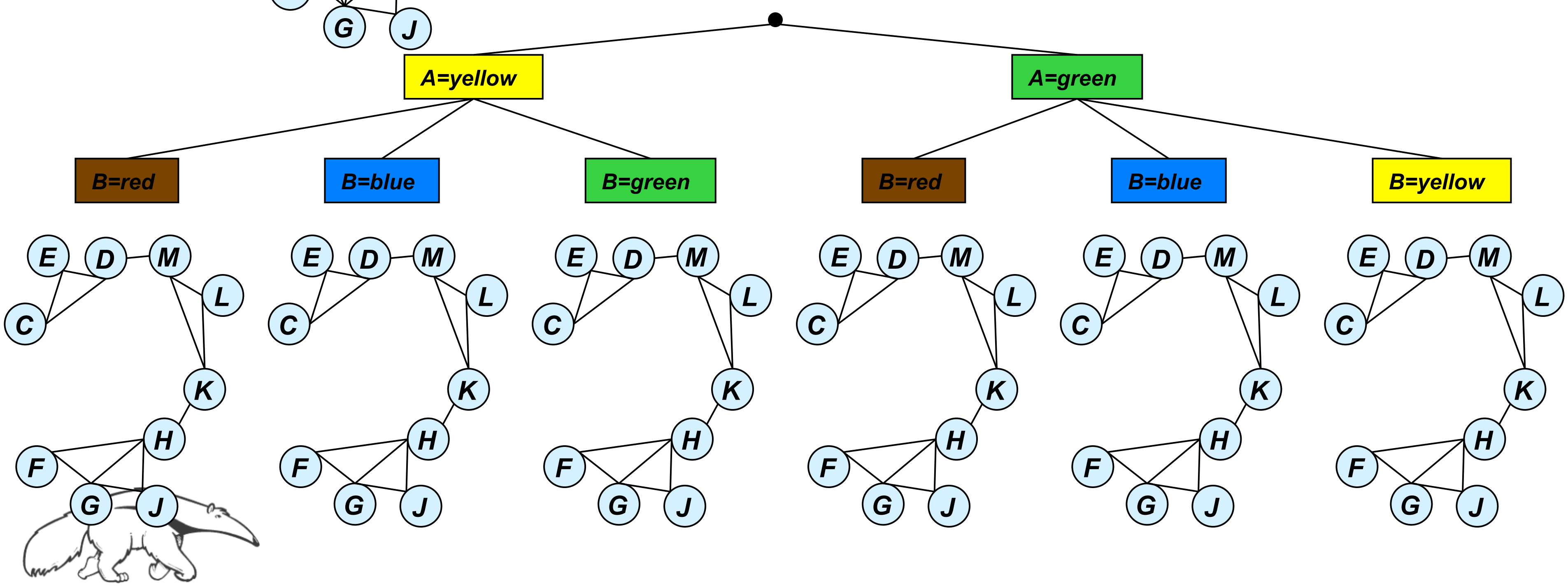
W-Cutset conditioning + inference.

Time exp in cycle-cutset
Memory-linear

Graph Coloring problem



- *Inference may require too much memory*
- *Condition on some of the variables*



Search+Inference : Trading Space for Time

- AO(j): searches depth-first, cache i-context
  - j = the max scope-size of a cache table.

i=0 → *i=j* → *i=w**

Space: $O(n)$
Time: $O(nk^{w*} \log n)$

Space: $O(nk^{j})$
Time: $O(nk^{j}+m_{j})$

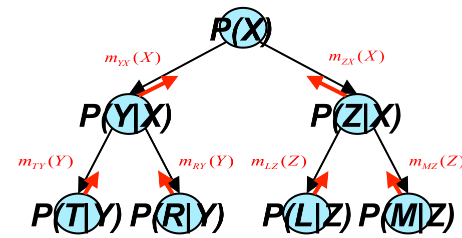
Space: $O(nk^{w*})$
Time: $O(nk^{w*+1})$

Search solvers at UCI

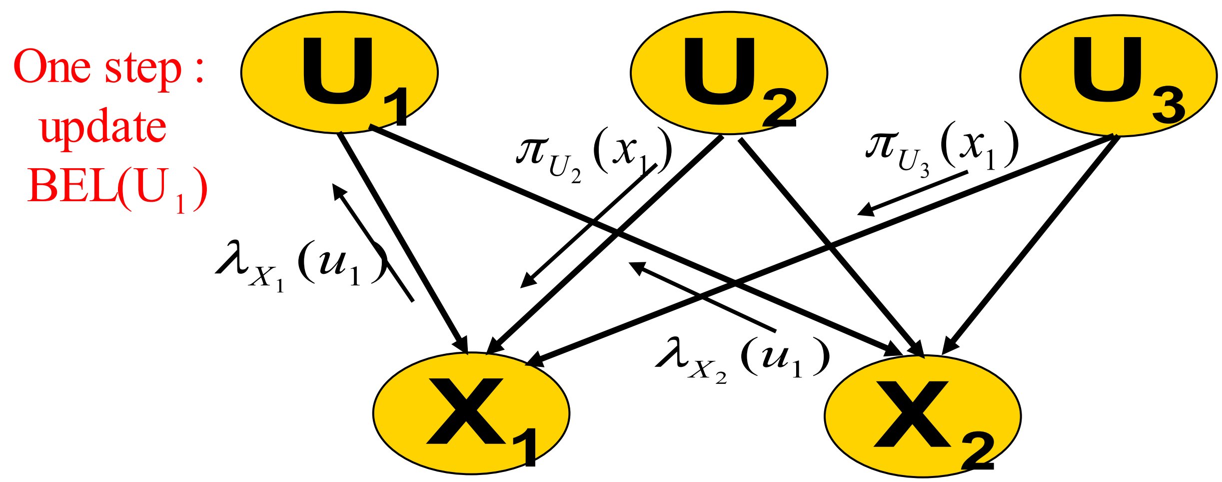
- MAP solvers (AND/OR Branch and Bound):
  - AOBB, AOBF(i-bound using MBE), (Marinescu, 2009)
  - BRAOBB(i-bound, MPLP,JGLP), (Otten 2013)
  - Distributed/parallel AOBB (Otten 2013)
- Likelihood solvers:
  - VEC(i): Variable elimination and conditioning (Gogate, 2009).
  - Aolib (AND/OR search for likelihood, (Mateescu 2007)
- Marginal-MAP (new)
  - AOBB-JG, AOBB-MM: AND/OR search+ weighted mini-bucket and cost-shifting, (Submitted to UAI 2014, Marinescu, Dechter and Ihler)

