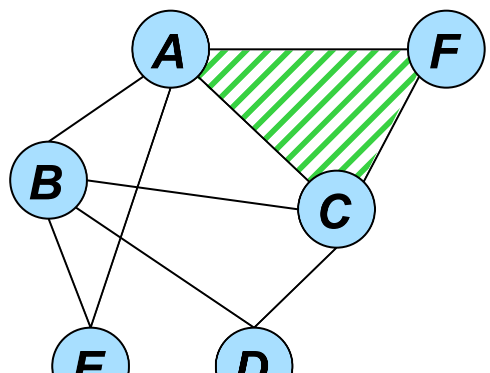
Modern Exact and Approximate MAP algorithms for Graphical Models

In the pursuit of universal solver

Rina Dechter

Donald Bren school of ICS, University of California, Irvine



TTI, May/2014

Sample Applications for Graphical Models

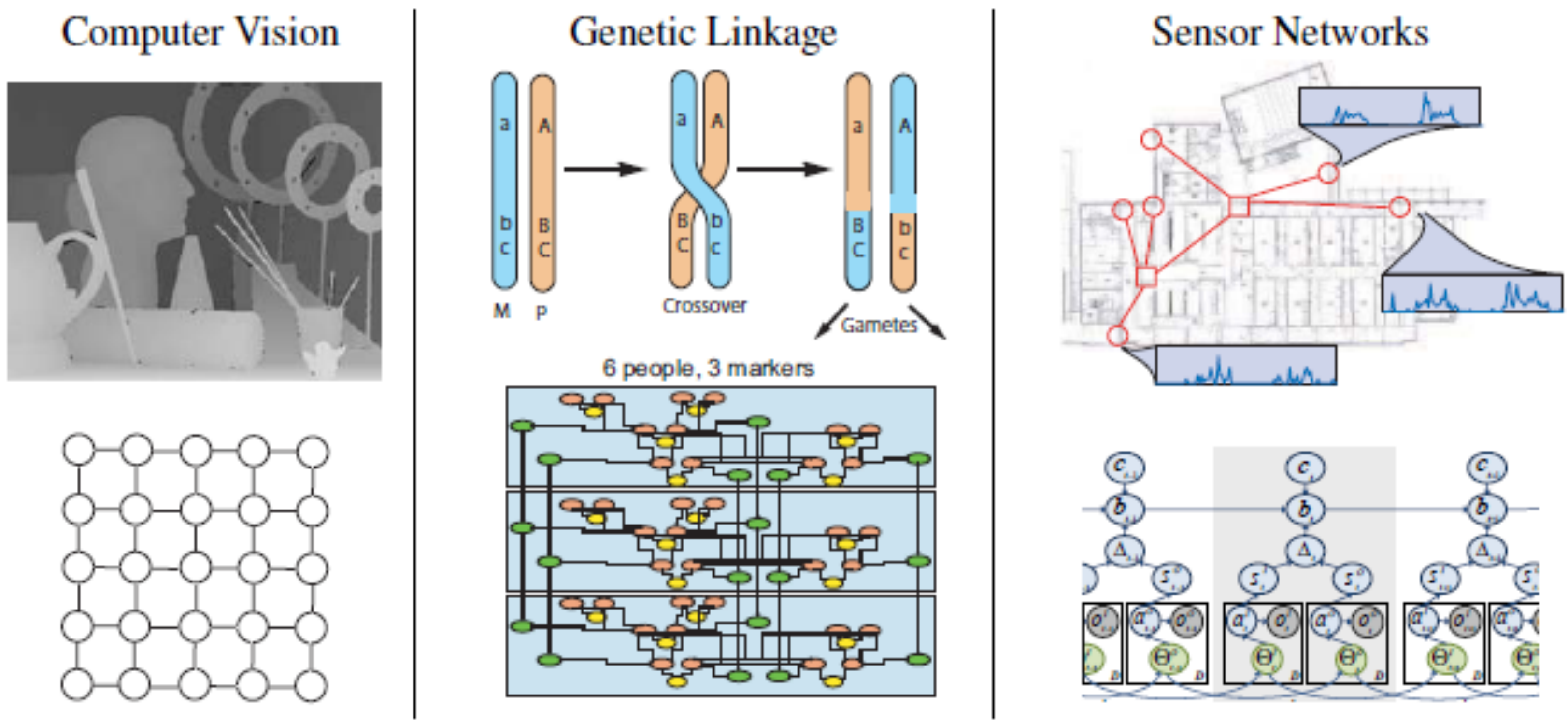
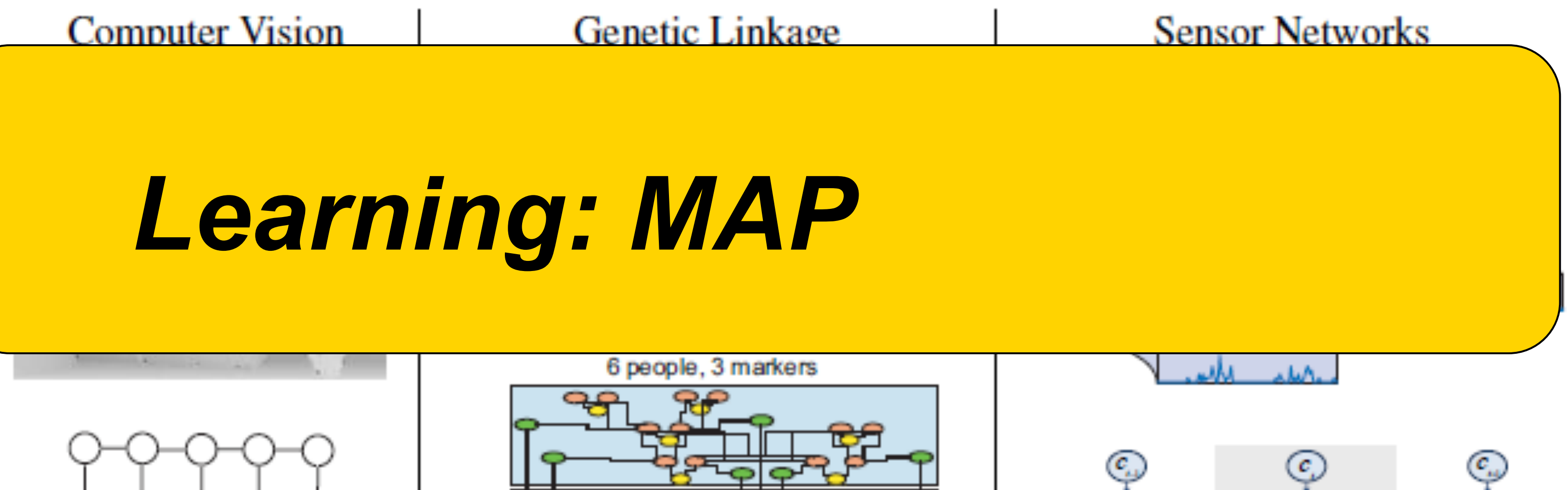


Figure 1: Application areas and graphical models used to represent their respective systems: (a) Finding correspondences between images, including depth estimation from stereo; (b) Genetic linkage analysis and pedigree data; (c) Understanding patterns of behavior in sensor measurements using spatio-temporal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

Sample Applications for Graphical Models



Learning: MAP

Reasoning: MAP

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.

How to design a good MAP solver

- Heuristic Search
- The core of a good search algorithm
  - A compact search space
  - A good heuristic evaluation function
  - A good traversal strategy
- Anytime search yields a good approximation.

Outline

- Graphical models, Queries
- Inference Algorithms
- AND/OR search
- Optimistic bounding schemes (mini-bucket, re-parameterization\cost-shifting, soft AC)
- BRAOBB: anytime DFS for AND/OR
- Experiments/competitions (putting it all together)
- Recent work: Parallelism, m-best, weighted best-first, marginal map, tree-SLS
- Conclusion

Graphical Models, Queries, Algorithms

Graphical Models

- A graphical model M = (X,D,F,⊗)
  - X = $\{X_1, \ldots, X_n\}$ variables
  - D = $\{D_1, \ldots, D_n\}$ domains
  - F = $\{f_1, \ldots, f_t\}$ functions over $\{S_1, \ldots, S_t\}$
  - Global function ⊗ **Combine**

$$F(X) = \otimes_{f \in F} f(x)$$

  - Reasoning (queries) ⇩ **marginalize**

$$G(Z) = \Downarrow_{X-Z} F(X) = \Downarrow_{X-Z} \otimes_{f \in F} f(x)$$

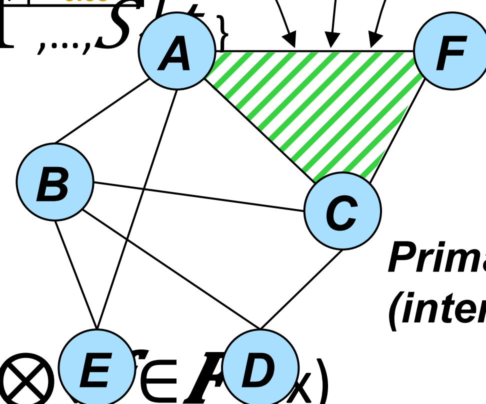
Conditional Probability Table (CPT)

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

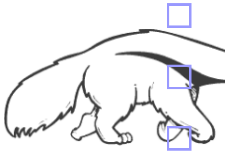
A	C	F
red	green	blue
blue	red	red
blue	blue	green
green	red	blue

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



Primal graph (interaction graph)

- Queries:
  - **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**: $\min_X \Sigma_j F_j$



When combine and marginalize obey Some properties they can be solved By the same algorithms

(Bistareli , Rossi and Montanari, 1995, Shenoy , Shafer, 1990, Kask et. al., 2005 .)

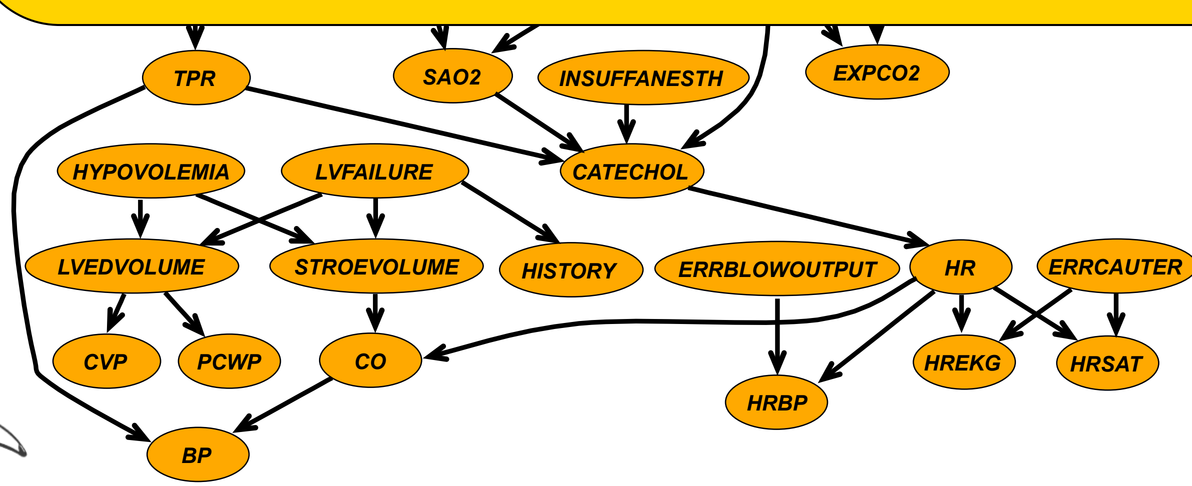
Monitoring Intensive-Care Patients

The “alarm” network - 37 variables, 509 parameters (instead of 2^{37})

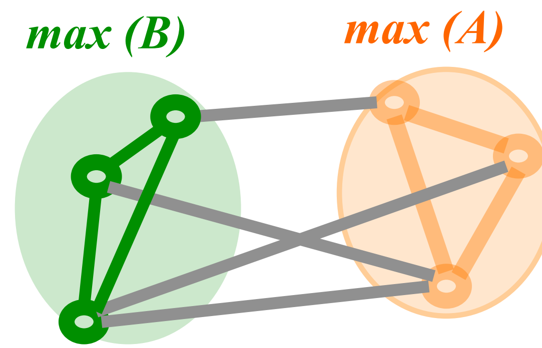
$$P(x_1 \ldots x_n) = \prod_i p(x_i \mid parent(x_i))$$

$$P(e) = \sum_{X-E} \prod_i p(x_i \mid parents(x_i))$$

$$MAP = \max_x \prod_i p(x_i \mid parents(x_i))$$



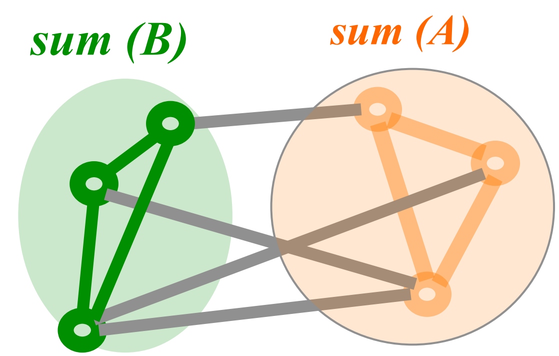
Queries



$X = A \cup B$

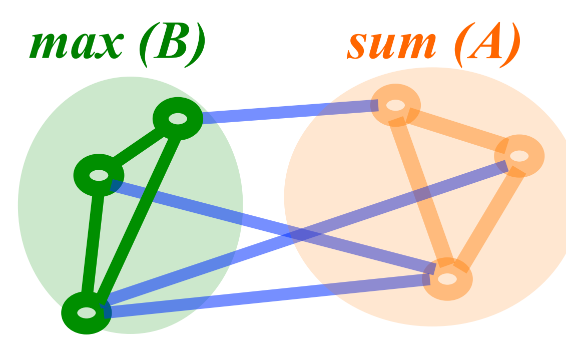
- **Optimization Queries: MAP/MPE queries:**

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



- **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***

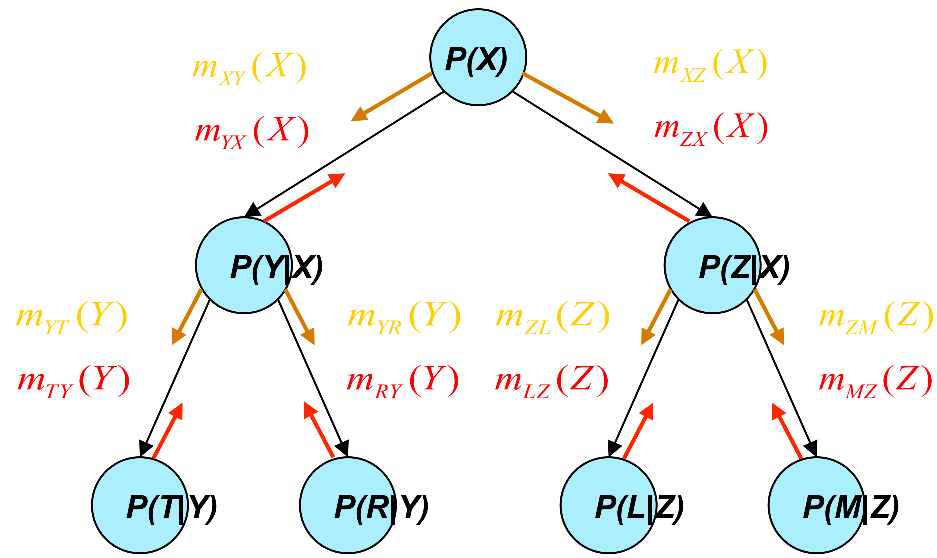
$$\boldsymbol{x}_B^* = \arg\max_{\boldsymbol{x}_B} \sum_{\boldsymbol{x}_A} \prod_{\alpha} \psi(\boldsymbol{x}_\alpha)$$

Also satisfiability and expected utility

Tree-solving is easy

Belief updating (sum-prod)

CSP – consistency (projection-join)

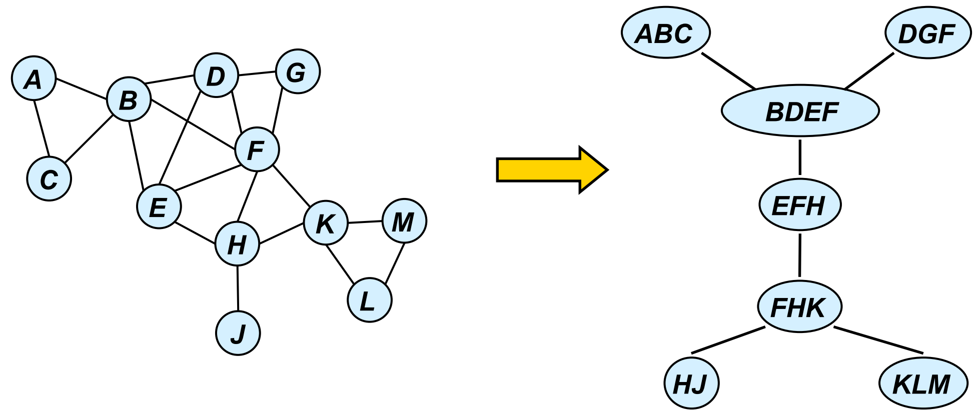


MPE (max-prod)

#CSP (sum-prod)

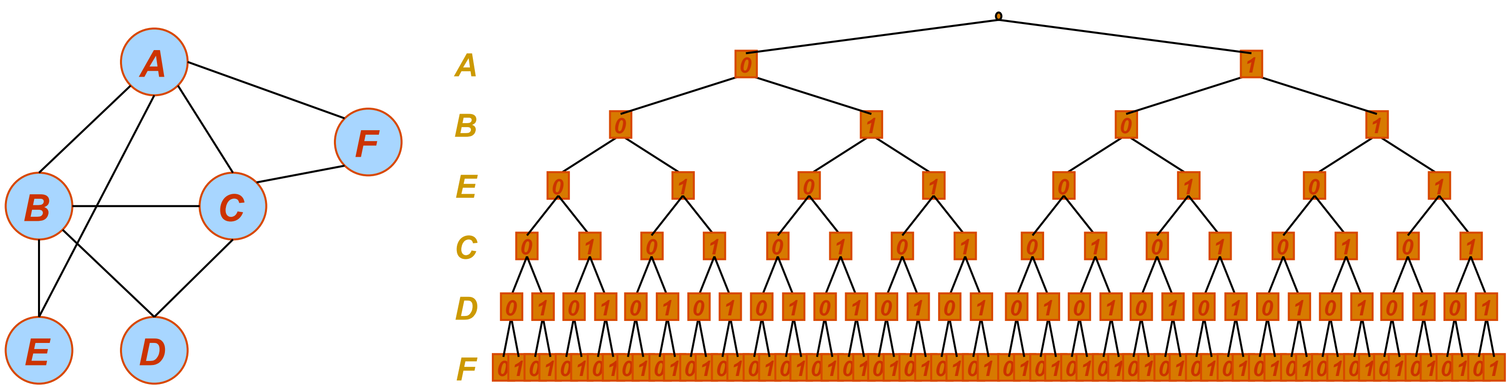
Trees are processed in linear time and memory