Approximate Inference

Loopy Belief Proapagation



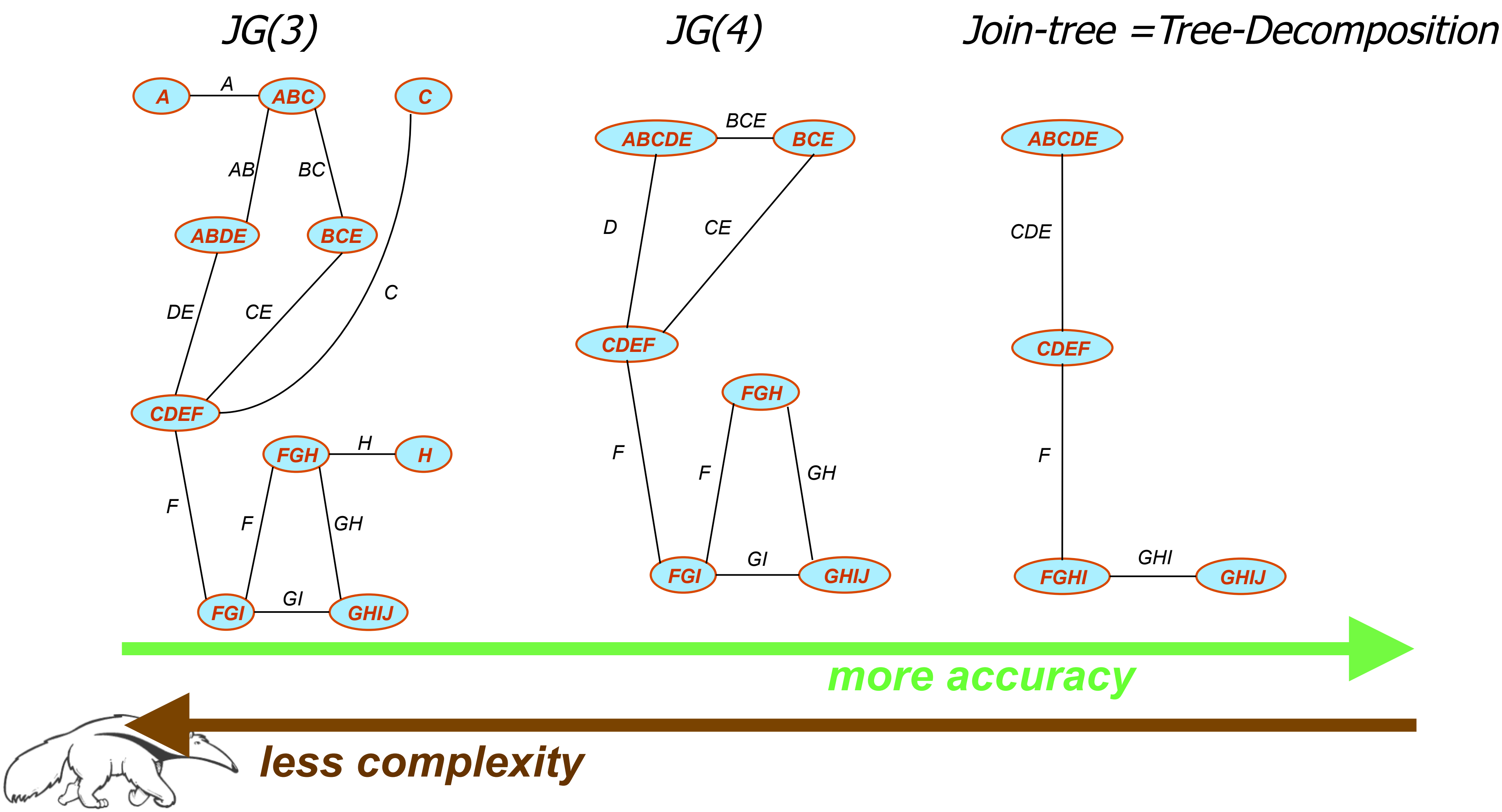
- Belief propagation is exact for poly-trees
- Loopy BP - applying BP iteratively to cyclic networks



- No guarantees for convergence
- Works well for many coding networks

Iterative Join-graphs Propagation (IJGP: kask, (Dechter and Mateescue, 2003), (GBP: yedidya et. Al., 2002….)

i-Bounded join-graph



Mini-Bucket Elimination

Node duplication, renaming

Bucket B $\overbrace{P(E|B,C)}^{\max_B \prod}$ $\overbrace{P(B|A)\ P(D|A,B)}^{\max_B \prod}$

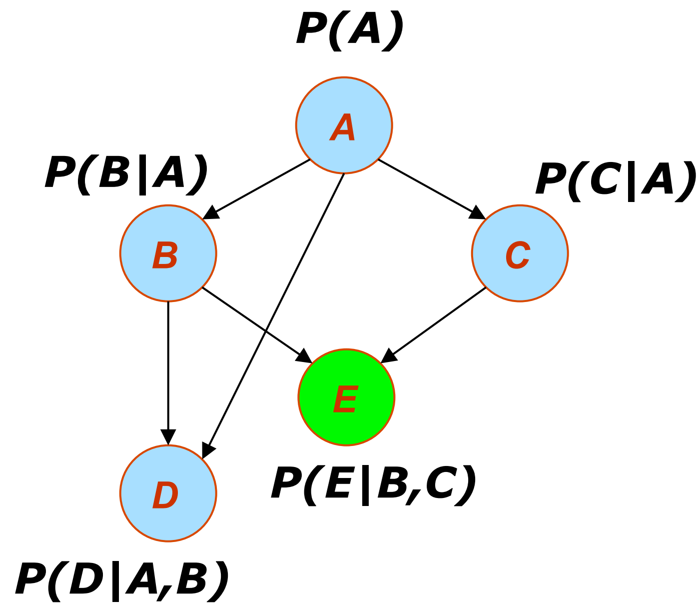
Bucket C $P(C|A)\ h^B(C,E)$

Bucket D $h^B(A,D)$

Bucket E $E = 0\ \ h^C(A,E)$

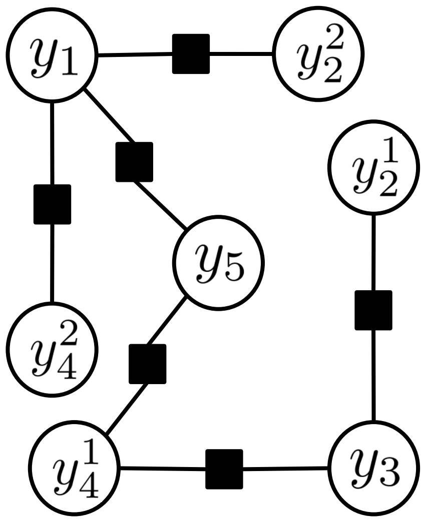
Bucket A $\underbrace{P(A)\ \ h^E(A)\ \ h^D(A)}$

W=2



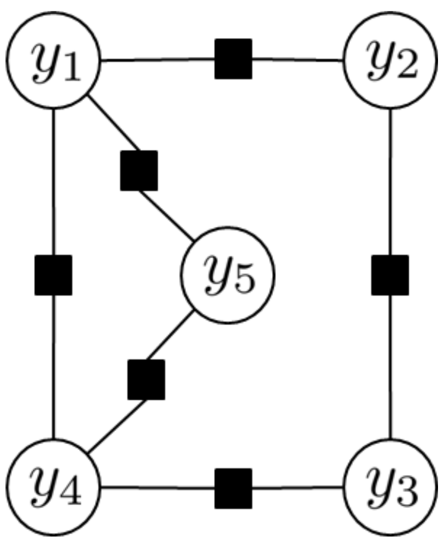
MPE* is an upper bound on MPE --U
Generating a solution yields a lower bound--L

Mini-bucket and mini-clustering



- **Complexity:** *O(r exp(i))* time and *O(exp(i)) space*.
- As ***i*** increases, both accuracy and complexity increase.
- Applicable to all queries.
- Weighted mini-bucket for optimization (

Pairwise Model



Mini-Buckets

$B_2 : \{\psi_{23}\}, \{\psi_{12}\}$

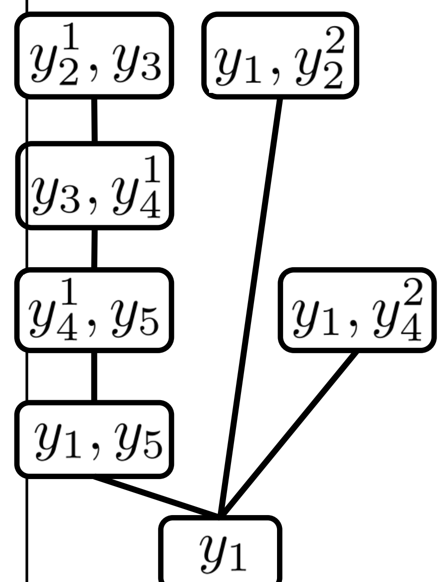
$B_3 : \{\psi_{34}, m_{2\to3}(y_3)\}$

$B_4 : \{\psi_{45}, m_{3\to4}(y_4)\}, \{\psi_{14}\}$

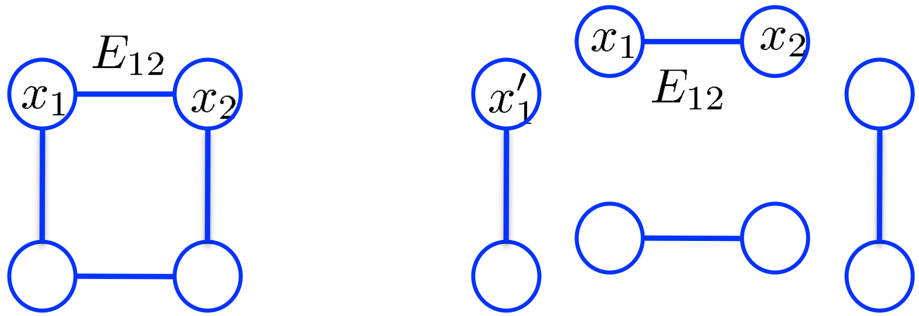
$B_5 : \{\psi_{15}, m_{4\to5}(y_5)\}$

$B_1 : \{m_{2\to1}(y_1), m_{4\to1}(y_1), m_{5\to1}(y_1)\}$

Mini-Cluster Tree



Tightening Bounds via cost-shifting

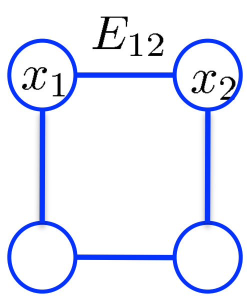


Original ***Decomposition***

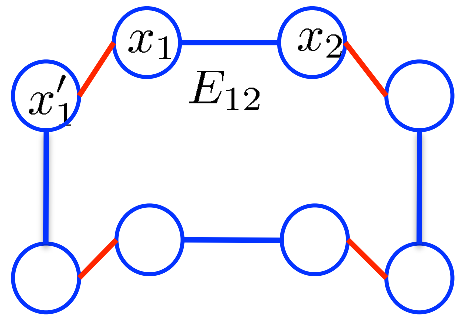
$$\max_{\underline{x}} \sum_{ij} E_{ij}(x_i, x_j) \leq \sum_{ij} \max_{\underline{x}} E_{ij}(x_i, x_j)$$

- Decompose graph into smaller subproblems
- Solve each independently; optimistic bound
- Exact if all copies agree

Decomposition view



Original



Decomposition

$$\forall i \sum_j \lambda_{ij}(x_i) = 0$$

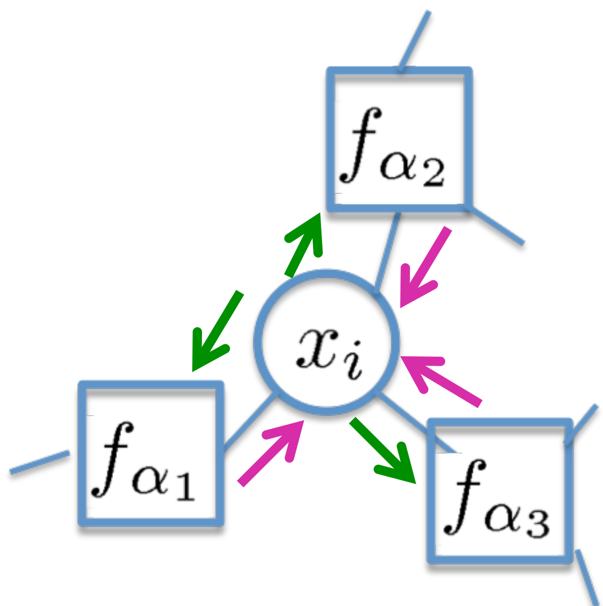
$$\max_{\underline{x}} \sum_{ij} E_{ij}(x_i, x_j) \le \min_{\lambda} \sum_{ij} \max_{\underline{x}} E_{ij}(x_i, x_j) + \lambda_{ij}(x_i) + \lambda_{ji}(x_j)$$

- Decompose graph into smaller subproblems
- Solve each independently; optimistic bound
- Exact if all copies agree
- Enforce lost equality constraints via Lagrange multipliers

Factor graph Linear Programming

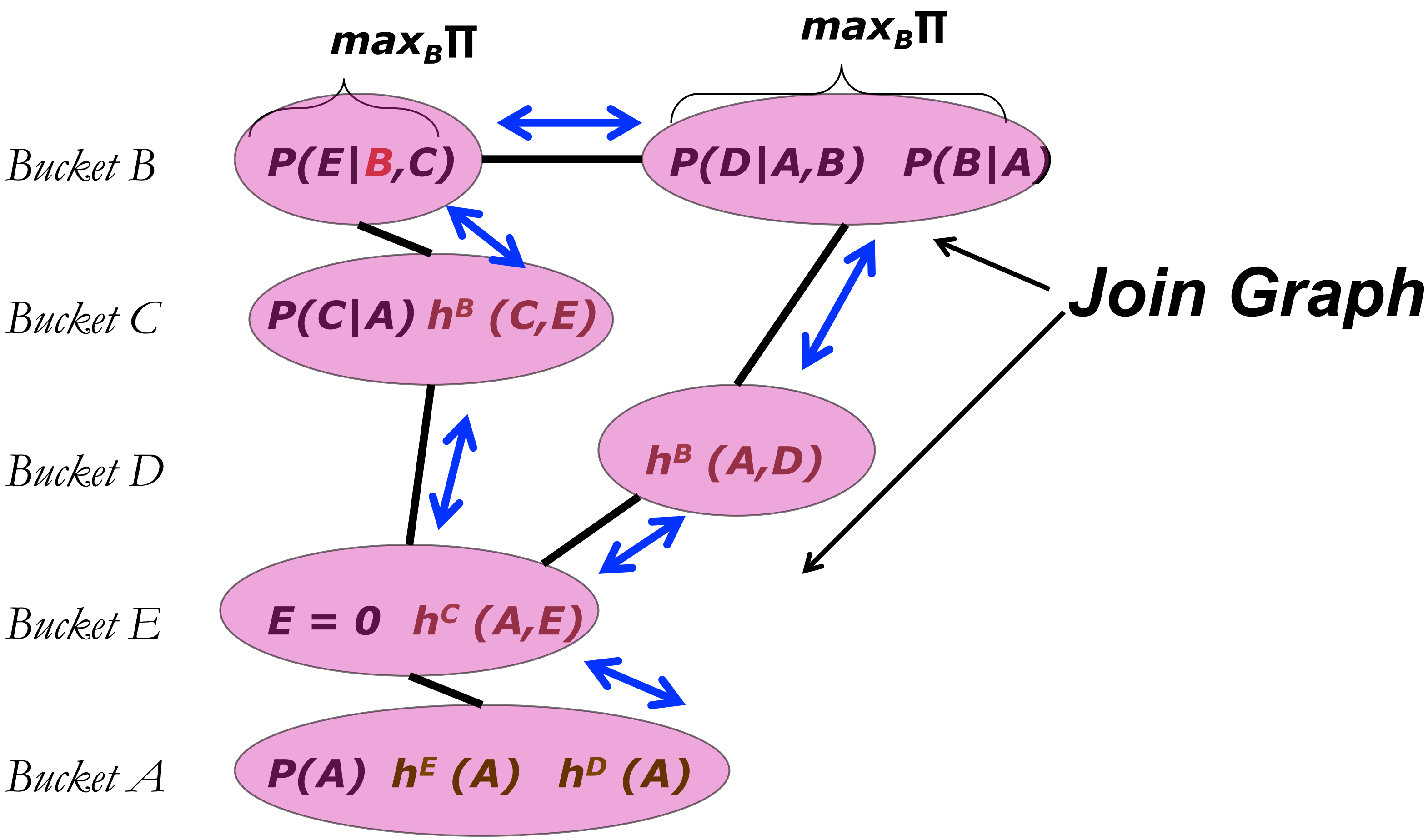
- Update the original factors (FGLP)
  - Tighten all factors over over x_i simultaneously
  - Compute max-marginals $\forall \alpha,\ \gamma_\alpha(x_i) = \max_{x_\alpha \setminus x_i} f_\alpha$
  - & update:

$$\forall \alpha,\ f_\alpha(x_\alpha) \leftarrow f_\alpha(x_\alpha) - \gamma_\alpha(x_i) + \frac{1}{|F_i|} \sum_\beta \gamma_\beta(x_i)$$



Join-graph based cost-shifting

(Ihler, Flerova, Dechter, Otten, UAI 2012)



Bounded Inference solvers at UCI

- BP, IJGP(i-bound): (Mateescue et. Al, 2002, 2010, Gogate)
- Bounding schemes:
  - MB(i-bound),
  - weighted-MB(i) (Ihler, 2012)
  - FGLP (Ihler, 2012)
  - JGLP(i-bound) (Ihler 2012)
- Bounding schemes provide heuristic for AND/OR search

Approximate Search: Sampling stochastic local search

Sampling and Local Search at UCI

- Likelihood queries:
  - W-cutset sampling (Gibbs and importance)
  - SampleSearch (Importance sampling)
  - AND/OR sampling (Importance sampling)
  - Hybrid of all (Importance sampling)
- MAP/MPE
  - GLS+ (Hutter et. al., 2005)
  - STLS (Stochastic tree local search, Milchgrub and Dechter, submitted)

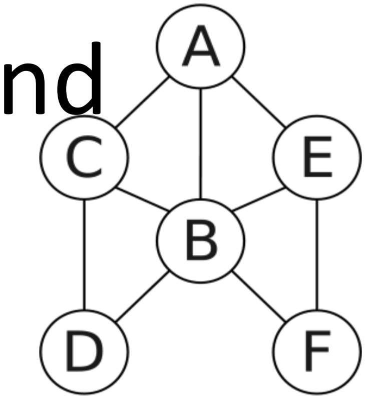
Outline

- What are graphical models? Queries
- Inference
- Search; via AND/OR search
- Time vs space, search vs inference
- Bounding inference (BP, GBP, mini-bucket Variational)
- Bounding Search (Sampling)
- Anytime search algorithms for MAP
- Optimization: Tayloring solver to problem
- UCI Algorithm Library
- Conclusions

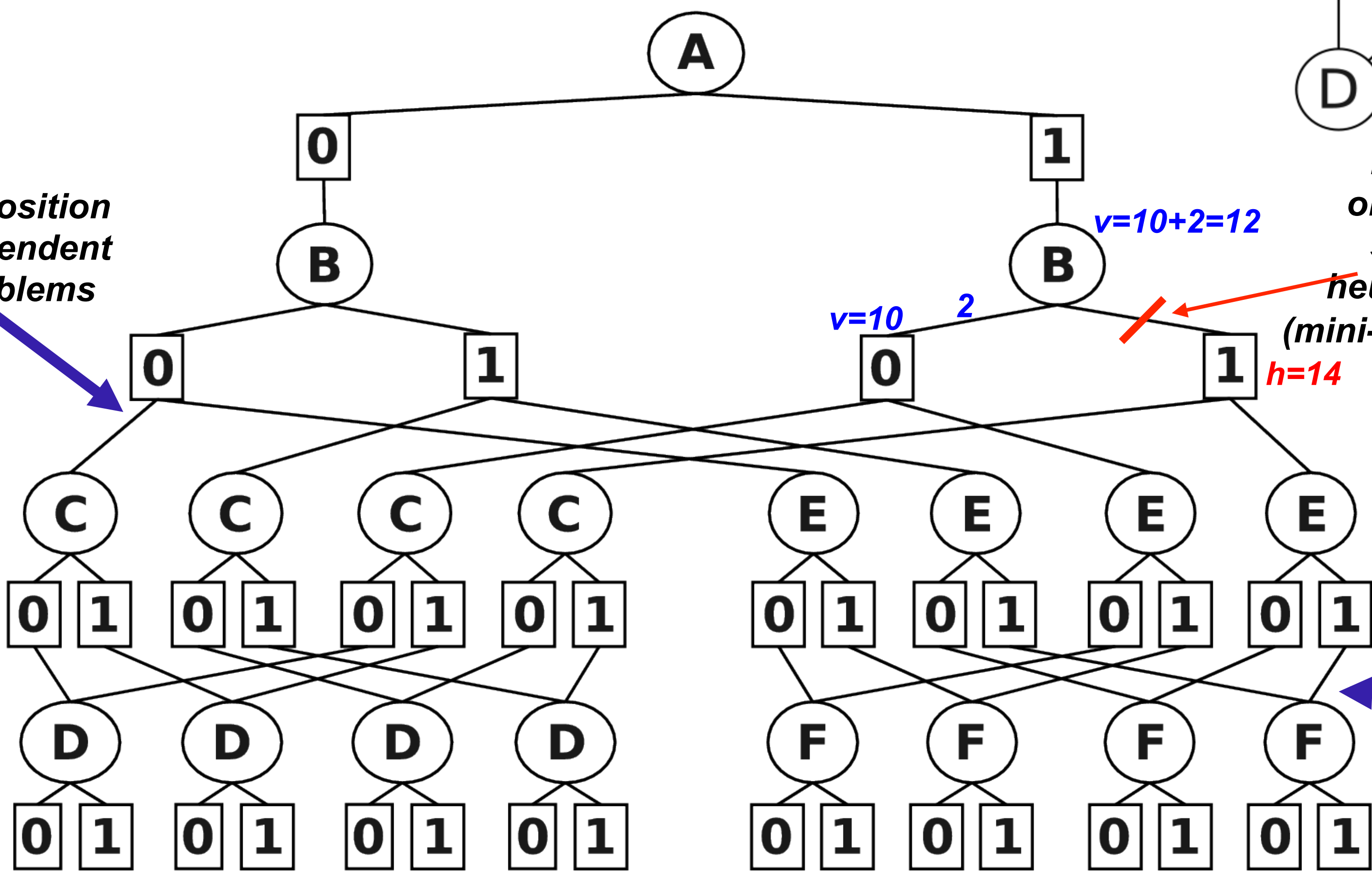
Exact algorithms

Approximations

MAP by AND/OR Branch-and-Bound



Decomposition of independent subproblems



Prune based on current best solution and heuristic estimate (mini-bucket heuristic).