Inference vs conditioning-search



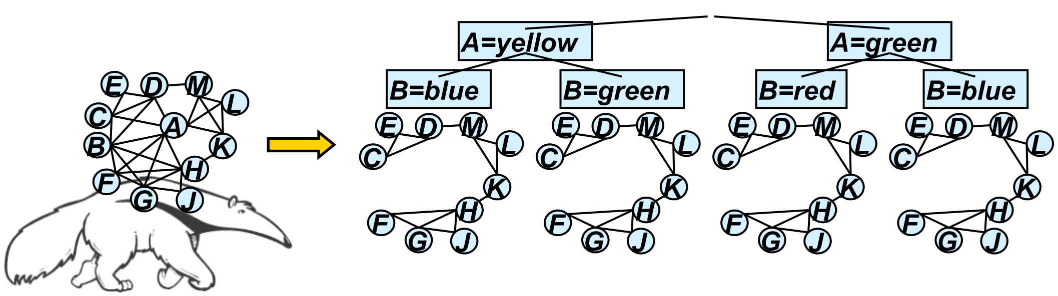
Inference

exp(w*) time/space



Search

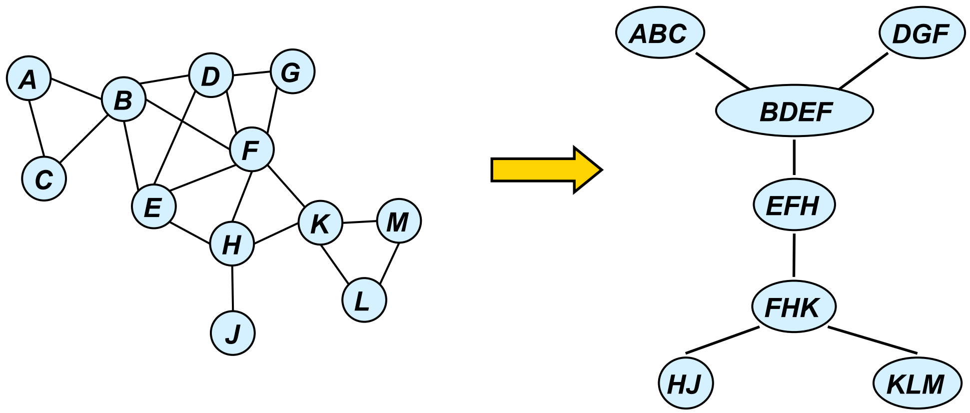
Exp(n) time
O(n) space



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

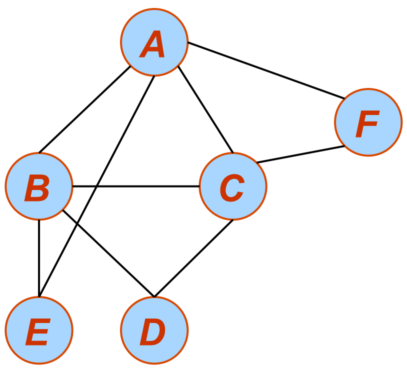
w: user controlled

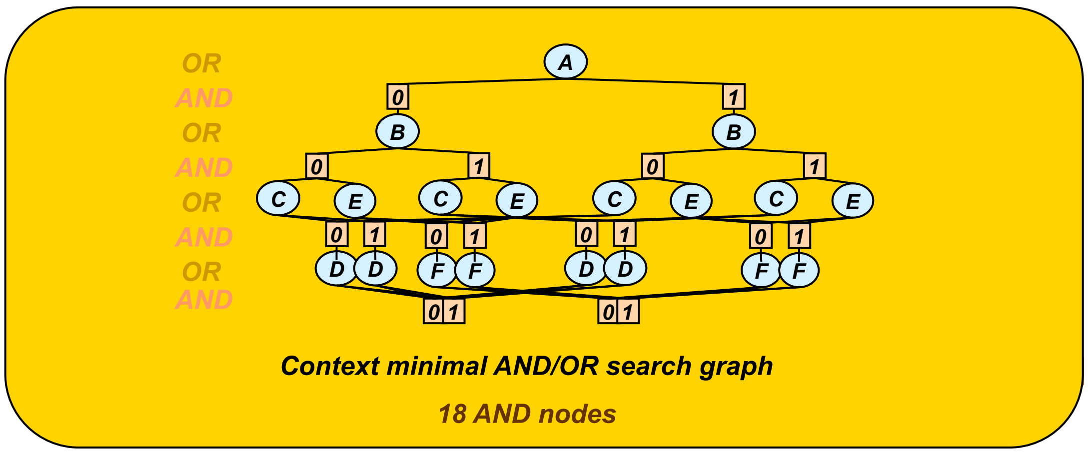
Inference vs conditioning-search



Inference

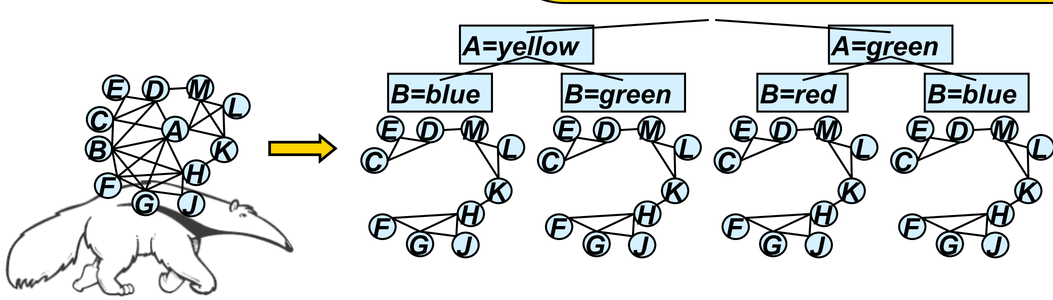
exp(w) time/space*





Search

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



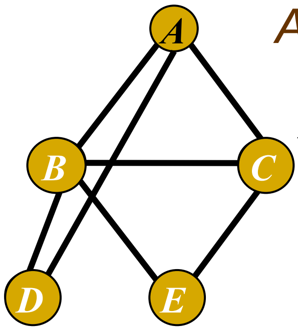
Search+inference:
Space: exp(q)
Time: exp(q+c(q))

q: user controlled

- ***Solving MAP by Inference:***
- ***Non-serial Dynamic programming***
- ***The induced-width/treewidth***

Finding MAP by Bucket El

Algorithm BE-mpe (Dechter 199



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

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

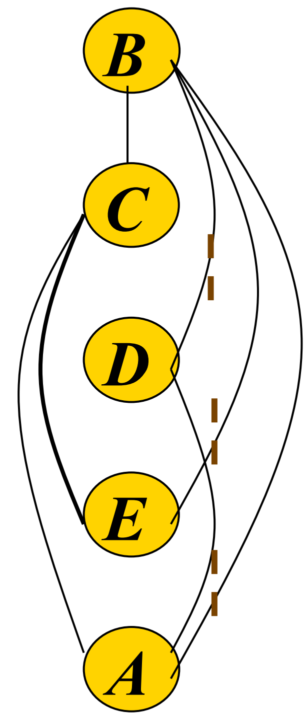
bucket B: $P(b|a)$ $P(d|b,a)$ $P(e|b,c)$

bucket C: $h^B(a,d,c,e)$

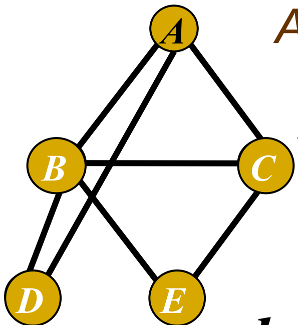
bucket D:

bucket E:

bucket A:



Finding MAP by Bucket Elimination



Algorithm BE-mpe (Dechter 199

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

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

$$\max_X \prod$$

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

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

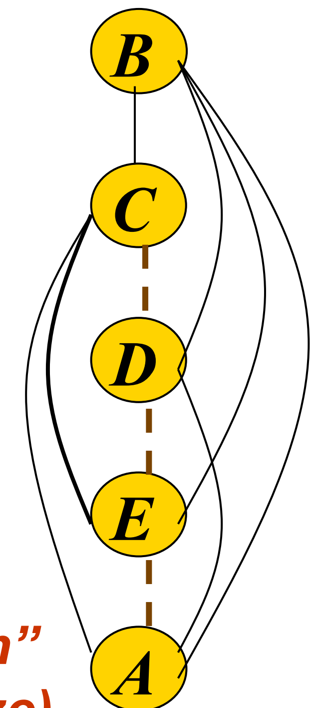
bucket D: $h^C(a,d,e)$

bucket E: $e=0$ $h^D(a,e)$

bucket A: $P(a)$ $h^E(a)$

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

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')

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

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

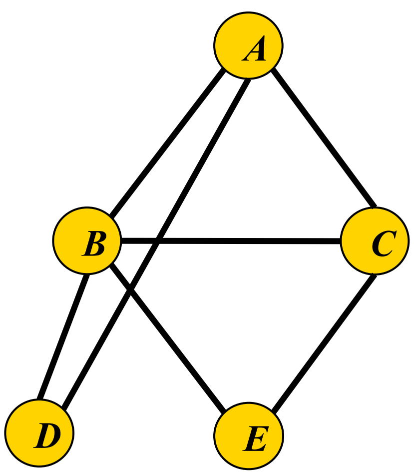
1. $a' = \arg\max_a P(a) \cdot h^E(a)$

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

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

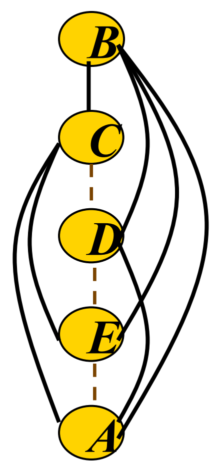
The Induced-width/treewidth

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

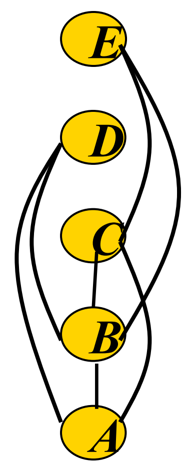


"Moral" graph

The effect of the ordering:



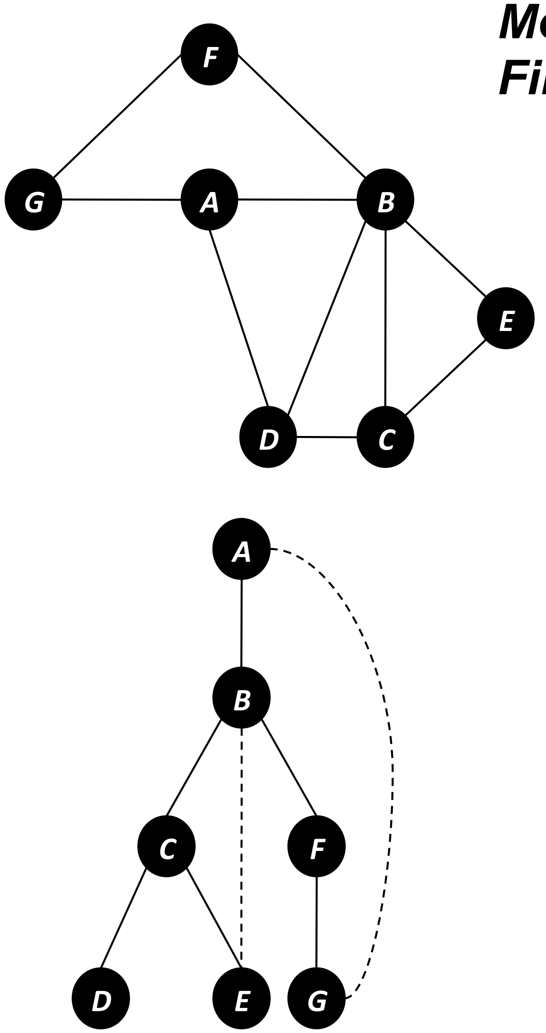
$w^*(d_1) = 4$

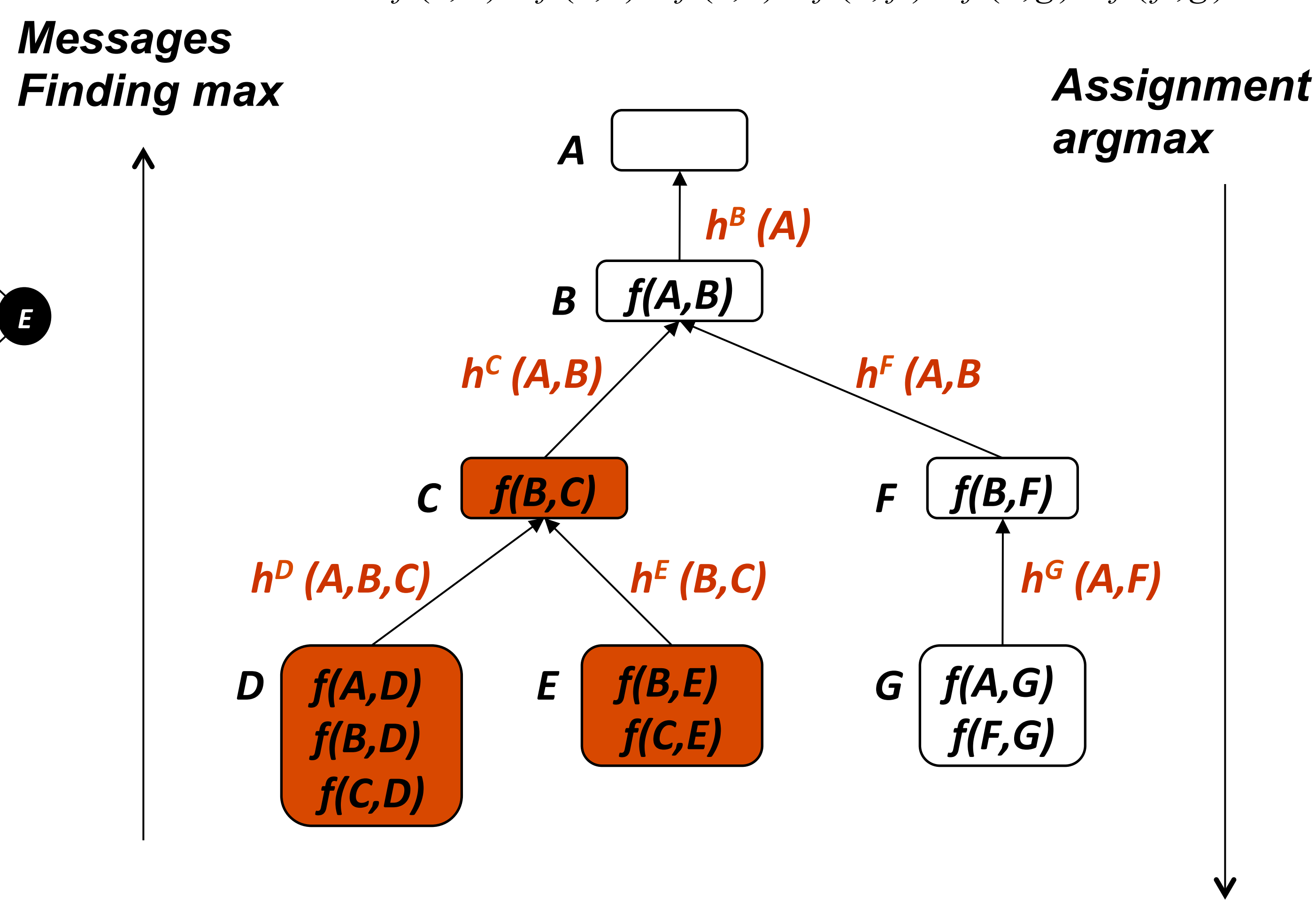


$w^*(d_2) = 2$

Bucket Elimination

$$\min_{a,b,c,d,e,f,g} f(a,b)+f(a,d)+f(b,c)+f(a,d)+f(b,d)+ f(c,d)+f(b,e)+f(c,e)+f(b,f)+f(a,g)+f(f,g) =$$



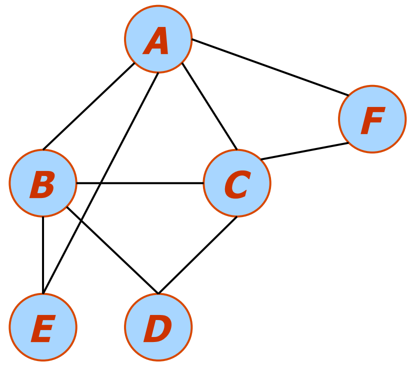


Ordering: (A, B, C, D, E, F, G)

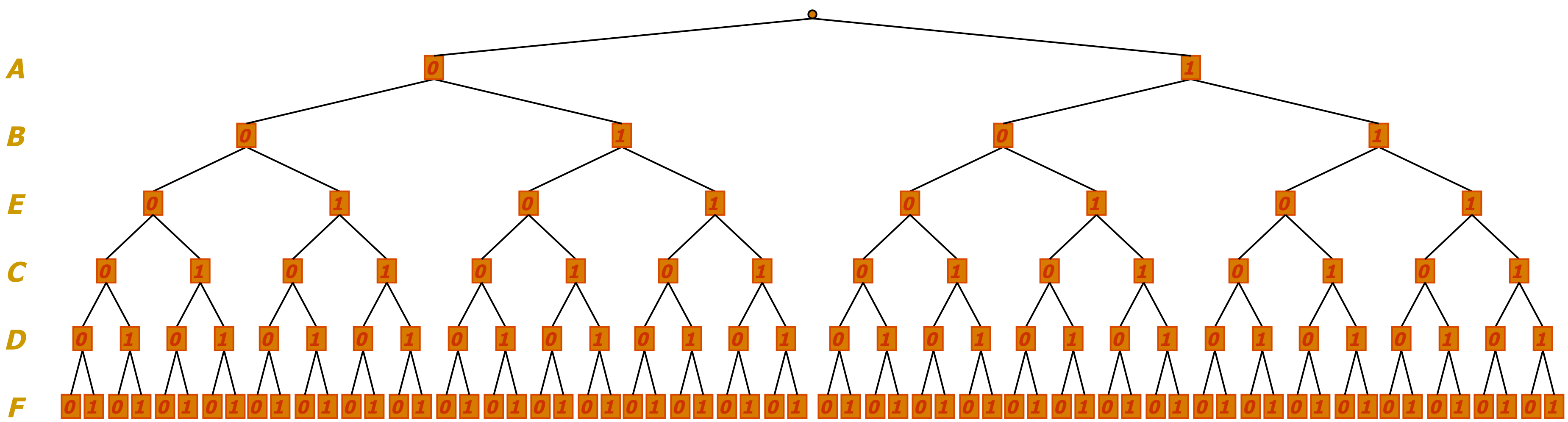
Search:

The AND/OR Search graph

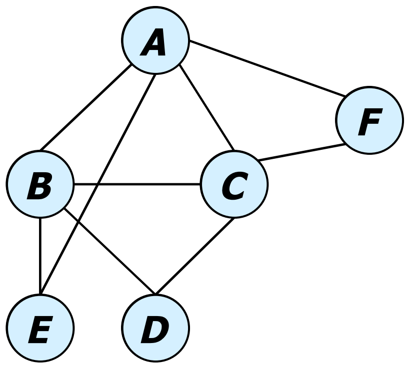
Classic OR Search Space



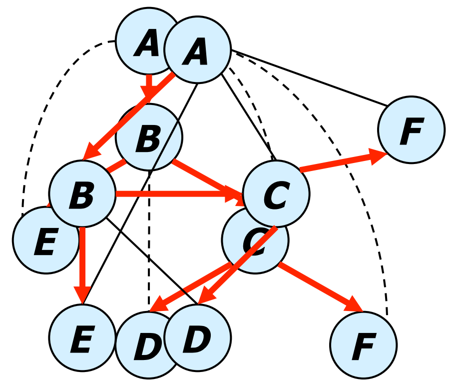
Ordering: A B E C D F



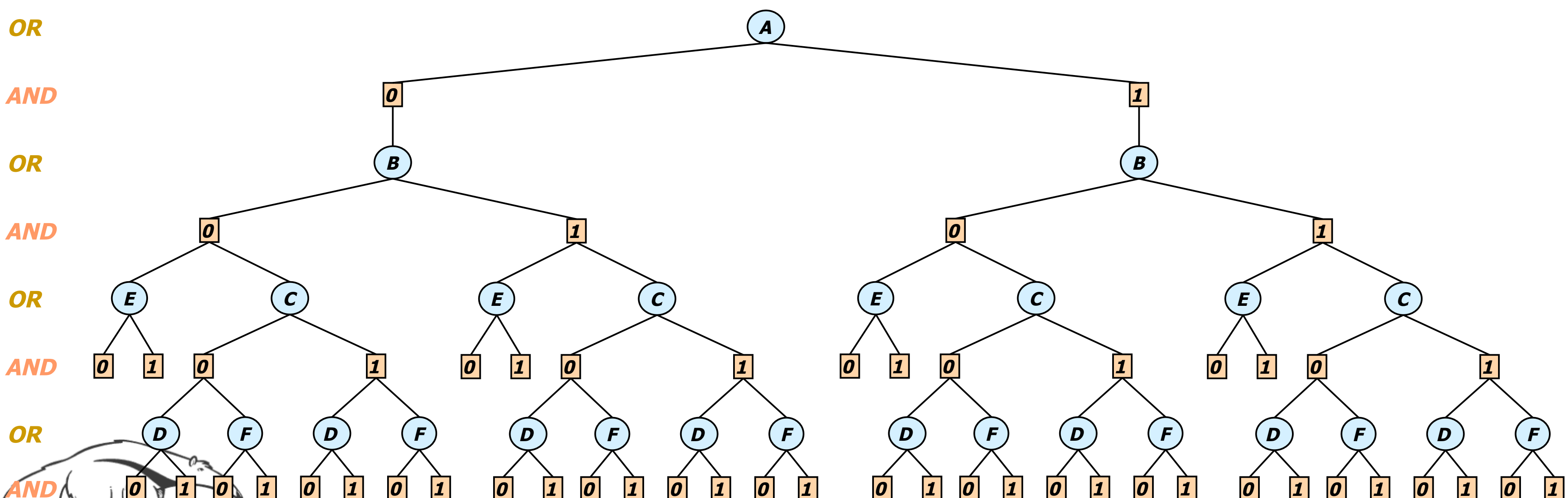
AND/OR Search Space



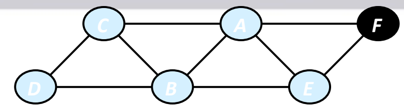
Primal graph

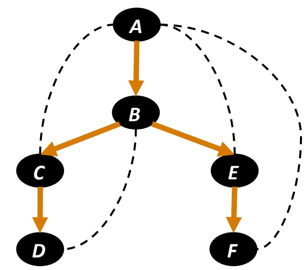


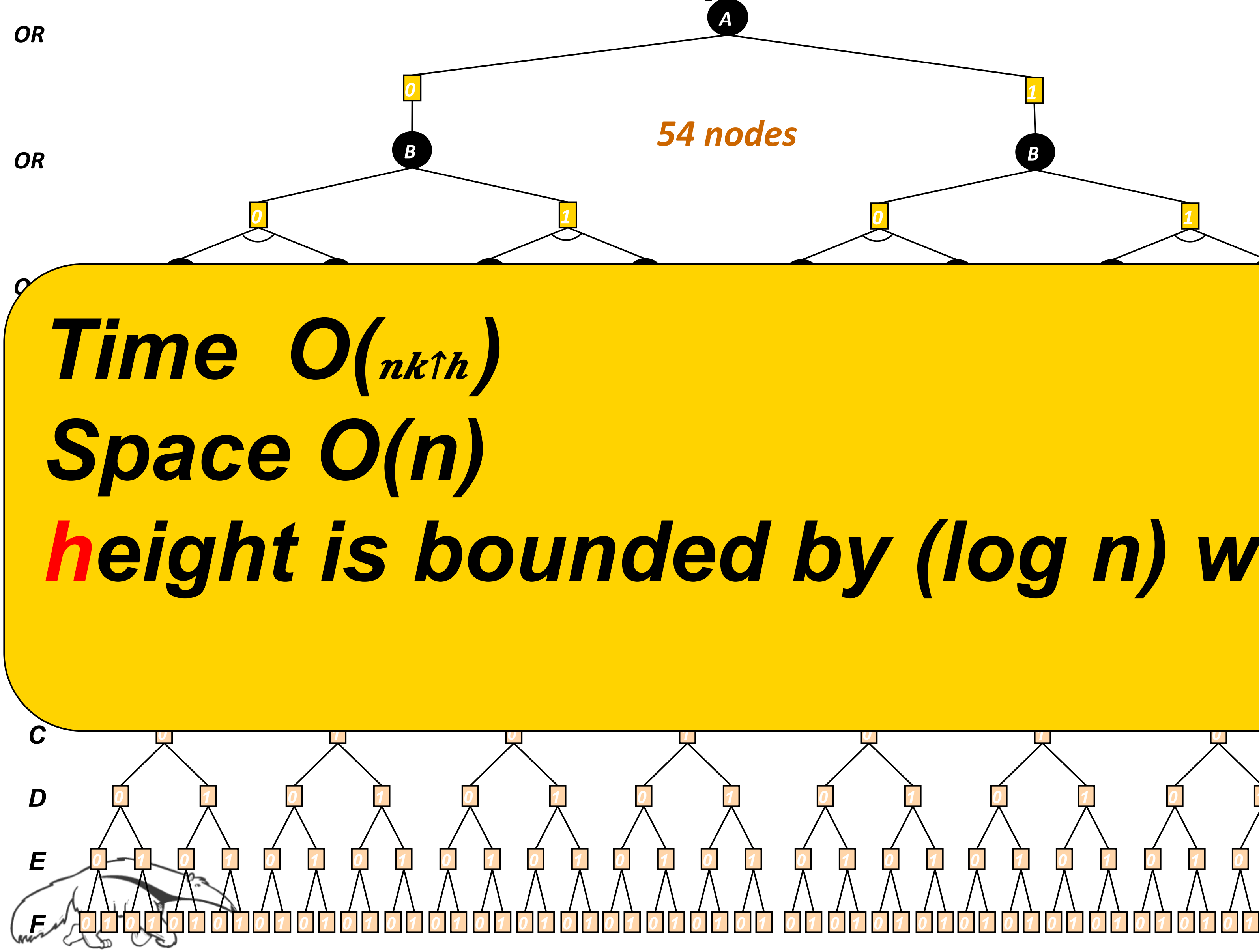
DFS tree



AND/OR vs. OR Spaces

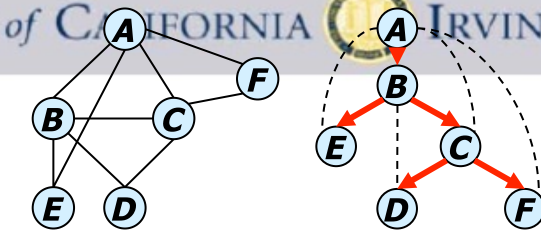


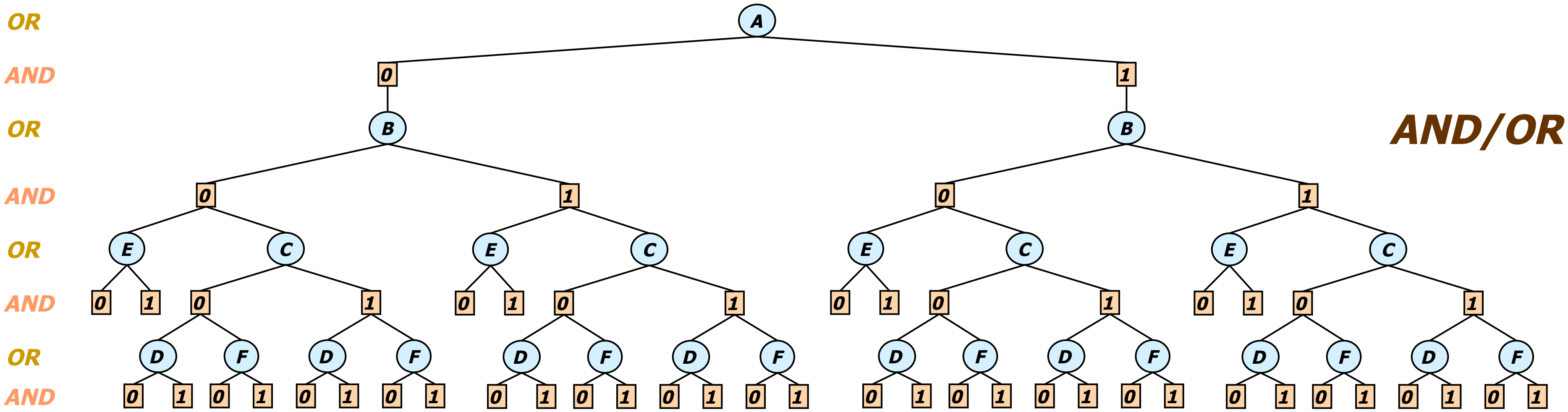




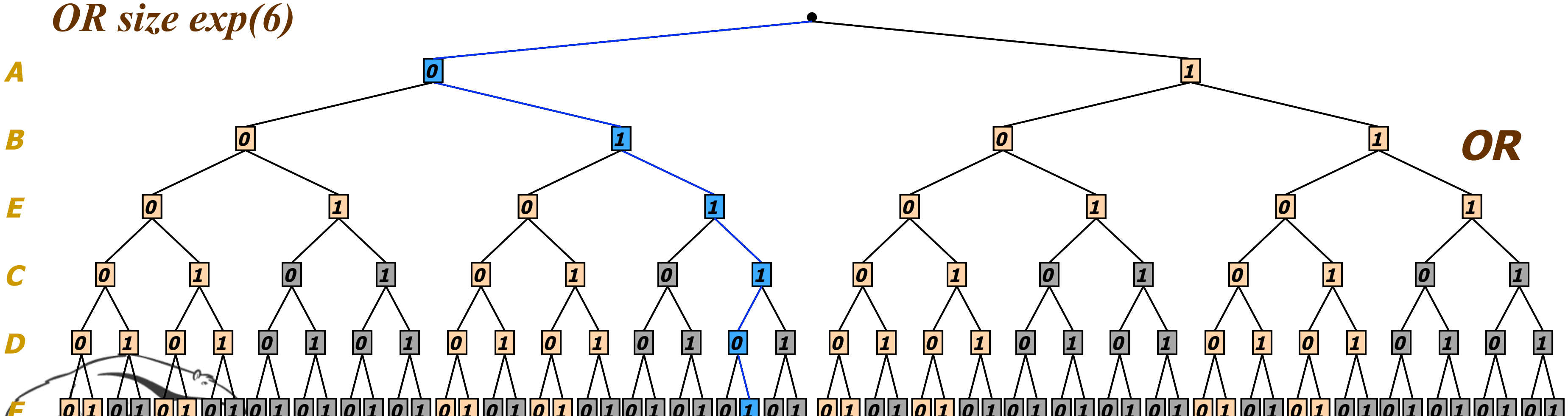
Time $O(nk^h)$
Space $O(n)$
height is bounded by (log n) w*

AND/OR vs. OR



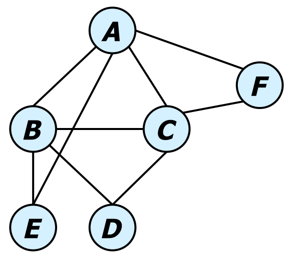


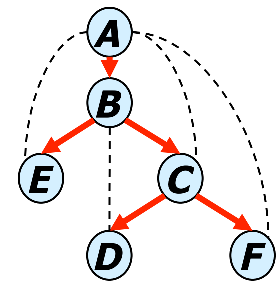
AND/OR size: exp(4),
OR size exp(6)

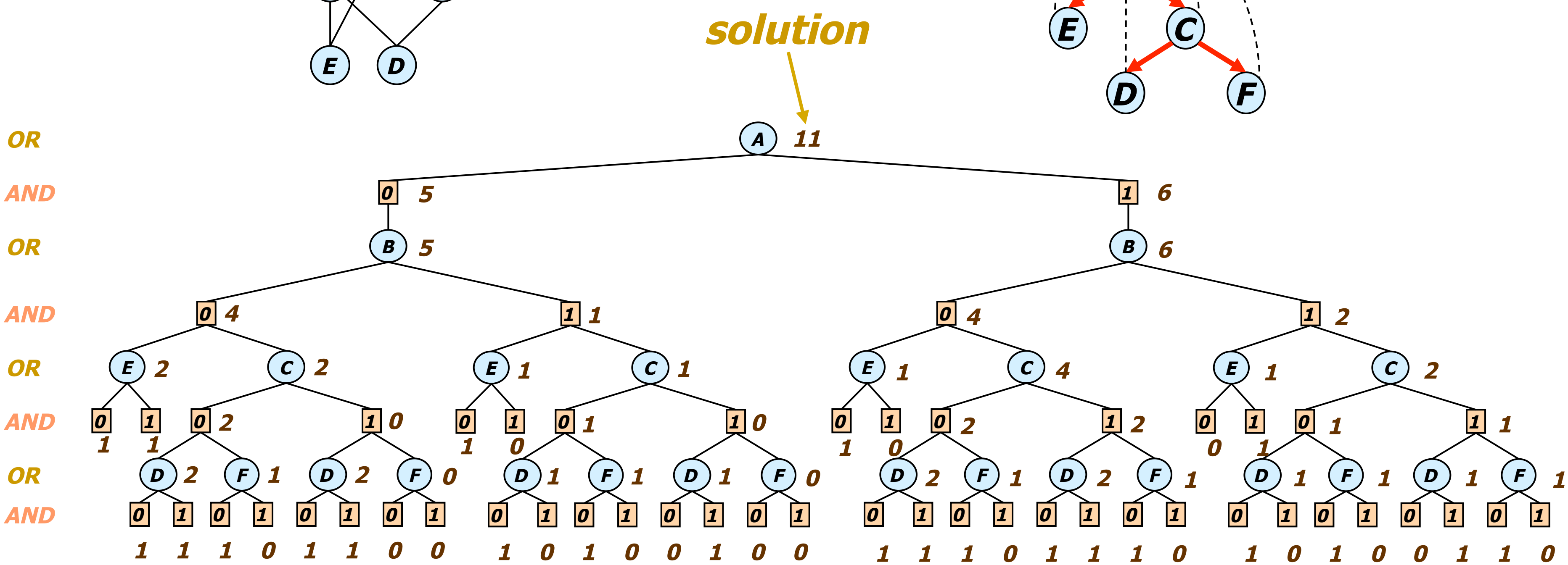


DFS algorithm (#CSP example)

Value of node = number of solutions below it



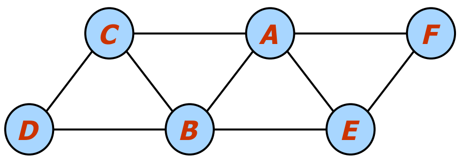


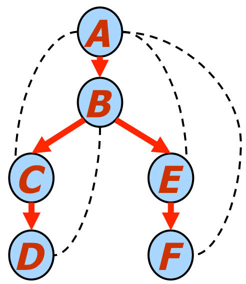


OR node: Marginalization operator (summation)

AND node: Combination operator (product)