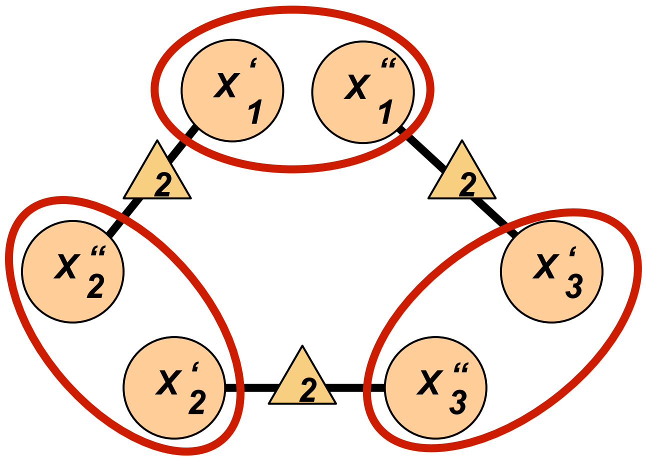
Cache table for F (independent of A)

B	E	cost
0	0	10
0	1	6
1	0	...
1	1	...

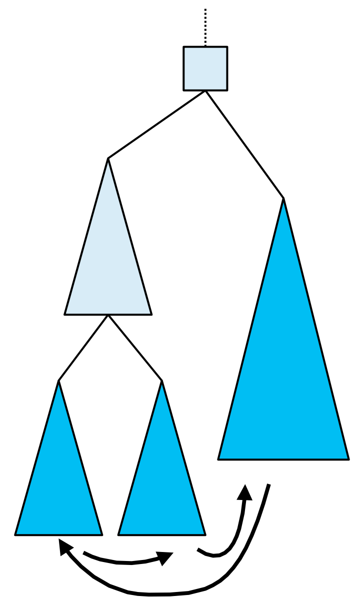
[Marinescu & Dechter, AIJ'09]

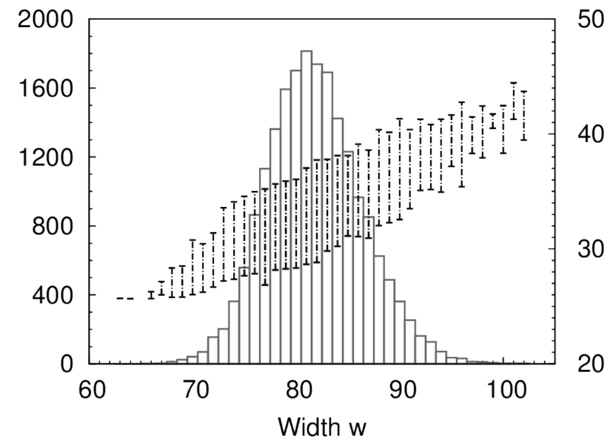
AOBB + Central Enhancements

+ *SLS* *(Hutter et. Al 2005)*



$$\min_{\lambda} \sum_{(ij)} \max_{X}(f_{ij}(X_i, X_j)+ \lambda_{ij}(X_i), \lambda_{ji}(X_j))$$





Cost-shifting (MPLP) Re-parametrization

Tighter bounds by iteratively solving linear programming relaxations and message passing on join graph.

Breadth-First Subproblem Rotation

Improved anytime performance through interleaved processing of independent subproblems.

Enhanced Variable Ordering Schemes

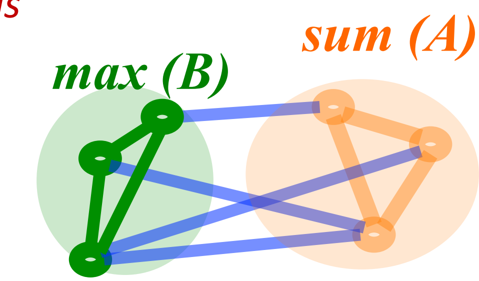
Highly efficient, stochastic minfill / mindegree implementations for lower-width orderings.

(Ihler ,Flerova, Dechter, Otten, 2012, Otten and Dechter 2011, Kask, Gelfand, Otten, Dechter 2010)

This year's advancements

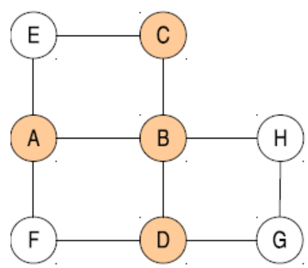
A New Algorithm for Marginal MAP

- *(Submitted to UAI-2014) Improving Marginal Map for Graphical Models"*
- Radu Marinescu, Rina Dechter, Alex Ihler.



- Problem: $$\boldsymbol{x}_B^* = \arg\max_{\boldsymbol{x}_B} \sum_{\boldsymbol{x}_A} \prod_{\alpha} \psi(\boldsymbol{x}_\alpha)$$

Marginalize away variables A, then and Find optimal configuration of variables B



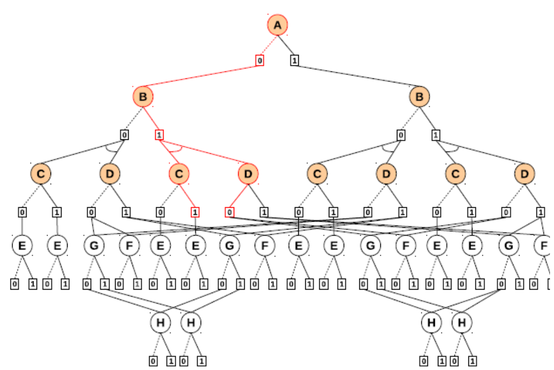


Figure 2: AND/OR search spaces for marginal MAI

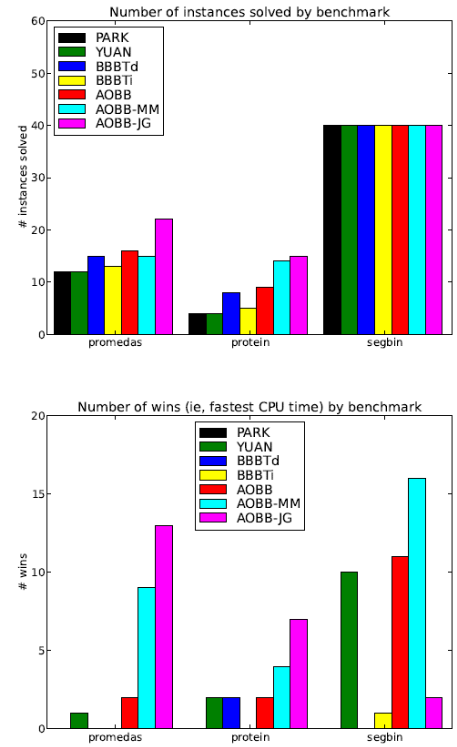


Figure 5: Number of instances solved (top) and number of wins (bottom) by benchmark.

Improving Marginal Map for Graphical Heuristics generated by weighted mini-bucket and moment-matching heuristics.

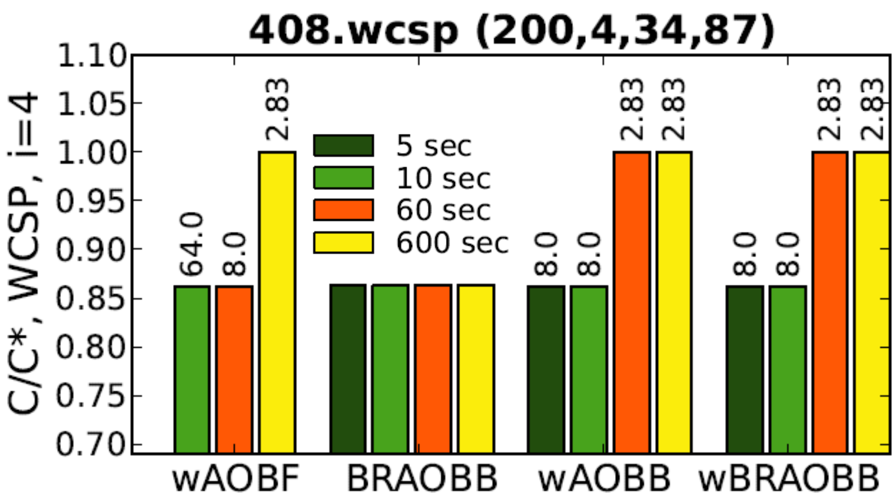
- ***Branch and Bound Search of AND/OR search***

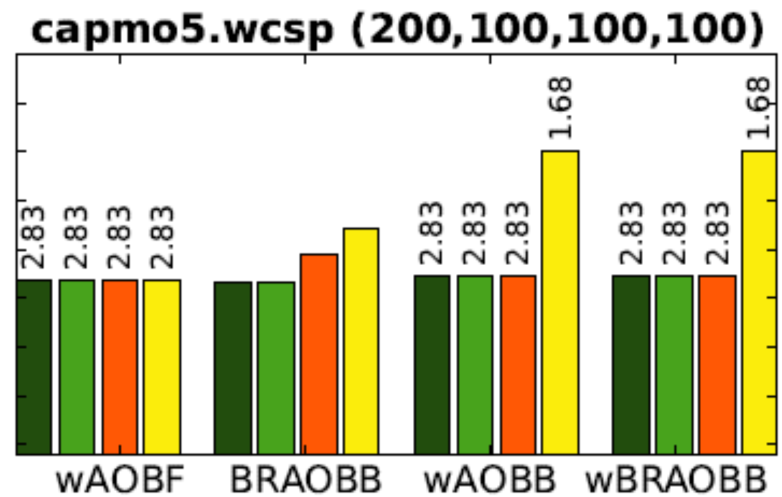
Weighted AND/OR Search

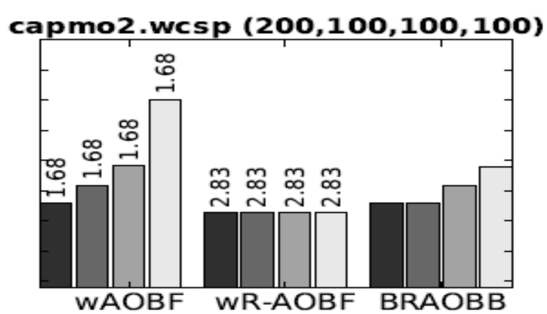
Paper submitted to ECAI-2014: "EvaluatingWeighted DFS Branch and Bound over Graphical Models"

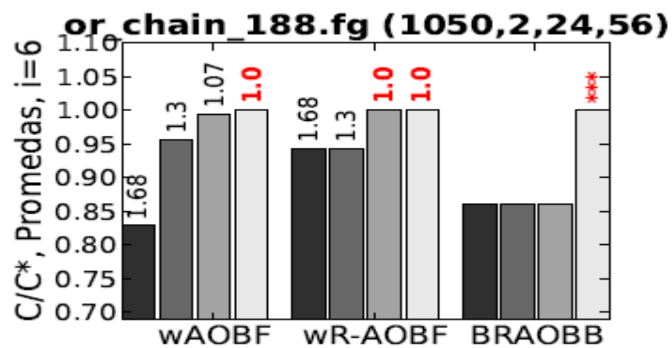
Natalia Flerova, Radu Marinescu, Rina Dechter

•*Empirically evaluation proposed algorithms wAOBB and wBRAOBB against Weighted Best-First search (wAOBB) and Breadth-First AND/OR Branch and Bound (BRAOBB)*









A New Tree-based Stochastic Local search for MAP

STLS: Cutest-driven Local Search for MPE" Alon Milchgrub and Rina Dechter, submitted UAI-2014

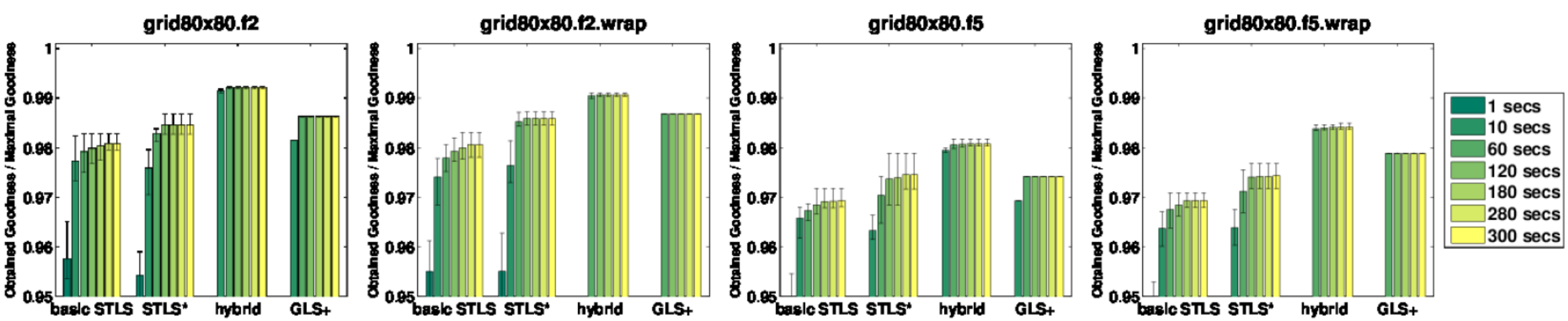
Problem:

$$\bar{x}^* = argmin_{\bar{x}} \sum_i \varphi_i(x_i) + \sum_{i<j\in E} \psi_{i,j}(x_i, x_j)$$

Algorithm STLS:
Input: an instance of the energy minimization problem
Output: an assigment to the problem approximating its minimum

1. Initialize the variables according to the current initialization scheme (see 3.2)
2. Find the set of variables T which are not a part of any cycle using the algorithm described in [15].
3. Repeat until stagnation is declared, once a set number of iterations in which no variable has changed its value have passed (or time bound has passed):
   (a) Find a cycle-cutset C for $V \backslash T$.
   (b) Update the values of the variables:
   - Given the values of the cycle-cutset C inducing a forest $F = V \backslash C$, use BP to find the optimal assignment on F.
   - Given the values of the tree variables T, use some local search algorithm (i.e. HOPFIELD MODEL, GLS^+[6], etc.) in order to update the values C (optional implementation).

   If this last feature is utilized the two update stages are performed alternately until convergence or until the set number of iterations has passed.