AND/OR Tree Search for COP





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

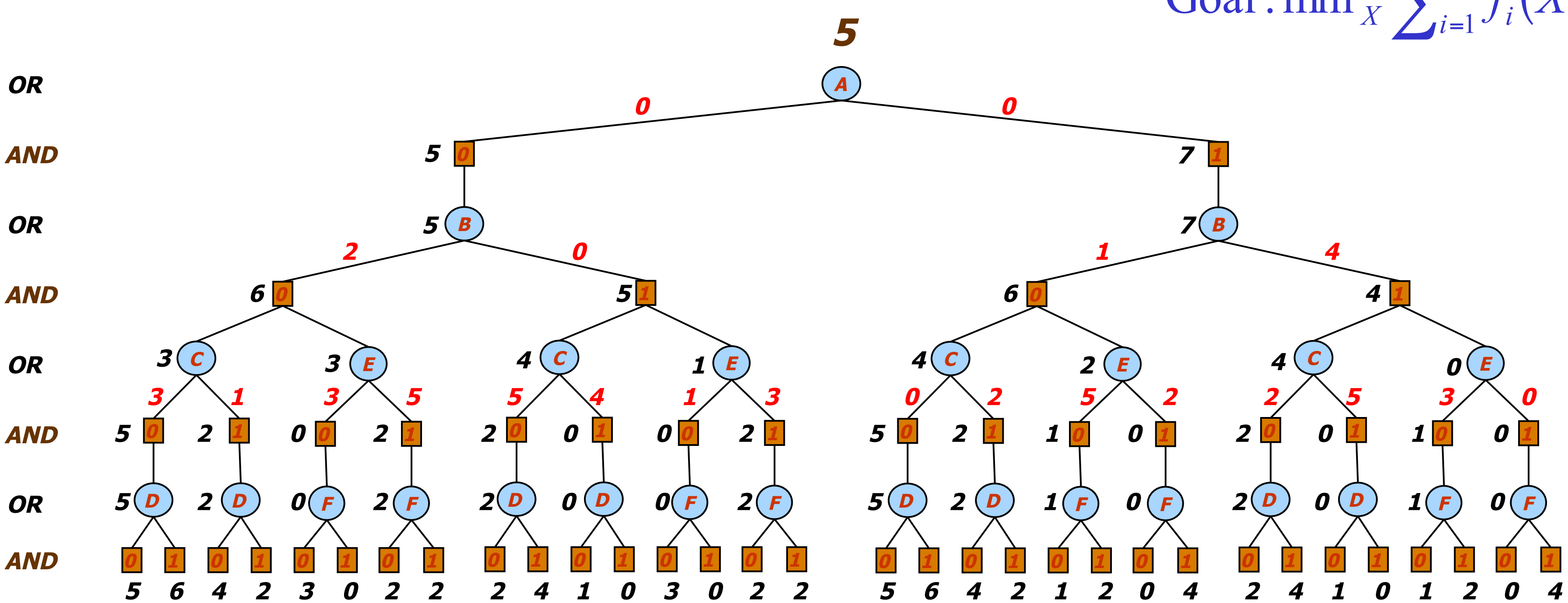
B	D	f_6
0	0	4
0	1	2
1	0	1
1	1	0

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

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

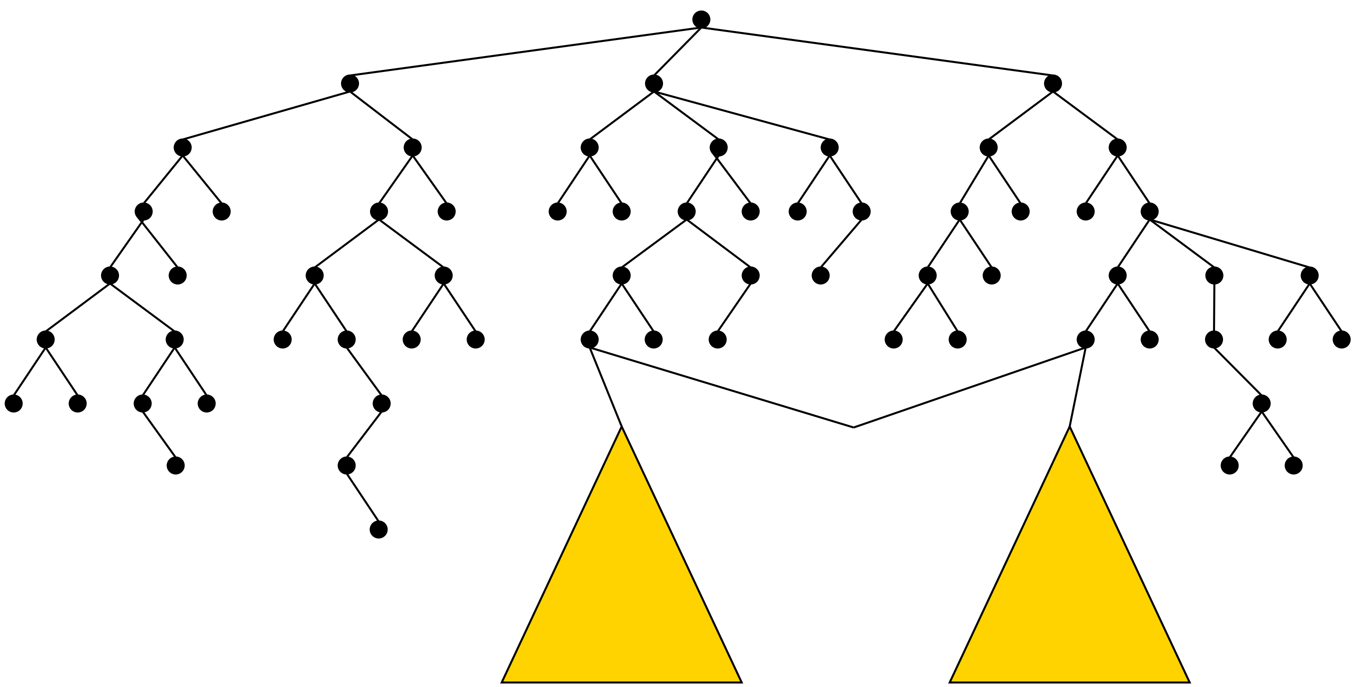


AND node = Combination operator (summation)

OR node = Marginalization operator (minimization)

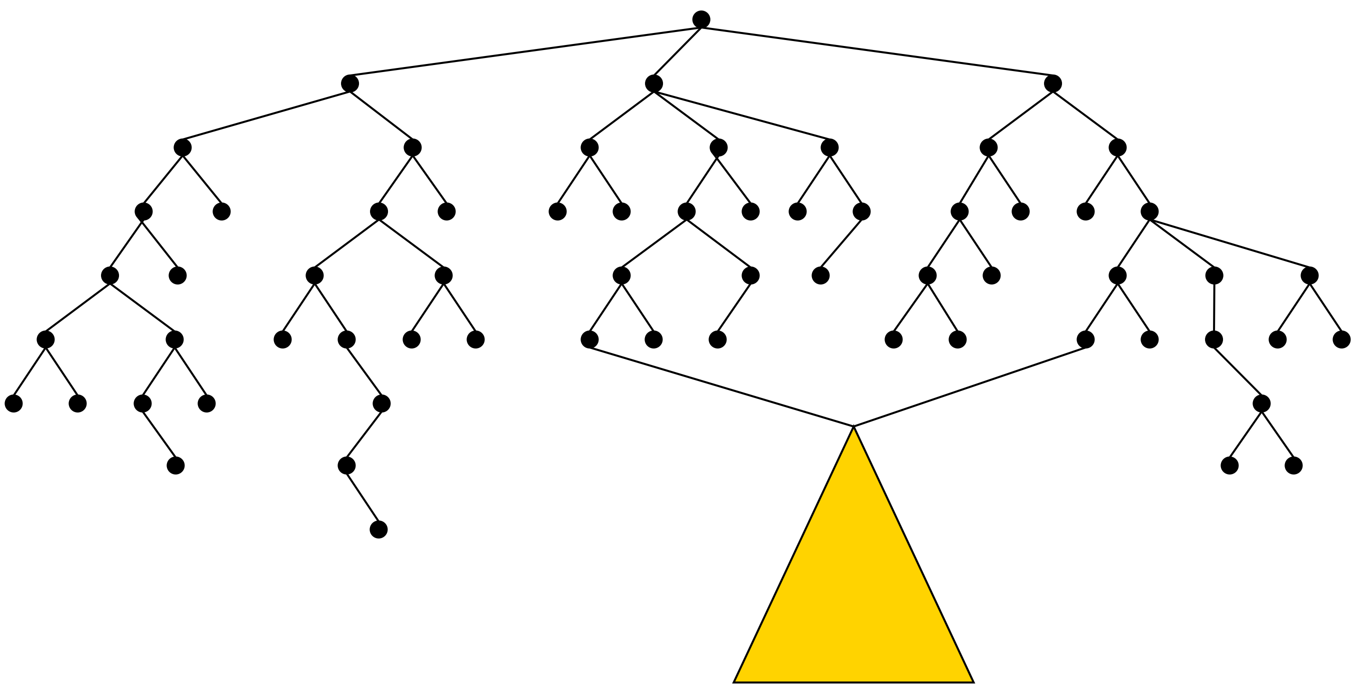
From Search Trees to Search Graphs

- Any two nodes that root identical sub-trees or sub-graphs can be merged

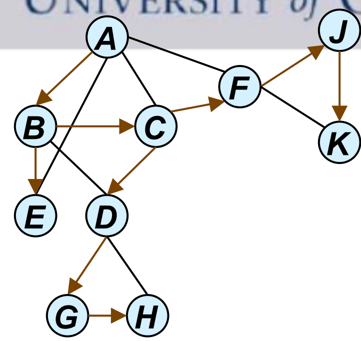


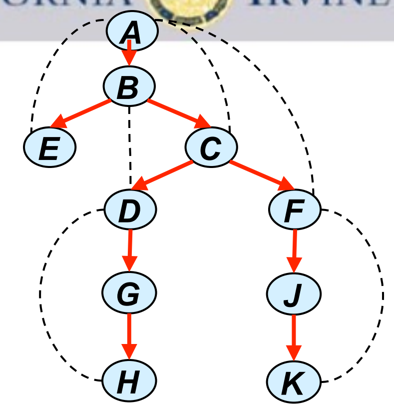
From Search Trees to Search Graphs

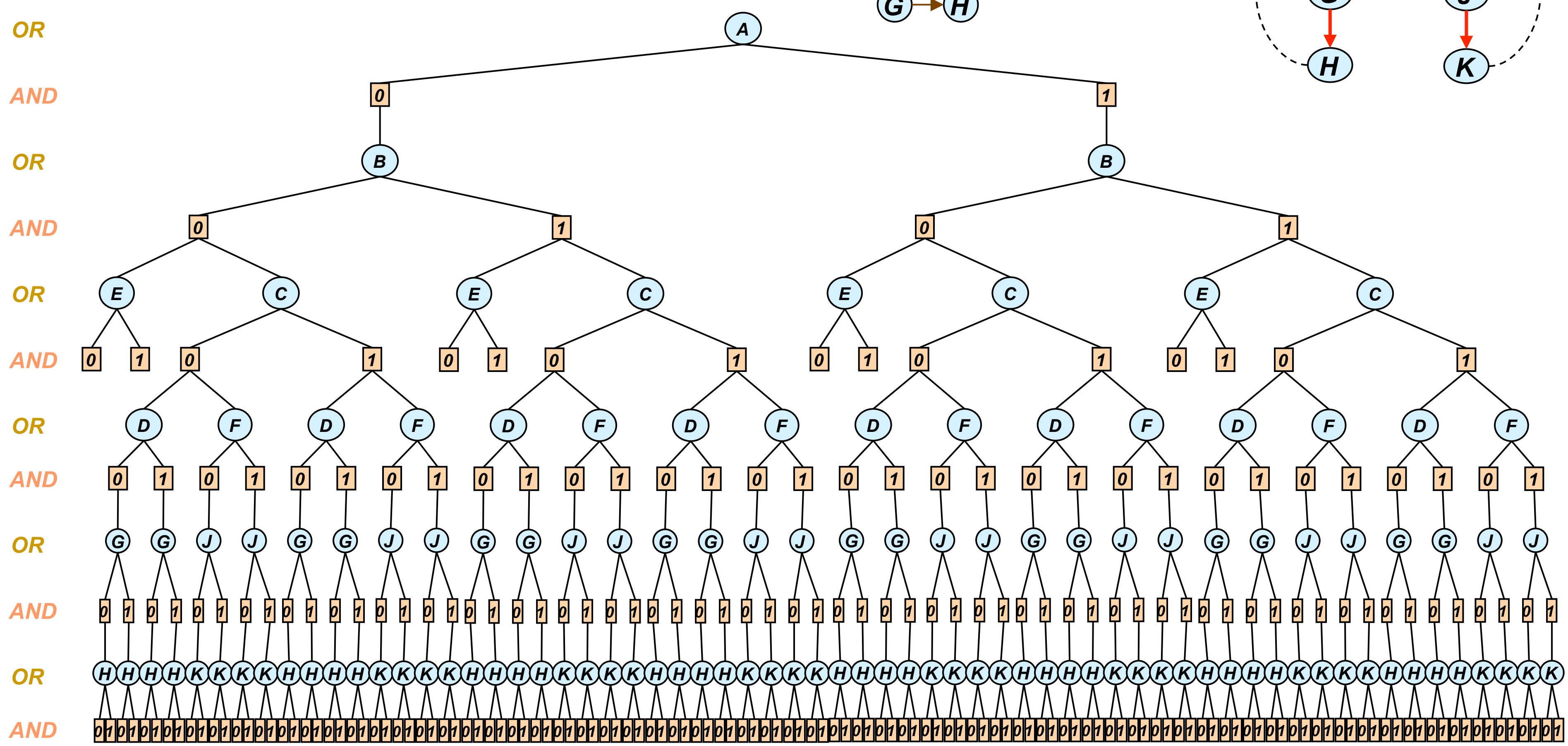
- Any two nodes that root identical sub-trees or sub-graphs can be merged



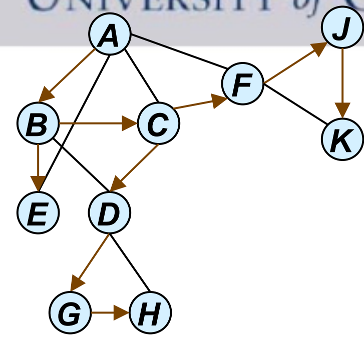
From AND/OR Tree

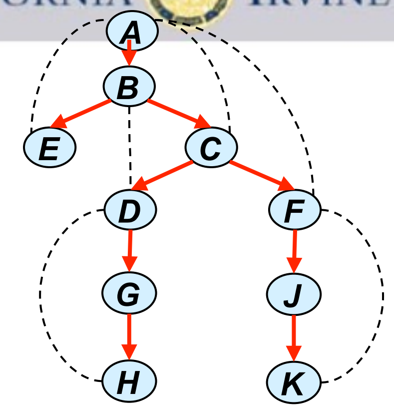


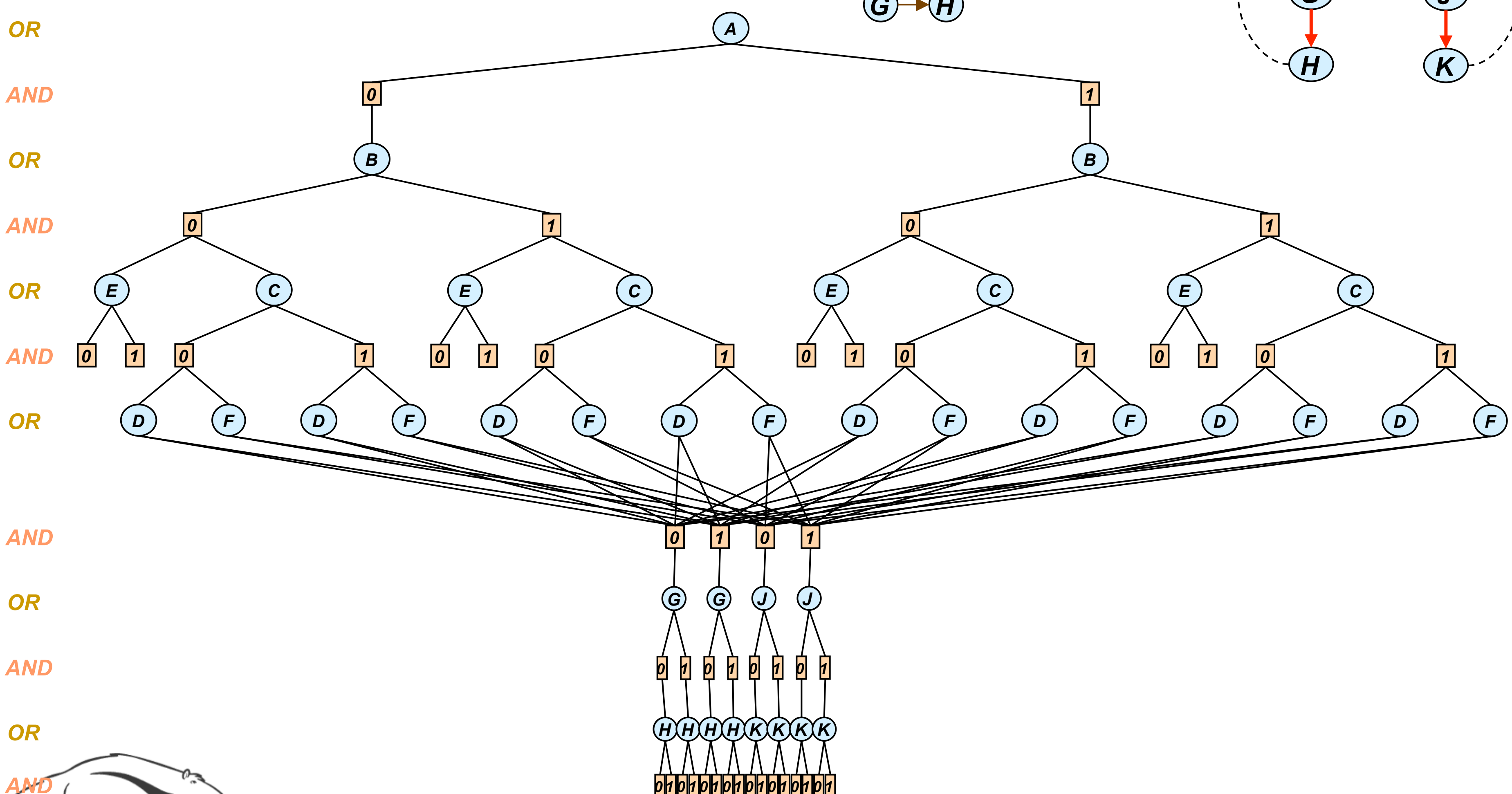




An AND/OR Graph

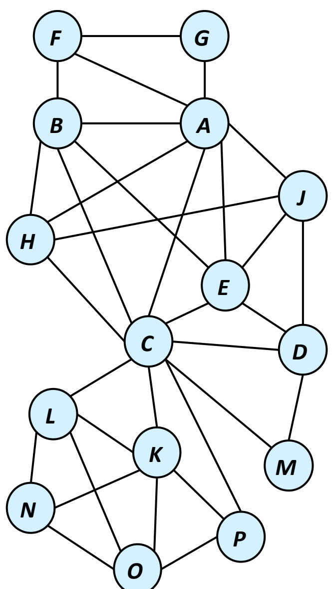


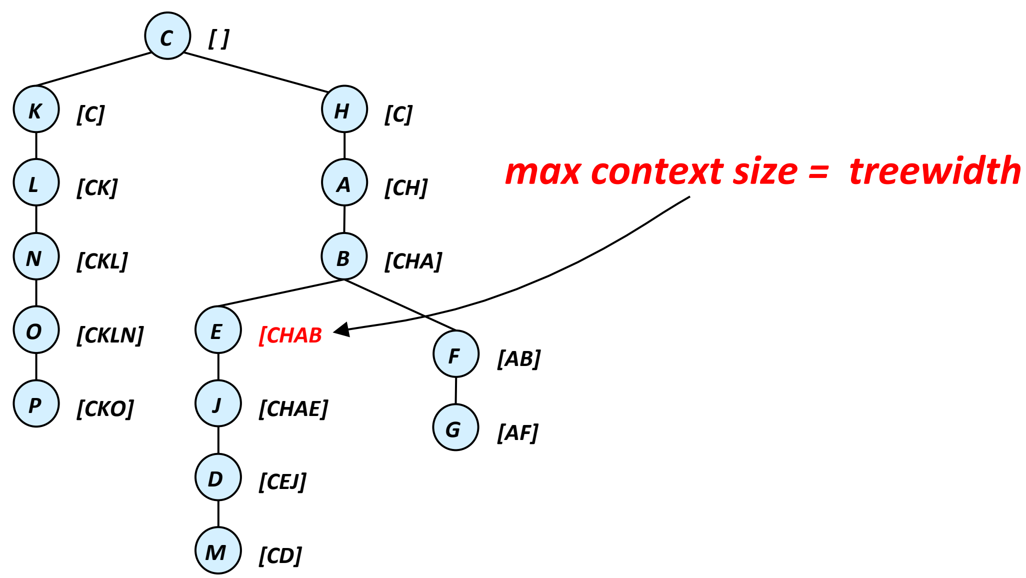




How Big Is The Context?