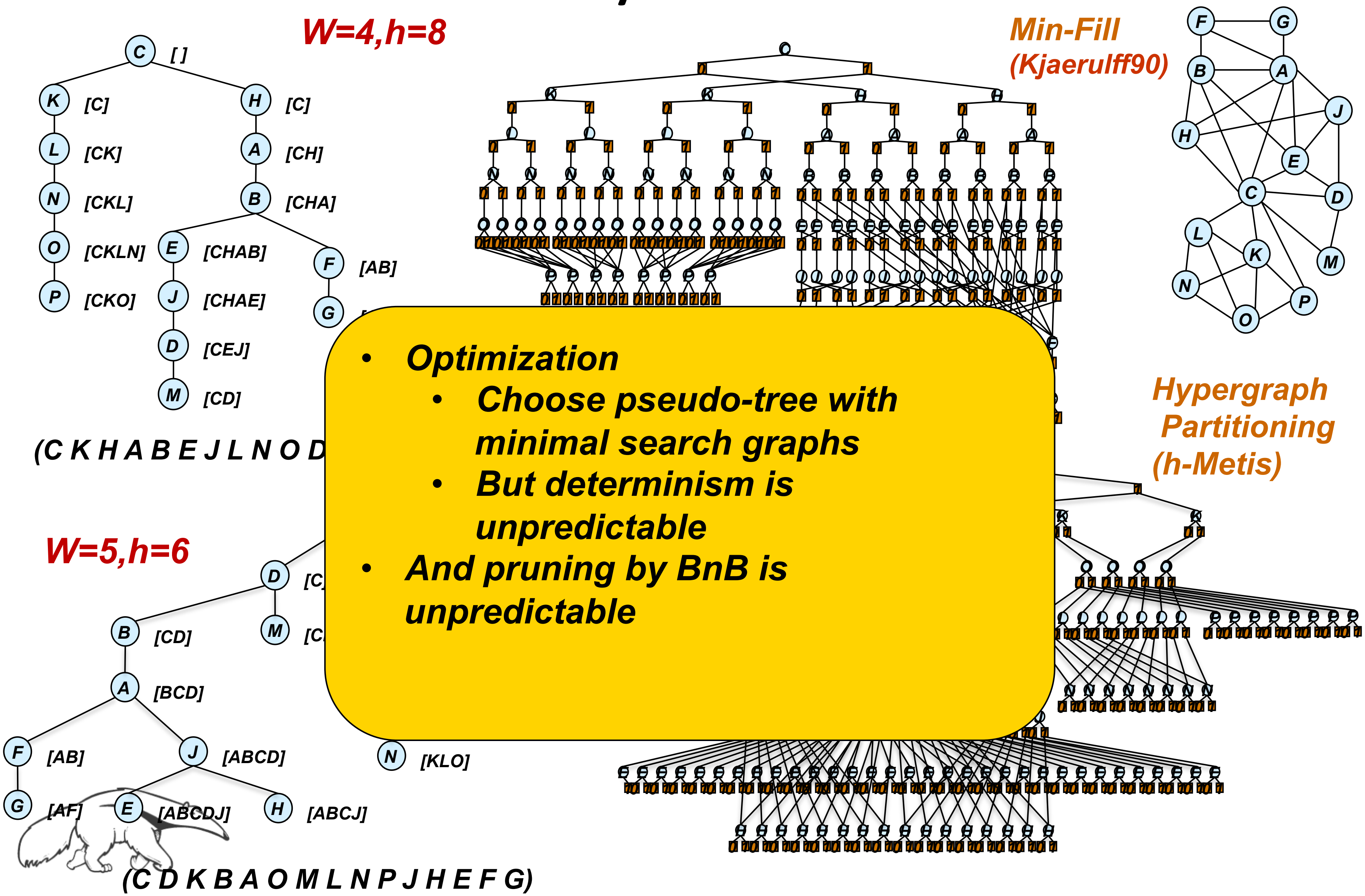


Optimization: Tailoring solver to the problem instance

Optimization

- The problem: determine how much time will take a solver $A(par)$ to solve a problem instance *c*
- Approaches:
  - Worst-case analysis based on the graph-parameters (tree-width), time space tradeoff for exact schemes
  - Learning: over a benchmark class
  - Stratified Sampling: of problem instance for a solver to estimate search space.

Worst-Case Analysis



- *Optimization*
  - *Choose pseudo-tree with minimal search graphs*
  - *But determinism is unpredictable*
- *And pruning by BnB is unpredictable*

Learning a Regression Model for Complexity Estimation (Otten and Dechter, 2012)

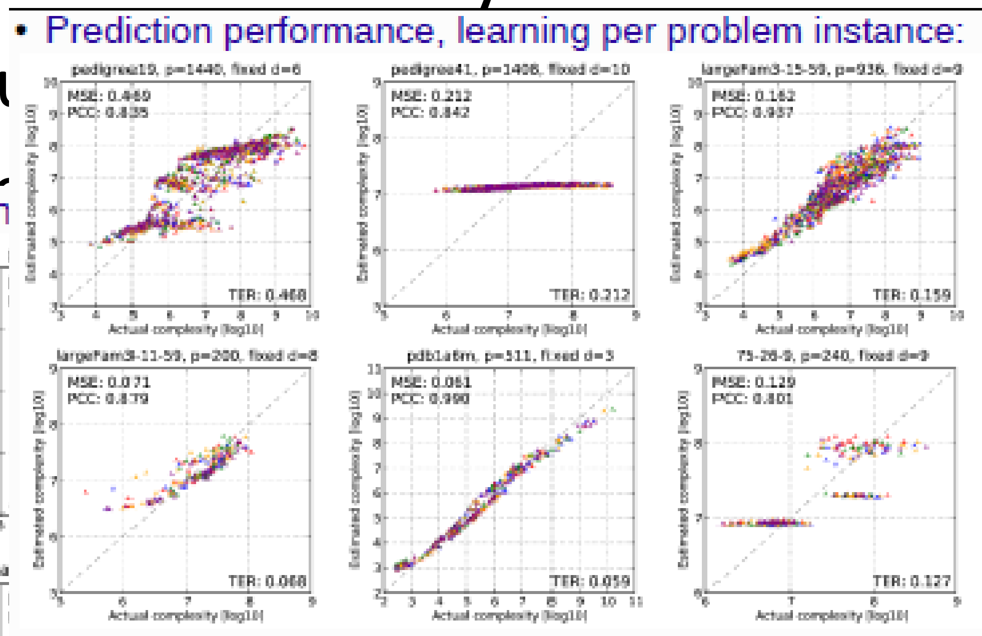
- Number of nodes *N(n)* as linear function of features $\varphi_j(n)$:

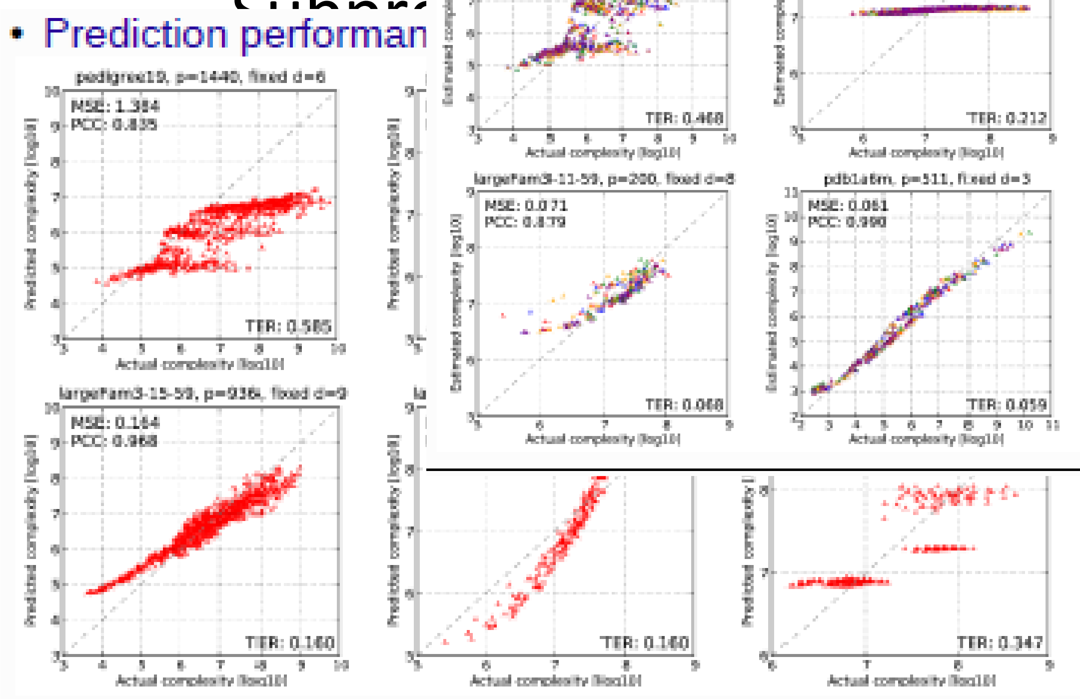
$$\log N(n) = \sum_j \lambda_j \varphi_j(n)$$

related: [Leyton-Brown, Nudelman, Shoham 2009]

Subproblem Features $\varphi_j(n)$

- Use both static and dynamic characteristics:
  - Structu
  - Subpro





mic):
roblem solution cost, derived
solution.
oblem solution cost, provided
.
en upper and lower bound, ex-
s" of the subproblem.
d on running 5000 node ex-

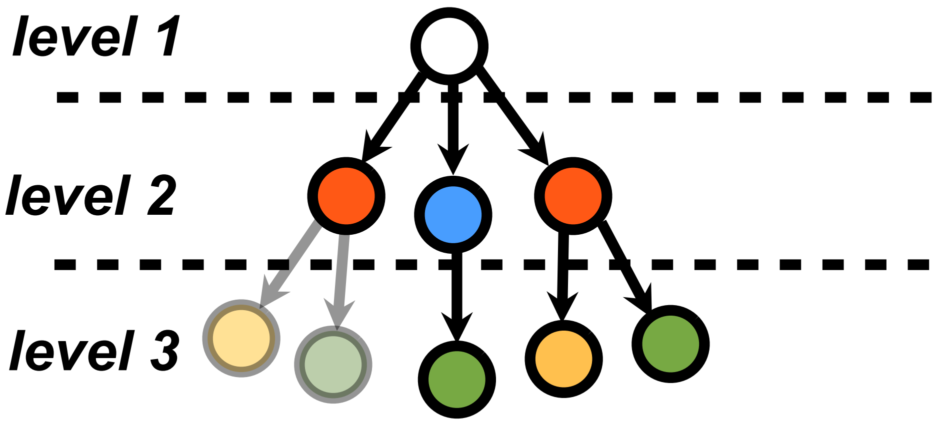
ing the heuristic.
e of determinism (zero proba-

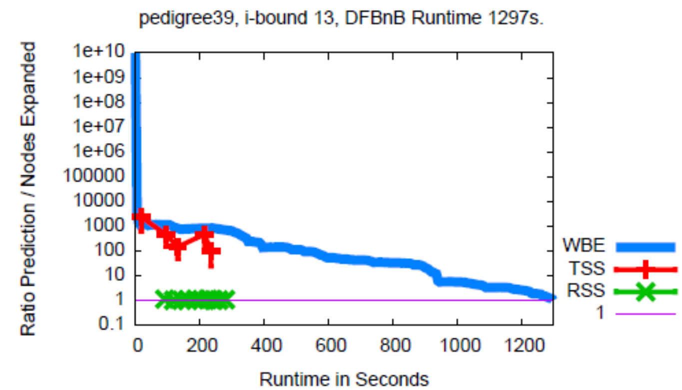
ding to pseudo tree leaf.
ed on running 5000 node ex-
obe of AOBB:
verage depth of terminal search nodes within probe.
verage node depth within probe (denoted $\bar{d}$).
verage branching degree, defined as $\sqrt[\bar{d}]{5000}$.
tatic):
ini bucket i-bound parameter.

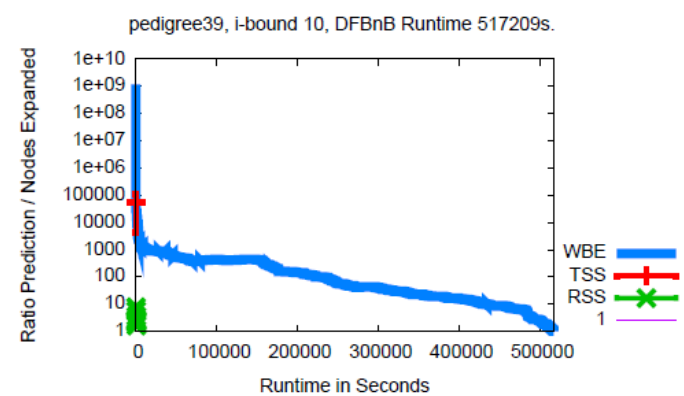
34: Max. subproblem variable context size minus mini bucket i-bound.

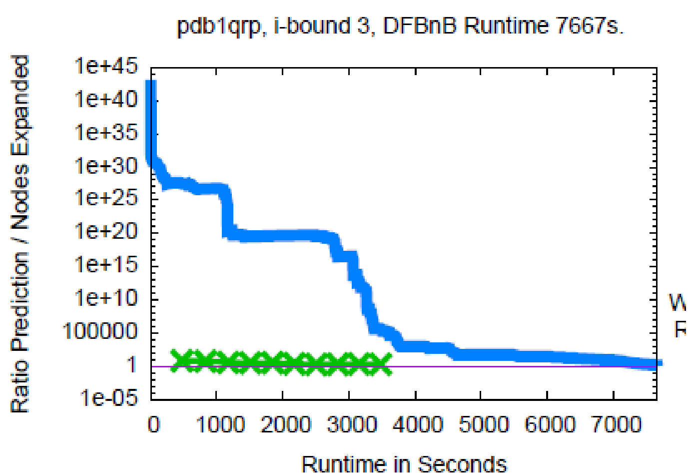
width of variables within subproblem, *when conditioning on subproblem root conditioning set.*

Predicting Depth-First Branch and Bound Search Trees (Levi, Lars and Dechter, IJCAI 2013, CP-2014 submission)









Chen, P.-C. 1992. Heuristic sampling: A method for predicting
the performance of tree searching programs. SIAM
Journal on Computing 21:295–315.

UCI Library: Summary

- Exact/anytime:
  - **Likelihood:** *BE, BEEM, VEC(w), AOlibPE(c-bound)*
  - ***MAP****: VE, BEEM (external memory/multi-core), AOBB(i), BRAOBB(i), DAOOPT(Distributed AOBB).*
  - Marginal Map (currently developed)
- *Approximation/anytime, for all queries:*
  - *BP, IJGP(i-bound)*
  - *IJGP-Importance Sampling(i-bound)*
  - *IJGP-SampleSearch(i-bound)*
  - *MBE (mini-bucket), Weighted-mini-bucket, reparameterized MB*
  - *STLS (currently developed, for MAP)*
- *Supporting schemes: Variable-ordering (IGVO)*

Summary

- What are graphical models? Queries
- Inference
- Search
- Time vs space, search vs inference
- Bounding inference (Variational: BP, GBP, weighted mini-bucket, cost-shifting)
- Bounding Search (Sampling)
- Anytime algorithms
- Optimization: Tailoring solvers to instance
- Recent algorithmic development
- Summary

Exact algorithms

Approximations Anytime

New work

For publication see:

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




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