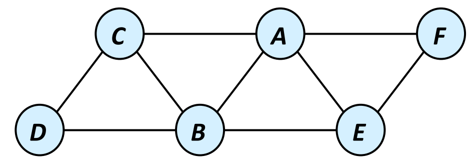
Theorem: The maximum context size for a pseudo tree is equal to the treewidth of the graph along the pseudo tree.



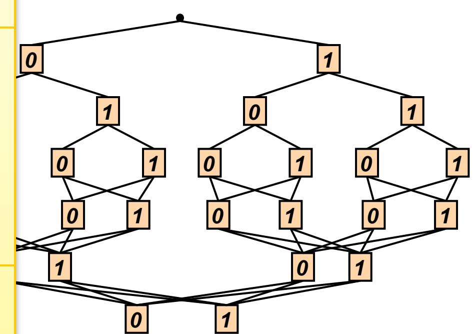


(C K H A B E J L N O D P M F G)

All Four Search Spaces



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

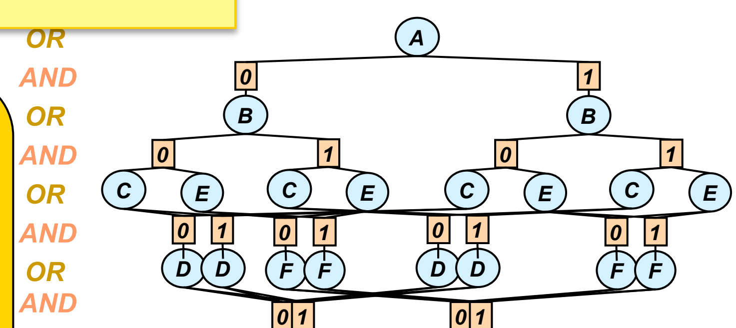


xt minimal OR search graph

28 nodes

Computes any query:

- *Constraint satisfaction*
- *Optimization*
- *Weighted counting*

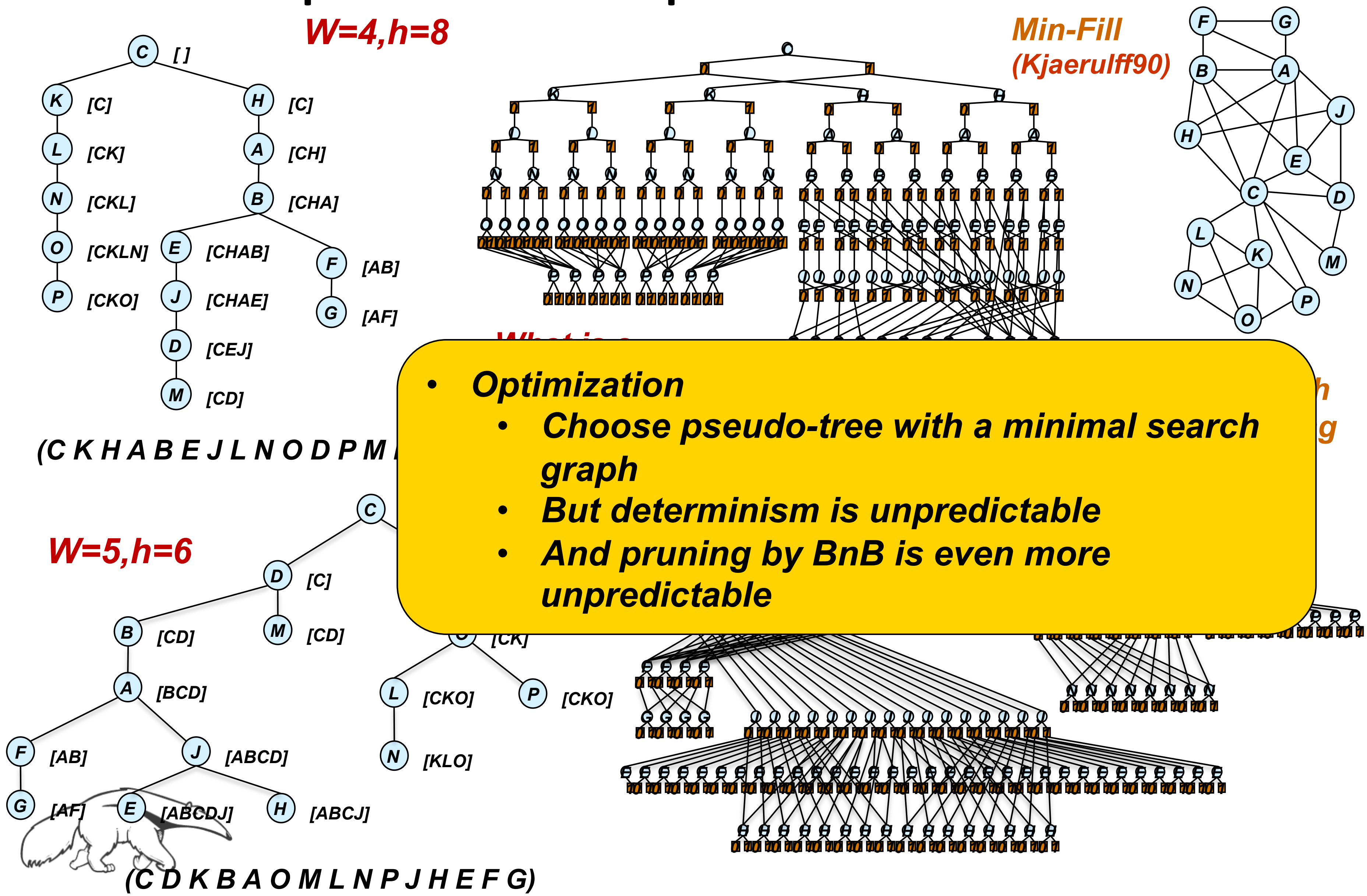


Context minimal AND/OR search graph

18 AND nodes

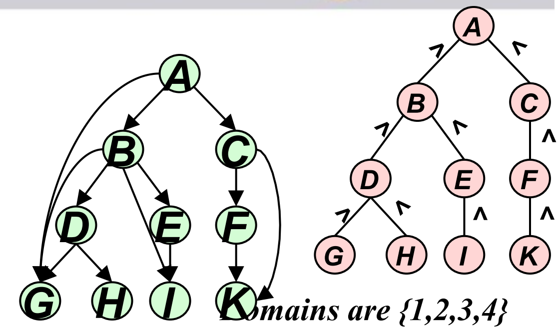
Any query is best computed Over the c-minimal AO space

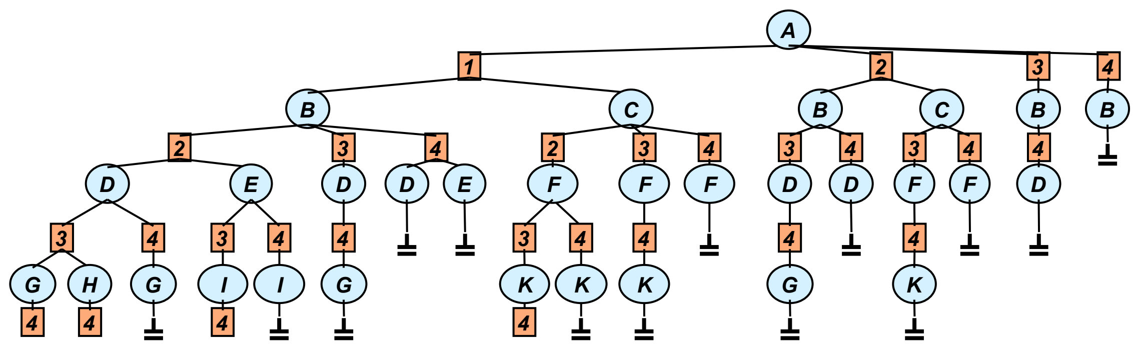
The impact of the pseudo-tree



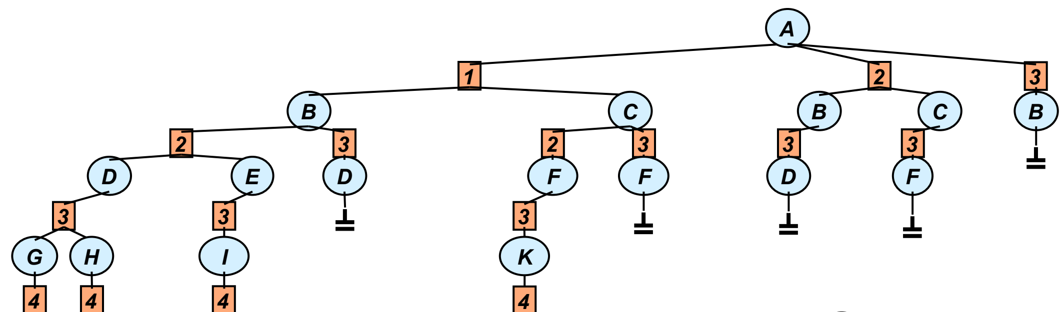
- *Optimization*
  - *Choose pseudo-tree with a minimal search graph*
  - *But determinism is unpredictable*
  - *And pruning by BnB is even more unpredictable*

The Effect of Constraint Propagation

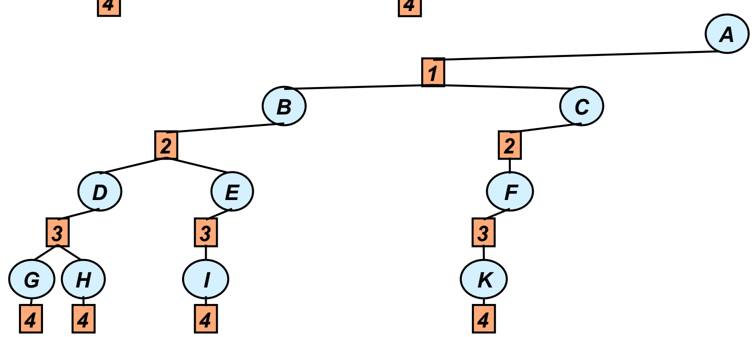




CONSTRAINTS ONLY



FORWARD CHECKING



MAINTAINING ARC CONSISTENCY

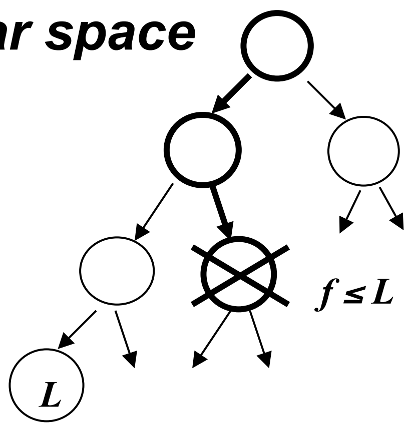
Basic Heuristic Search Schemes

Heuristic function $f(x^p)$ computes a lower bound on the best extension of x^p and can be used to guide a heuristic search algorithm. We focus on:

1. *Branch-and-Bound*

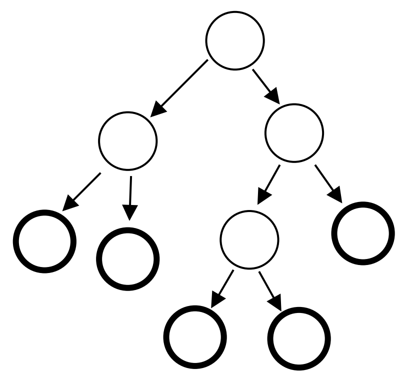
Use heuristic function $f(x^p)$ to prune the depth-first search tree

Linear space



2. *Best-First Search*

Always expand the node with the highest heuristic value $f(x^p)$ needs lots of memory

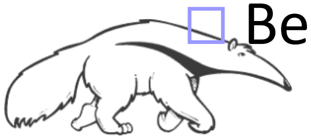


Bounding approximations:

- The mini-bucket scheme
- The cost-shifting or re-parameterization scheme
- Combining the two

Two Bounding Schemes

- Goal: bound $\min_x \sum_i f_i(x)$ or $\max_x \prod_i f_i(x)$
- **Mini-bucket; Node duplication control:**
  - Mini-bucket scheme: (Dechter and Rish 1997,2003, Kask and Dechter, 1999, Rollon and Dechter 2010)
- **Reparameterization schemes:**
  - Soft arc-consistency (Bistareli, 2000, Sciex 2000)
  - Linear relaxation/ Dual-decomposition: (Globerson and Jaakkola 2007, Sontag, Globerson and Jaakkola, 2010, Kovalevsky et al. 1975)
  - Belief Propagation can be viewed as re-parameterization



Mini-bucket Approximation
(Dechter and Rish, 1997, 2003)

Split a bucket into mini-buckets =>bound complexity

$$\text{bucket}(X) = \underbrace{\{ h_1, \ldots, h_r, h_{r+1}, \ldots, h_n \}}_{h^X = \max_X \prod_{i=1}^{n} h_i}$$

X1 $\rightarrow$ $\underbrace{\{ h_1, \ldots, h_r \}}$ $\qquad$ $\underbrace{\{ h_{r+1}, \ldots, h_n \}}$ $\leftarrow$ X

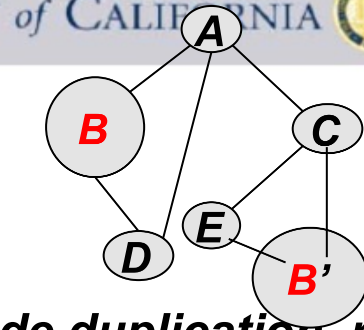
$$g^X = \left(\max_X \prod_{i=1}^{r} h_i \right) \cdot \left(\max_X \prod_{i=r+1}^{n} h_i \right)$$

$$\downarrow$$

$$\boxed{h^X \le g^X}$$

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

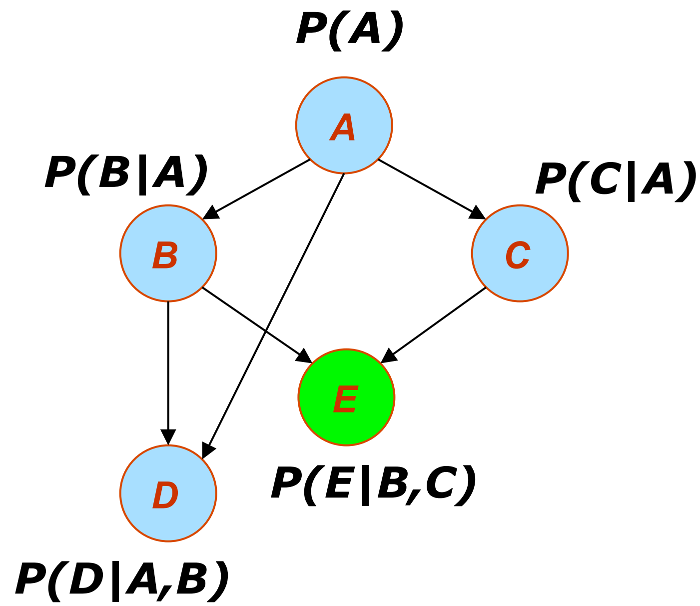
Mini-Bucket Elimination



Node duplication, renaming

	$\max_B \prod$	$\max_B \prod$
Bucket B	**P(E|B,C)**	**P(B|A) P(D|A,B)**
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	**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

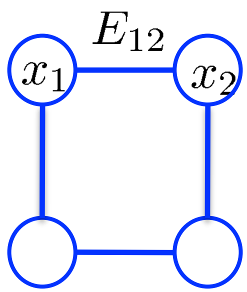
Properties of Mini-Bucket Eliminaton

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

Outline

- Graphical models and the MAP task
- Bounding approximations
  - The mini-bucket scheme
  - The cost-shifting or re-parameterization scheme
  - Combining the two
- New Algorithms combinations
- Experiments

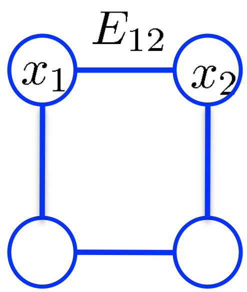
Tightening bounds via cost-shifting



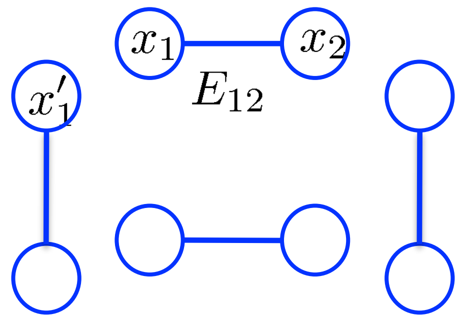
Original

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

Tightening bounds via cost-shifting



Original

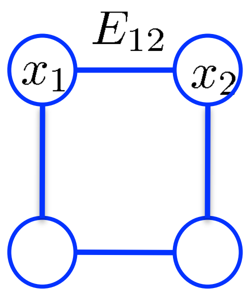


Decomposition

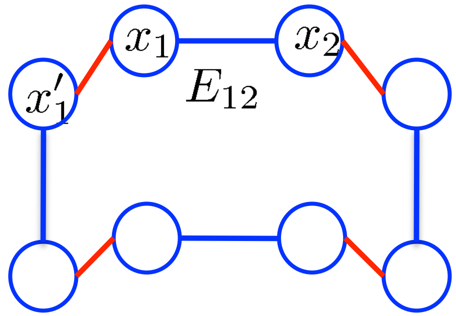
$$\max_{\underline{x}} \sum_{ij} E_{ij}(x_i, x_j) \le \sum_{ij} \max_{\underline{x}} E_{ij}(x_i, x_j)$$

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

Decomposition view



Original

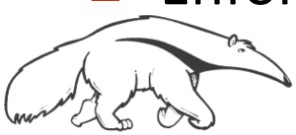


Decomposition

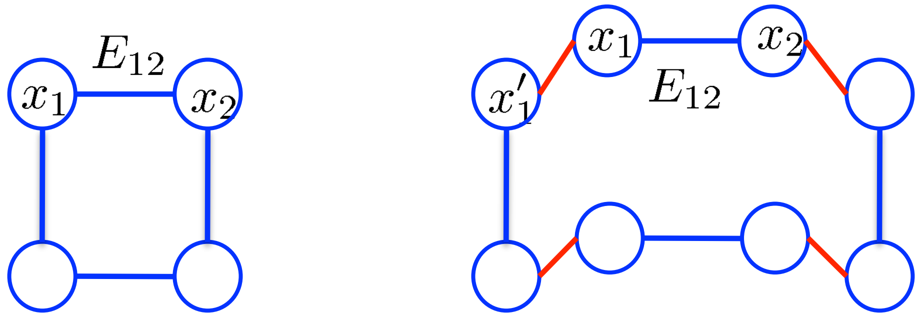
$$\forall i \sum_j \lambda_{ij}(x_i) = 0$$

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

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



Decomposition view



Original ***Decomposition***

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

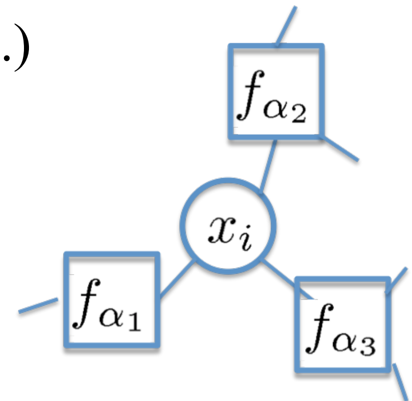
$$\max_{\underline{x}} \sum_{ij} E_{ij}(x_i, x_j) \le \min_{\lambda} \sum_{ij} \max_{\underline{x}} E_{ij}(x_i, x_j) + \lambda_{ij}(x_i) + \lambda_{ji}(x_j)$$

Same bound by different names

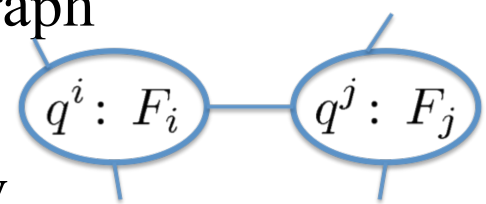
- Dual decomposition (Komodakis et al. 2007)
- TRW, MPLP (Wainwright et al. 2005; Globerson & Jaakkola 2007)
- Soft arc consistency (Cooper & Schiex 2004)

Various Update Schemes

- Can use any decomposition updates
  - (message passing, subgradient, augmented, etc.)
- FGLP: Update the original factors



- JGLP: Update clique function of the join graph

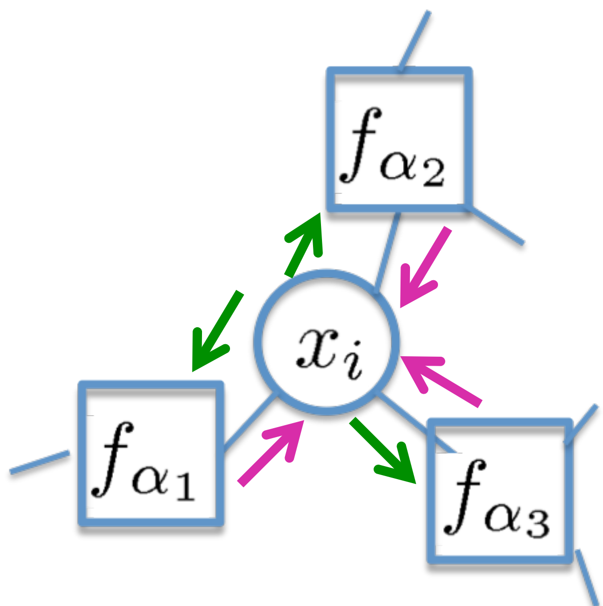


- MBE-MM Update within each bucket only

Factor graph Linear Programming

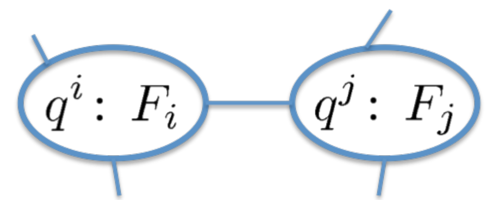
- Update the original factors (FGLP)
  - Tighten all factors over over x_i simultaneously
  - Compute max-marginals $\forall \alpha,\ \gamma_\alpha(x_i) = \max_{x_\alpha \setminus x_i} f_\alpha$
  - & update:

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



Mini-bucket + fixed-point updates

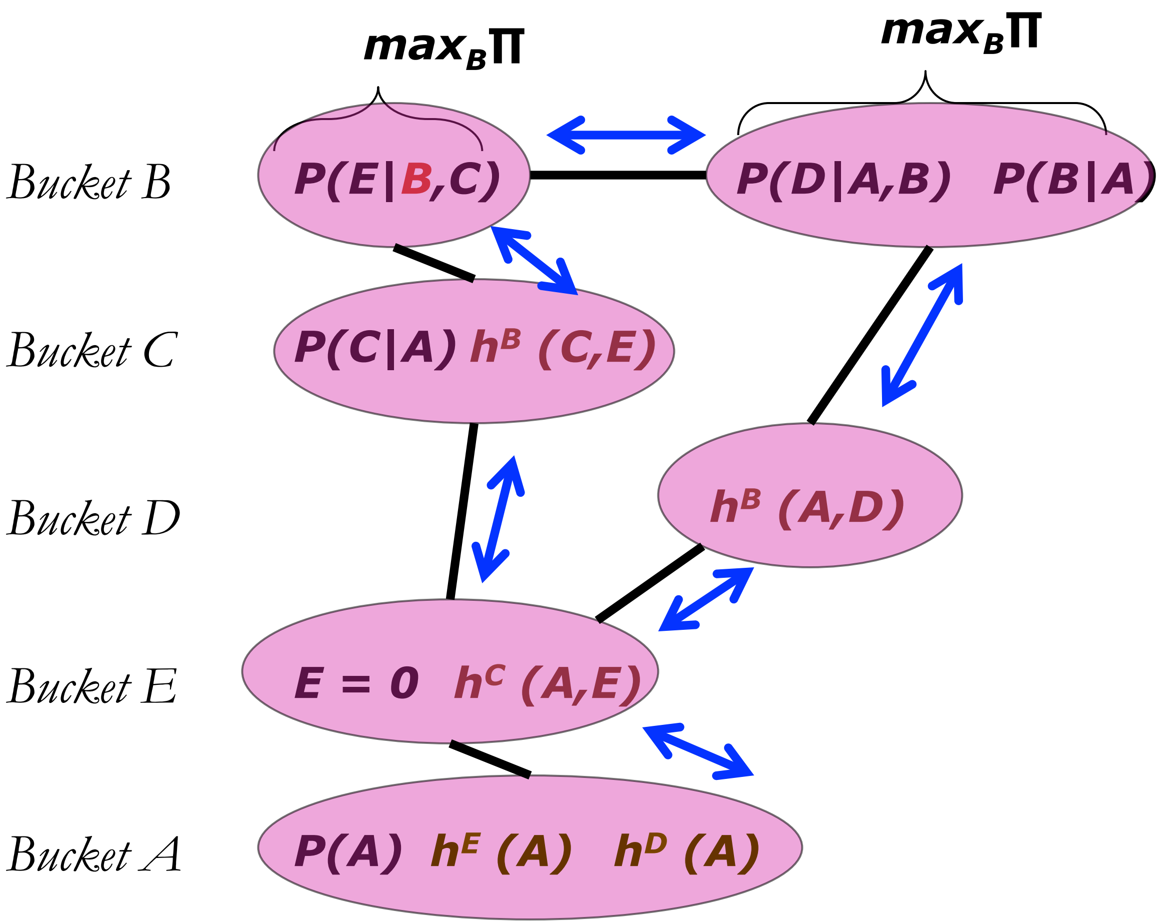
- **JGLP:** Update clique function of the join graph
  - Use MBE to generate the join graph
  - Define function F_i for each clique (mini-bucket) q^i
  - Update each edge over separator set



$$F_i \leftarrow F_i + \frac{1}{2}\left(\gamma_j(x_s) - \gamma_i(x_s)\right)$$

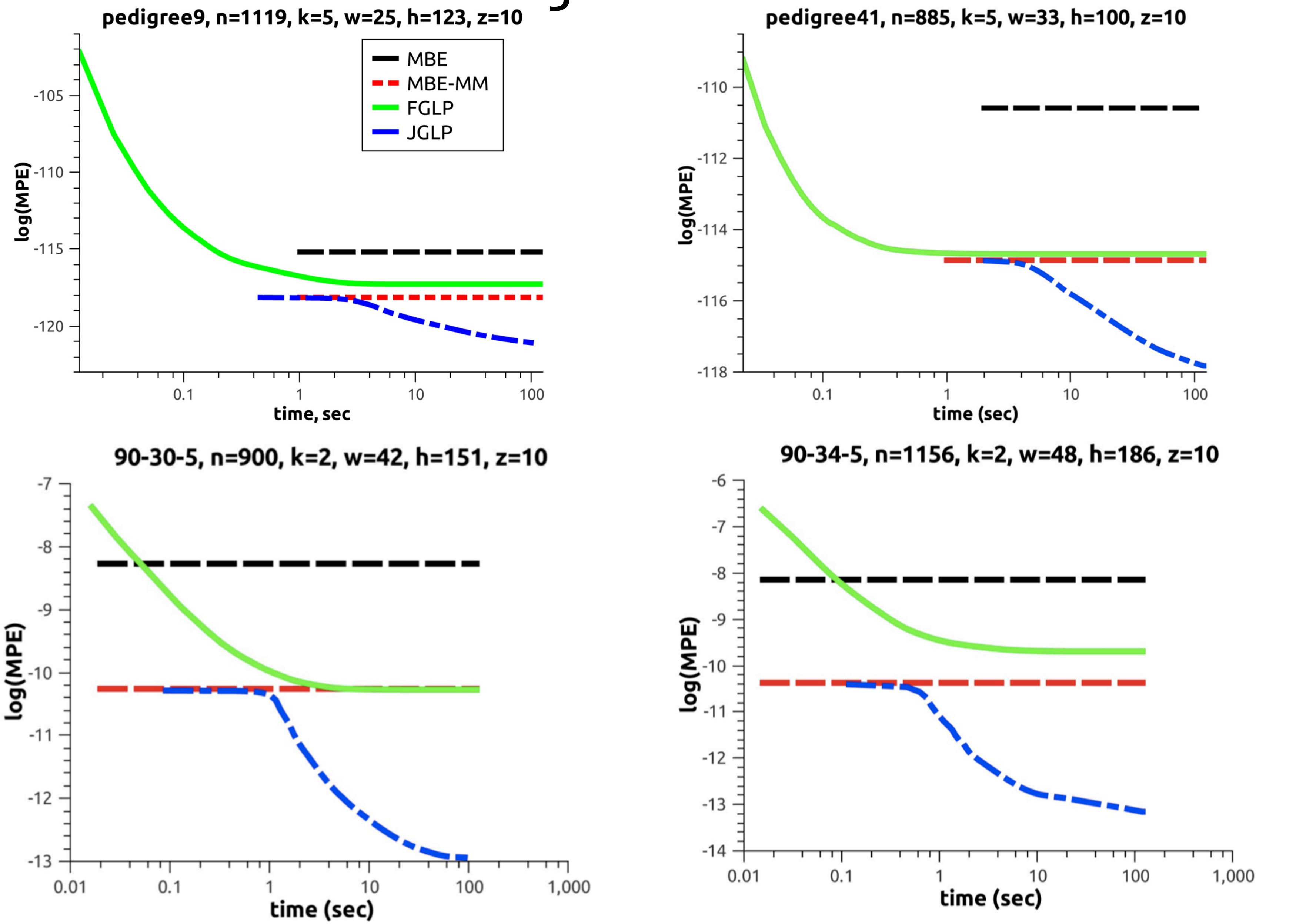
- **MBE-MM:** Mini-bucket with moment matching
  - Apply cost-shifting within each bucket only

Join Graph Linear Programming (JGLP(i))



MB defines
A Join Graph

Iterative tightening as bounding schemes



Anytime AND/OR Branch and Bound (BRAOBB)

+ MBE/JGLP(i)

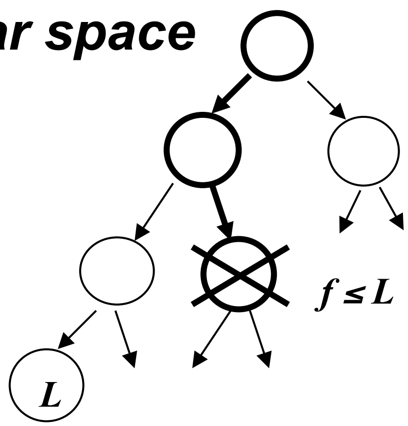
Basic Heuristic Search Schemes

Heuristic function $f(x^p)$ computes a lower bound on the best extension of x^p and can be used to guide a heuristic search algorithm. We focus on:

1. Branch-and-Bound

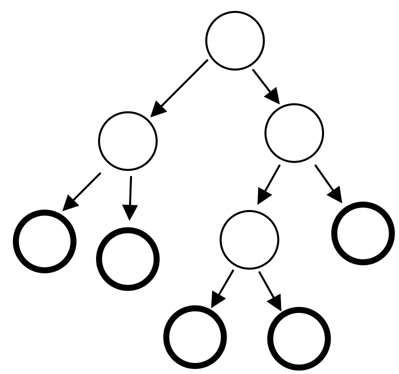
Use heuristic function $f(x^p)$ to prune the depth-first search tree

Linear space



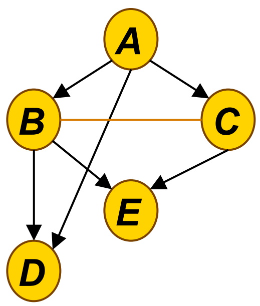
2. Best-First Search

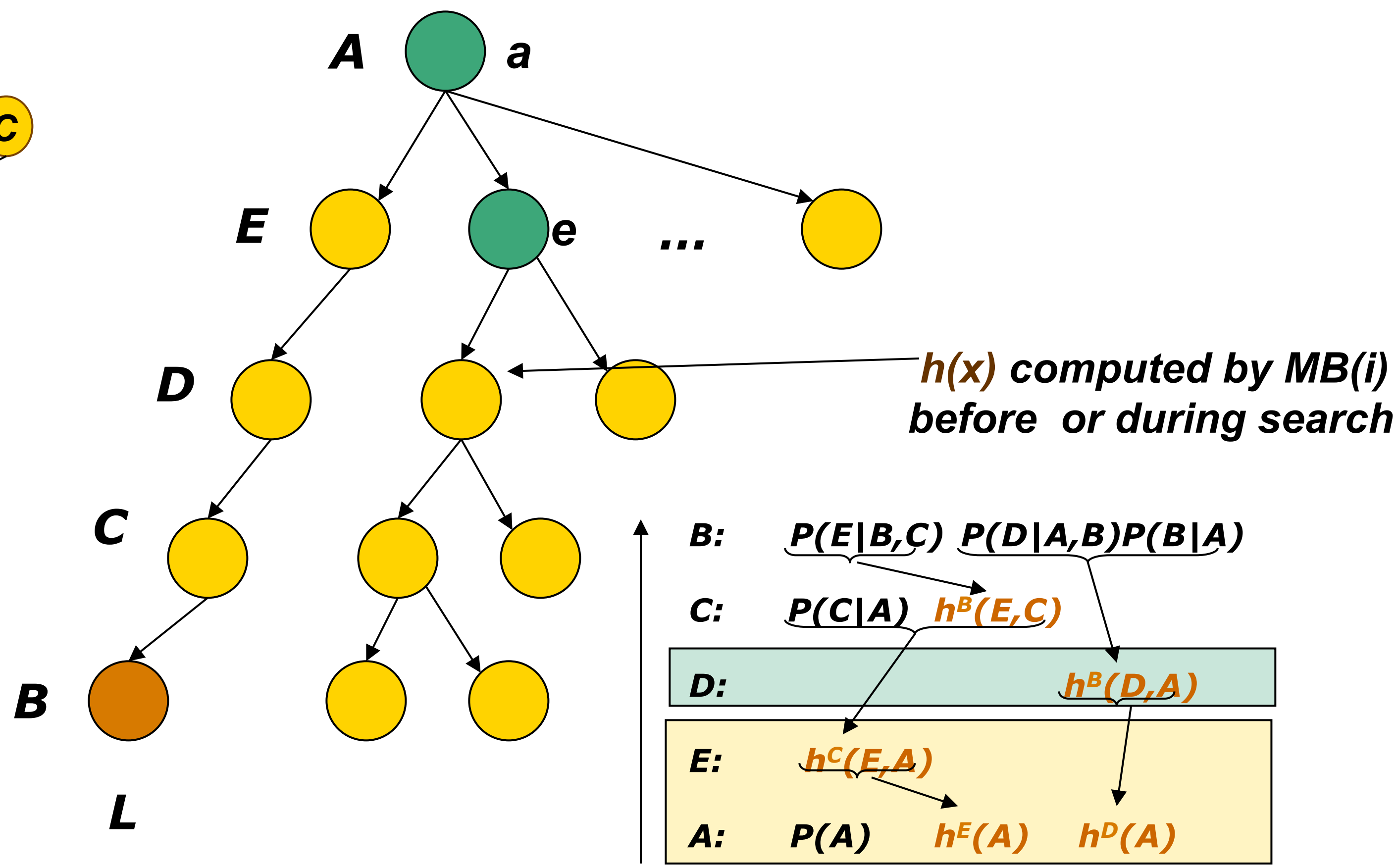
Always expand the node with the highest heuristic value $f(x^p)$ needs lots of memory



Mini-bucket Heuristics for BB search

(Kask and dechterAIJ, 2001, Kask, Dechter and Marinescu 2004, 2005, 2009, Otten 2012)



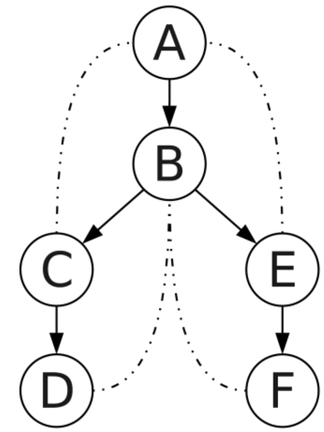


$$f(a,e,D) = P(a)\cdot h^B(D,a)\cdot h^C(e,a)$$

AND/OR Branch-and-Bound

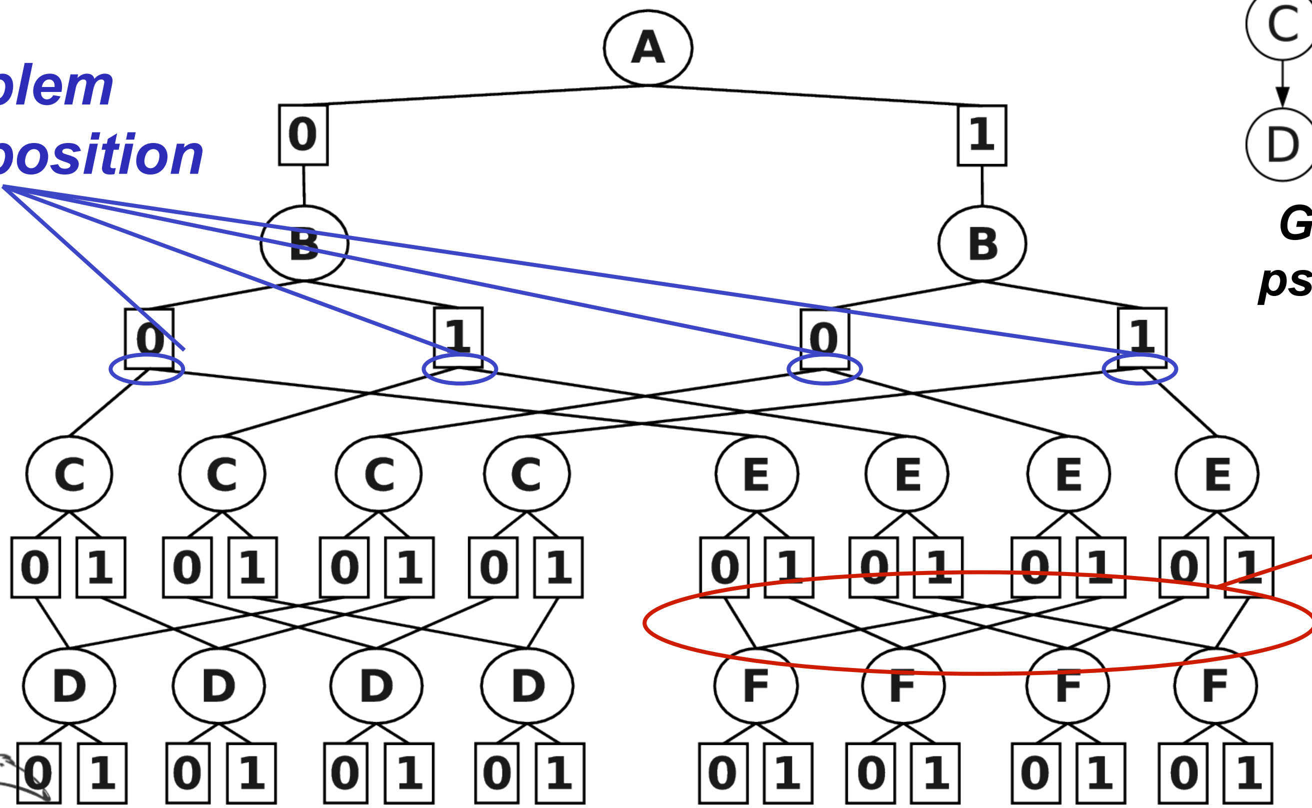
- ***Problem decomposition and caching.***

  Mini-bucket heuristic for pruning.



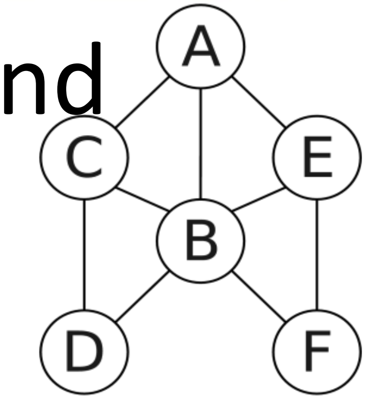
Guided by pseudo tree

Problem decomposition

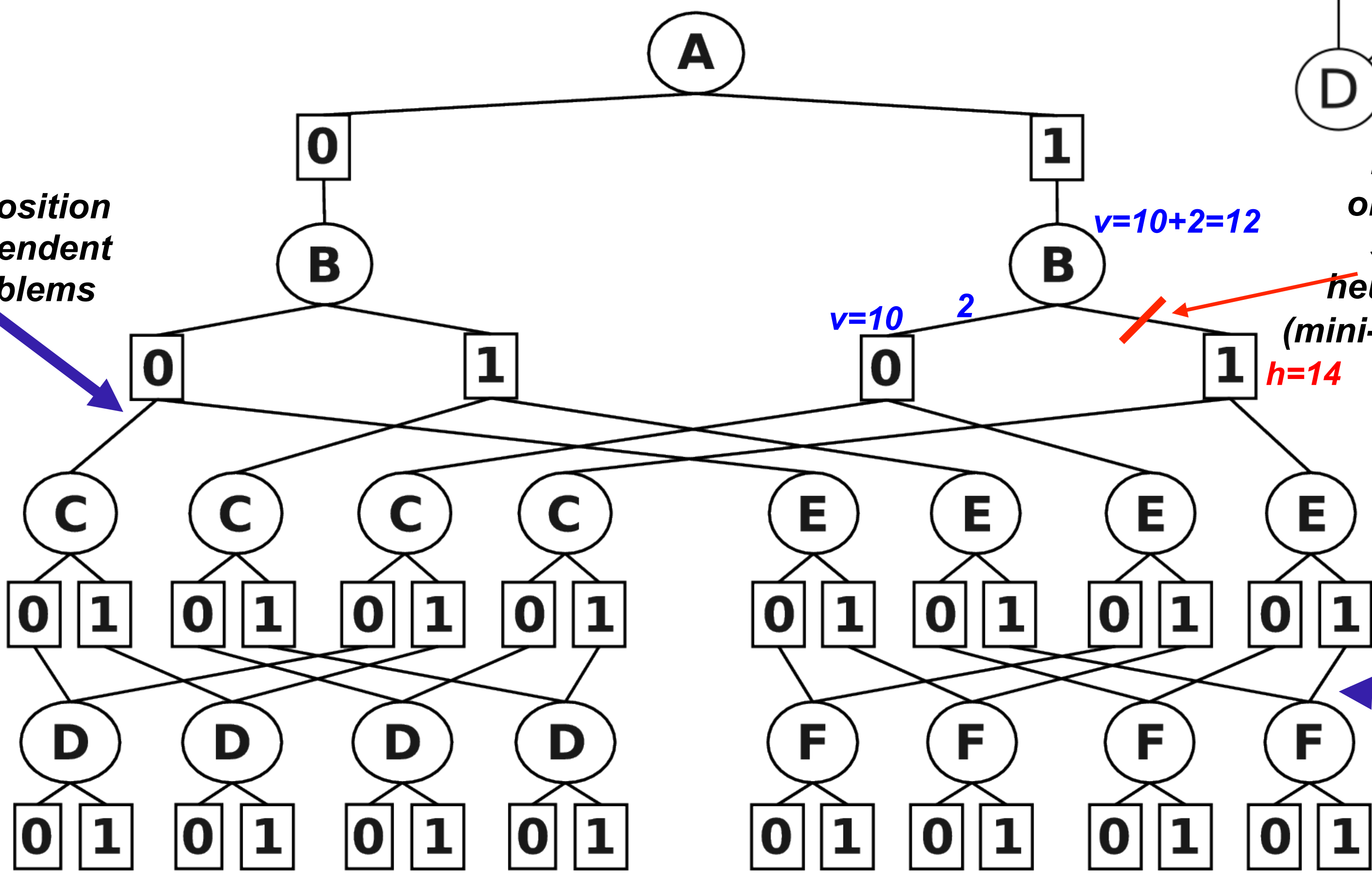


Caching

MAP by AND/OR Branch-and-Bound



Decomposition of independent subproblems



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

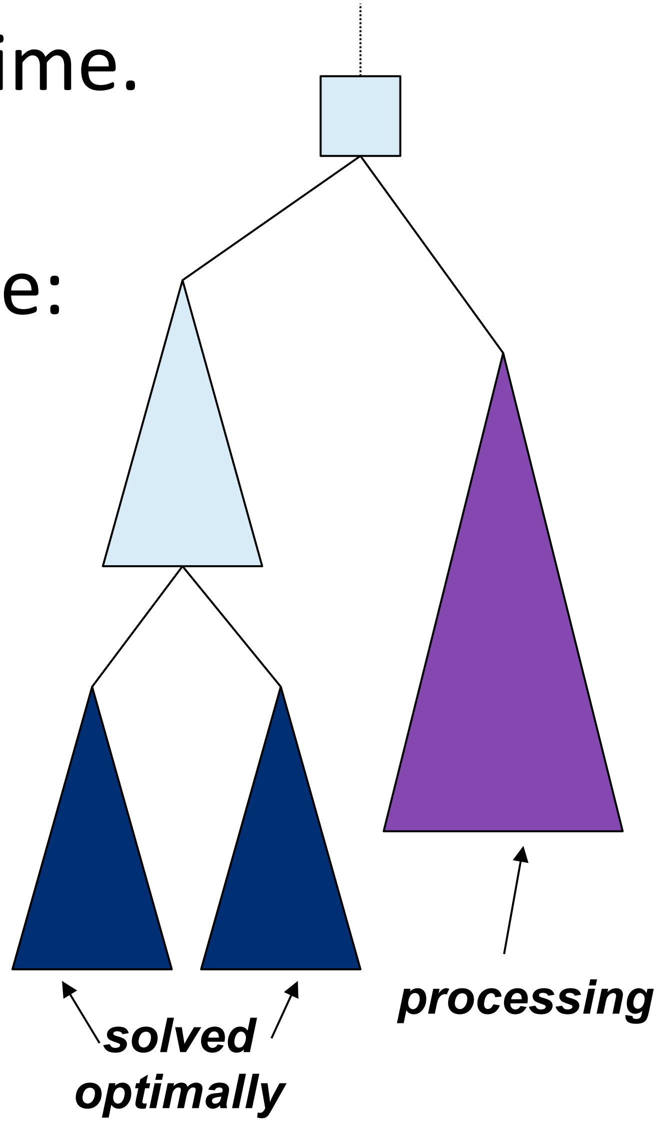
Cache table for F (independent of A)

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

[Marinescu & Dechter, AIJ'09]

Anytime Performance

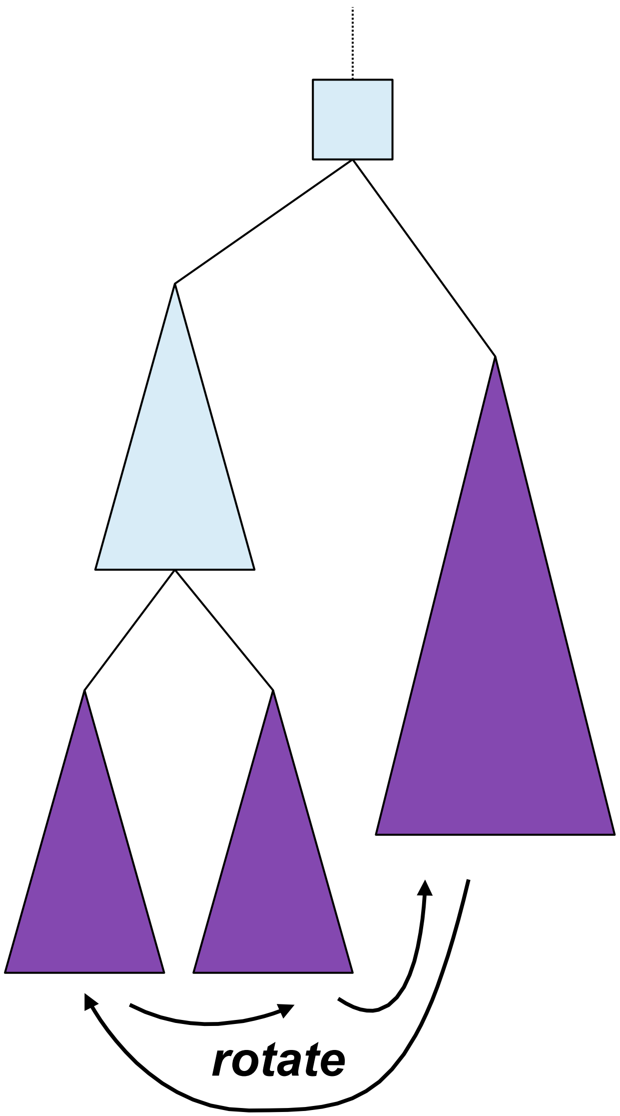
- OR Branch-and-Bound is anytime.
- But AND/OR breaks anytime behavior of depth-first scheme:
  - First anytime solution delayed until last subproblem starts processing.
  - Very bad anytime behavior for unbalanced subproblem spaces.
    - One complex subproblem early on delays everything.



[Otten & Dechter, AICom '12]

Anytime Performance

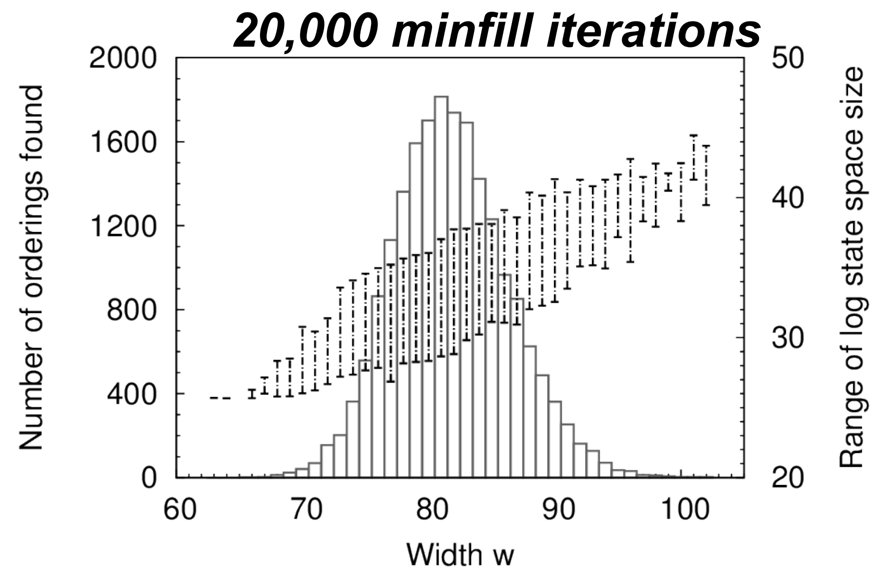
- **Breadth-Rotating AOBB:**
  - Take turns processing subproblems, rotate on:
    - Child subproblem branching.
    - Subproblem solved.
    - Node expansion limit reached.
  - Process each subproblem depth-first.
    - Maintain favorable complexity bounds: linear number of nodes on stack(s).



[Otten & Dechter, AICom '12]

Stochastic Variable Orderings

- AOBB complexity: $O(nk^w)$
  - High variance in width of orderings.
- Our implementation:
  - Minfill heuristic.
  - Random tie-breaking.
  - Allow deviation from heuristic optimum.



  - Highly optimized data structures, early termination.
    - Can do many thousands of iterations.

[Kask, Gelfand, Otten, Dechter, AAAI '11]

Empirical Evaluation: Pascal 2011 Competition Comparing with Gurobi

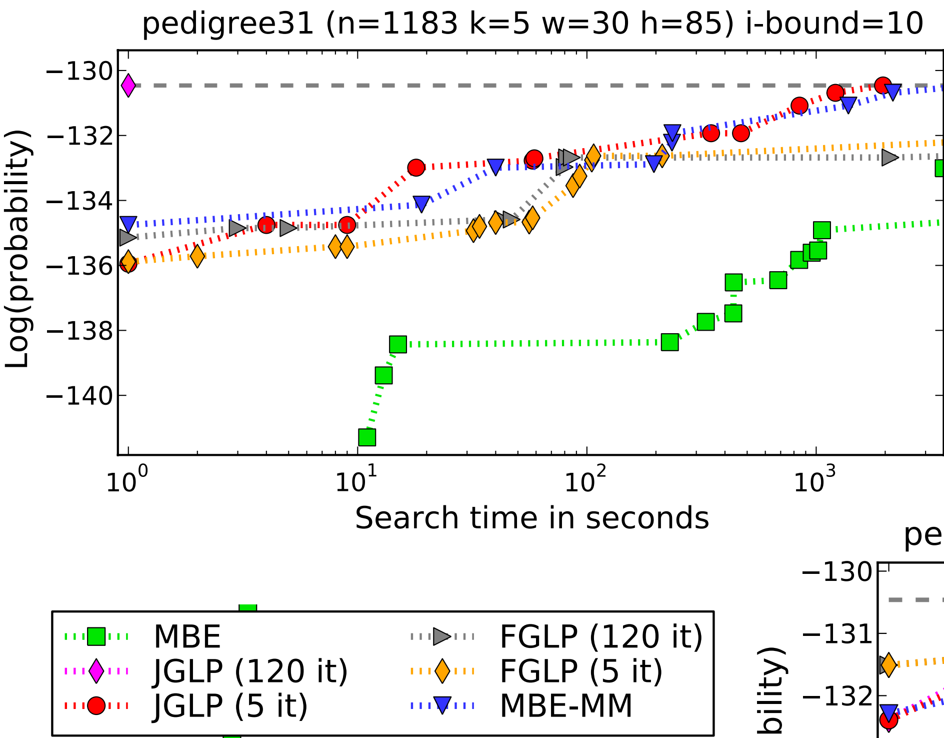
Iterative tightening as heuristic generators

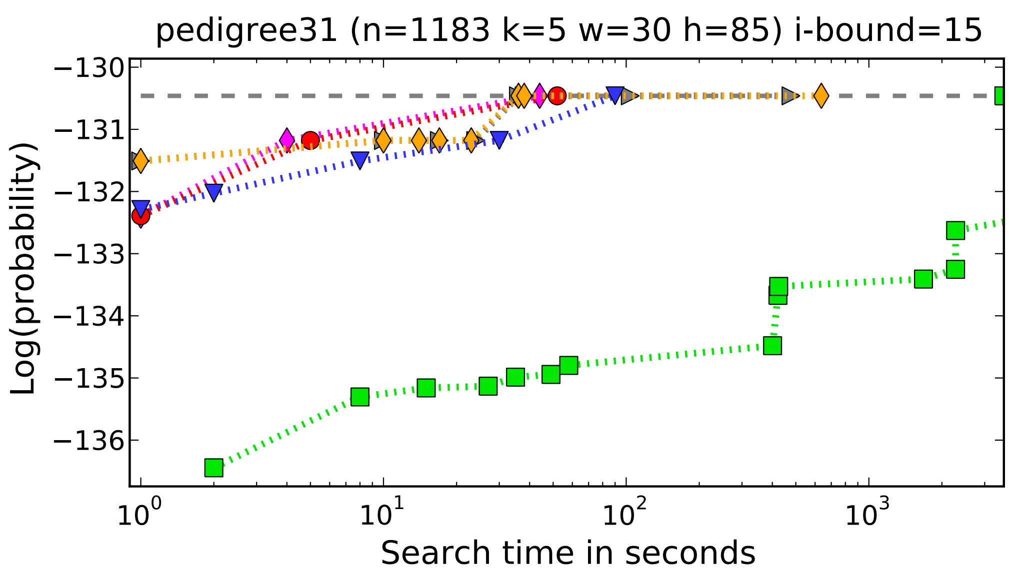
- 4 schemes used:
  - **AOBB-MBE**: AOBB guided by pure MBE heuristics
  - **AOBB-MBE+MM**: AOBB guided by MBE and max- marginal matching
  - **AOBB-FGLP+MBE**: AOBB with heuristics from FGLP followed by MBE
  - **AOBB-JGLP**: AOBB guided by JGLP-produced heuristics

- FGLP, JGLP ran for 30 seconds
- Total search time bound 24 h
- Memory limit 3 Gb
- Mini-bucket z-bounds={10,15,20}

Empirical Evaluation: Haplotype problems

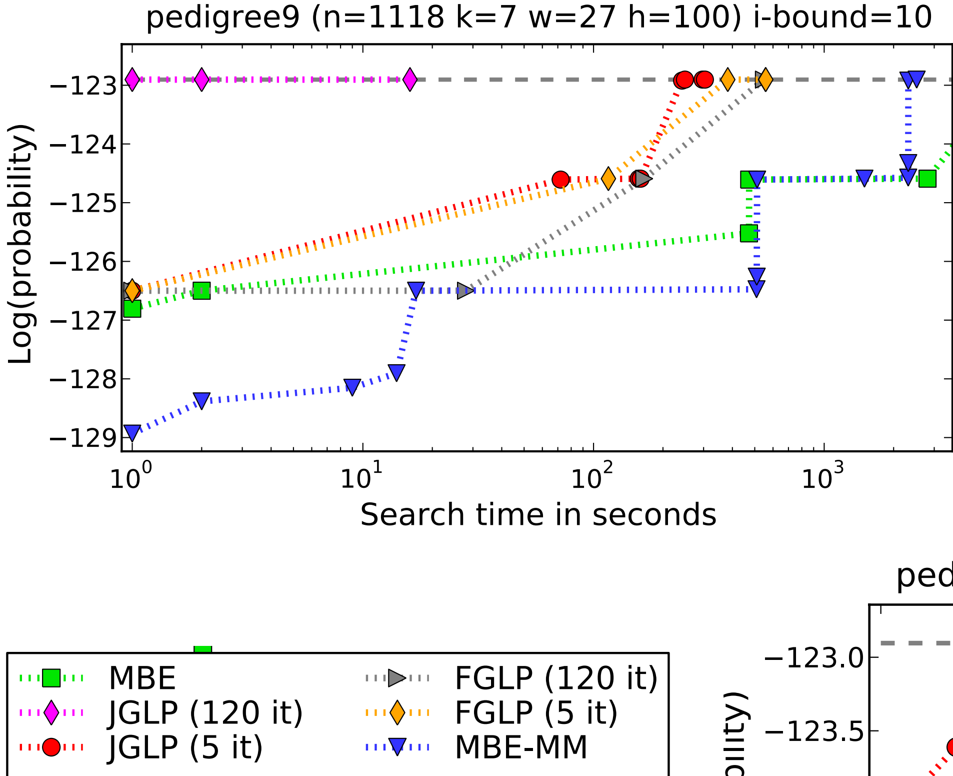
Time bound – 24 h

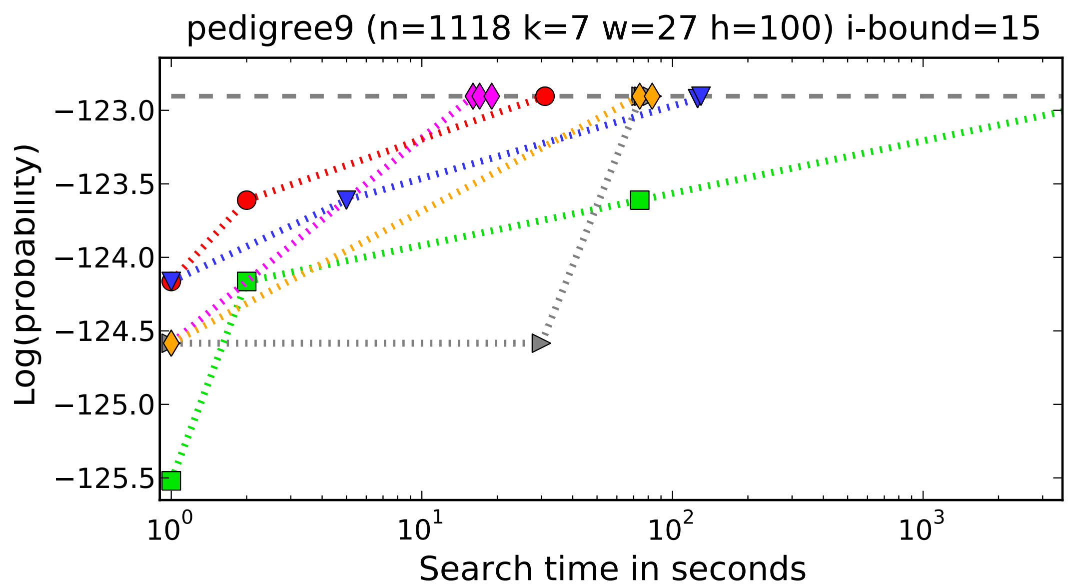




Empirical Evaluation: Haplotype problems

Time bound – 24 h





BREN:ICS INFORMATION AND COMPUTER SCIENCES — UNIVERSITY of CALIFORNIA IRVINE

PASCAL 2011 Inference Challenge

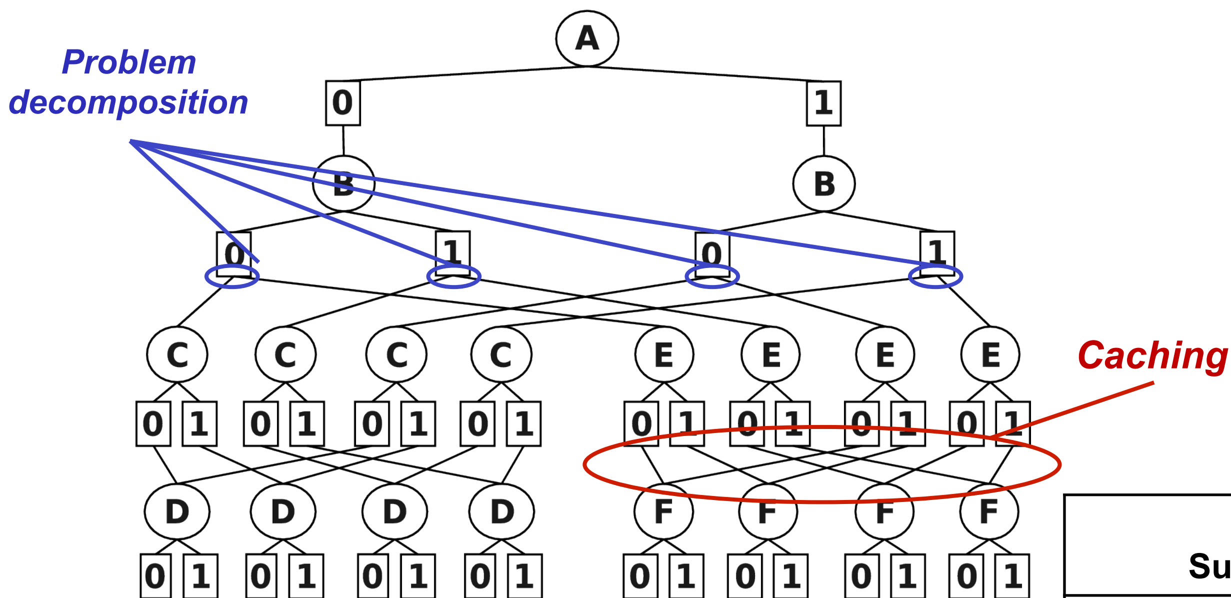
DAOOPT: Improving AND/OR Branch-and-Bound for Graphical Models

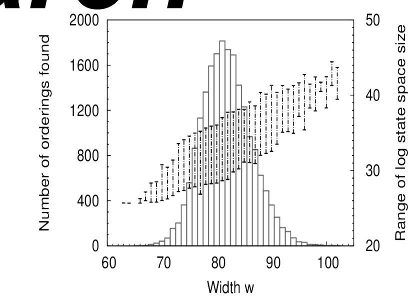
Lars Otten, Alexander Ihler,
Kalev Kask, Rina Dechter

Dept. of Computer Science
University of California, Irvine

ITA, 2/14/2013

State-of-the-art AOBB Search



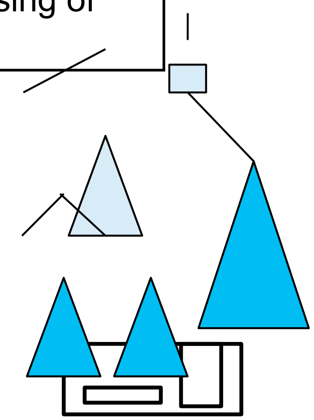


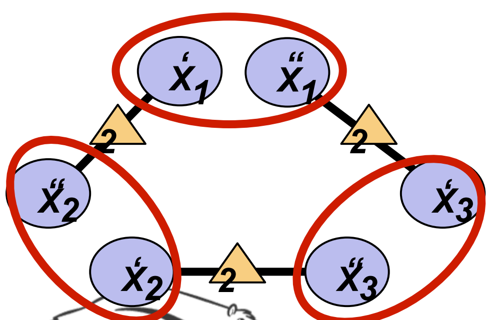
Enhanced Variable Ordering Schemes

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

Breadth-First Subproblem Rotation

Improved anytime perfor-mance through interleaved processing of independent subproblems.





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

Mini-Buckets Cost-shifting (MPLP) Re-parametrization

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

Putting It All Together

- Three time limits: 20 sec, 20 min, 1h.
  - 4 GB memory limit in each case.
- Chose different parameter sets through experimentation.

	20 seconds	20 minutes	1 hour
1. MPLP in input graph	2 sec	30 sec / 500 iter	60 sec / 2000 iter
2. Stochastic Local Search	2x 2 sec	10x 6 sec	20x 10 sec
3. Iterative Variable Ordering	3 sec / 500 iter	60 sec / 10K iter	180 sec / 30K iter
4. MPLP on join graph (JGLP)	2 sec	30 sec / 250 iter	60 sec / 1K iter
5. Mini-buckets + MM	i=15, max.125MB	i=25, max.4GB	i=35, max.4GB
6. Full BRAOBB	To completion or until timeout		

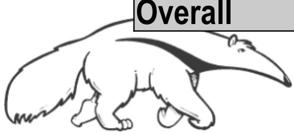
Competing Solvers

- INRA Toulouse, variants of *Toulbar2*:
  - All methods employ soft local arc consistency.
  - dfbbvemcs:
    - Variable elimination preprocessing, depth-first BaB.
  - ficolofo & vns/lds+cp:
    - Initial greedy search or limited discrepancy search, resp.
    - Interleave variable neighborhood search and exact depth-first BaB, limited to varying subspaces.
- Not as competitive, details/authors unknown:
  - Vanilla-MPLP, nutcracker

Competition Results

- 20 sec, 20 min, 1 hour categories
  - Score computed relative to a baseline/BP solution.
  - $E(x) = -\sum \log f_i(x)$, $Score(x^s) = E(x^s) - \min\{E(x^{bp}), E(x^{df}$
  - **1st place** in all three categories!

	20 sec			*20 min*			*1 hour*		
Category	*daoopt*	*ficolofo*	*dfbbvemcs*	*daoopt*	*dfbbvecms*	*ficolofo*	*daoopt*	*ficolofo*	*vns/lds+cp*
CSP	**-0.9123**	-0.8669	-0.8669	**-0.8739**	-0.7862	-0.7862	**-0.8442**	-0.6958	-0.6975
Deep belief nets	-	-	-	-1.6286	**-1.6342**	**-1.6342**	-5.0470	-5.1707	**-5.1709**
Grids	**-0.3403**	-0.3210	-0.3174	**-0.2437**	-0.2241	-0.2241	**-0.1721**	-0.1590	-0.1589
Image alignment	0.0000	0.0000	0.0000	**-0.0006**	0.0000	**-0.0006**	-0.0006	-0.0006	-0.0006
Medical diagnosis	-0.0028	-0.0046	**-0.0460**	-0.0037	**-0.0043**	**-0.0043**	-0.0041	**-0.0043**	**-0.0043**
Object detection	-4.8201	**-4.8287**	-4.8023	-4.8237	**-4.8743**	**-4.8743**	-1.9368	**-1.9628**	-1.9572
Protein folding	-0.0308	-0.0308	-0.0308	-0.1135	**-0.1187**	**-0.1187**	-0.1146	**-0.1183**	**-0.1183**
Prot/prot inter.	-	-	-	**-0.1341**	-0.1317	-0.1317	-0.1681	**-0.1744**	-0.1735
Relational	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Segmentation	**-0.0300**	**-0.0300**	-0.0298	-0.0300	-0.0300	-0.0300	-0.0338	-0.0338	-0.0338
Overall	**-6.3164**	-6.0819	-6.0518	**-7.8519**	-7.8041	-7.8000	**-8.3214**	-8.3196	-8.3150



Competition: Gurobi vs AOBB on Pascal2 instances (Junkyu and Lam, ongoing)

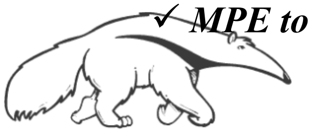
Benchmark

Problem	PEDIGREE	WCSP	Protein Folding
Total #. Instances	22 From UAI'08 Competition	61 From PASCAL2Competition	10 From PASCAL2Competition
Compared Instances	22	15 Exclude 46 Indeterminate Cases*	7 Exclude 3 Indeterminate Cases*
BRAOBB (G.M of Time)	51.07 sec G.M of 20 instances terminated for both	15.96 sec G. M of 13 instances terminated for both	1004.25 sec G. M of 7 instances, terminated for AOBB
GUROBI (G.M of Time)	7.15 sec G.M of 20 instances terminated for both	72.65 sec G. M of 13 instances terminated for both	NA 9/10 Memory Out
BRAOBB VS .GUROBI	0 VS. 22	12 VS. 3	6 Vs. 1 **AOBB won by memory out of Gurobi**

Indeterminate Cases : G.M= geometric mean

- ✓ ***Both Time out, or BRAOBB Time out and Gurobi Memory out (4GB),***
- ✓ ***One of the solver time out earlier than the other's running time***
- ✓ ***MPE to 0/1 ILP Conversion was not available for experiments***

Link to more results



WCSP

	name (n,f,k,s,i,w,h)	Gurobi Time	Tg/Ta	AOBB Time
1	SPOT5 1502 209,412,4,3,6,6,15	1.72	24.57%	7
2	SPOT5 29 82,463,4,2,14,14,24	25.31	76.70%	33
3	SPOT5 404 100,711,4,3,19,19,45	310.58	2218.43%	14
4	SPOT5 503 143,636,4,3,9,9,44	622.49	8892.71%	7
5	SPOT5 54 67,272,4,3,11,11,19	28.15	402.14%	7
6	SPOT5 42 190,1395,4,3,13,26,87	3600 (TO)	NA	289
7	bwt3ac 45,686,11,2,12,16,27	164.4	342.50%	48
8	bwt4ac 179,7110,18,2,7,42,90	(MO)	NA	697

	name (n,f,k,s,i,w,h)	Gurobi Time	Tg/Ta	AOBB Time
9	driverlog01ac 71,619,4,2,9,9,30	7.74	96.75%	8
10	GEOM30a_3 30,82,3,2,6,6,15	42.42	707.00%	6
11	GEOM30a_4 30,82,4,2,6,6,15	6.76	112.67%	6
12	GEOM30a_5 30,82,5,2,6,6,15	6.69	111.50%	6
13	myciel5g_3 47,237,3,2,14,21,24	1012.8	1875.56%	54
14	queen5_5_3 25,161,3,2,15,18,20	2744.57	8072.26%	34
15	queen5_5_4 25,161,4,2,12,18,20	977.24	534.01%	183

BRAOBB : Tout 60 min (MBE-MM, MPLP 2ec, JGLP 2 sec, SLS 2x2 sec, Ordering from Kalev's Code 3sec)
GUROBI : Tout 60 min(Default, Single Processor, Dual Simplex at the Root)
Memory Limit : Both 4 GB

Indeterminate Cases Total 46 (Not presented Above)

- ✓ ***Both Time Out : 5/46***
- ✓ ***AOBB Time Out, Gurobi Memory out 15/46***
- ✓ ***ILP Conversion NA : 26/46***

PROTEIN FOLDING

	name (n,f,k,s,i,w,h)	Gurobi Time	Tg/Ta	AOBB Time
1	pdb1b25 1972,8817,81,2,3,51,178	(MO)	NA	21600 (TO)
2	pdb1d2e 1328,5220,81,2,3,22,136	(MO)	NA	557
3	pdb1fmj 614,2760,81,2,3,35,118	(MO)	NA	21600 (TO)
4	pdb1i24 337,1360,81,2,3,33,58	78.54	21.76%	361
5	pdb1iqc 1040,4042,81,2,3,26,107	(MO)	NA	1801
6	pdb1jmx 739,2943,81,2,3,37,80	(MO)	NA	21600 (TO)
7	pdb1kgn 1060,4715,81,2,3,38,164	(MO)	NA	536
8	pdb1kwh 424,1881,81,2,3,27,93	(MO)	NA	16638
9	pdb1m3y 1364,5037,81,2,3,29,93	(MO)	NA	647
10	pdb1qks 926,3712,81,2,3,36,124	(MO)	NA	493

BRAOBB : Tout 6Hr (MBE-MM, MPLP 60 sec, JGLP 60 sec, SLS 20x10 sec, Ordering from Kalev's Code 3min)
GUROBI : Tout 6Hr (Default, Single Processor, Dual Simplex at the Root)
Memory Limit : Both 4 GB

Indeterminate Cases Total 3

- ***AOBB Time Out, Gurobi Memory Out 3***

Pedigree

	name (n,f,k,s,i,w,h)	Gurobi Time	Tg/Ta	AOBB Time
1	Pedigree1 (298,335,4,5,15,15,60)	0.7	10.00%	7
2	Pedigree13 (888,1078,3,4,20,32,163)	15.8	2.33%	679
3	Pedigree18 (931,1185,5,5,19,19,98)	6.45	53.75%	12
4	Pedigree19 (693,794,5,5,14,24,99)	173.66	NA	1800 (TO)
5	Pedigree20 (387,438,5,4,19,22,73)	11.81	18.75%	63
6	Pedigree23 (309,403,5,4,18,25,60)	2.72	4.77%	57
7	Pedigree25 (993,1290,5,5,20,24,70)	3.1	16.32%	19
8	Pedigree30 (1015,1290,5,5,20,20,121)	7.25	55.77%	13
9	Pedigree31 (1006,1184,5,5,18,30,116)	20.32	18.81%	108
10	Pedigree33 (581,799,4,5,20,27,139)	4.12	9.36%	44
11	Pedigree34 (922,1161,5,4,17,30,130)	40.06	41.30%	97

	name (n,f,k,s,i,w,h)	Gurobi Time	Tg/Ta	AOBB Time
12	Pedigree37 (726,1033,5,4,13,21,59)	3.09	11.44%	27
13	Pedigree38 (581,725,5,4,12,16,62)	5.99	13.31%	45
14	Pedigree39 (953,1273,5,4,20,20,78)	6.01	30.05%	20
15	Pedigree40 (842,1031,7,5,14,28,160)	221.15	NA	1800 (TO)
16	Pedigree41 (885,1063,5,5,18,32,113)	24.5	8.75%	280
17	Pedigree42 (390,449,5,4,15,23,60)	2.36	3.58%	66
18	Pedigree44 (644,812,4,5,20,26,72)	9.83	18.90%	52
19	Pedigree50 (478,515,6,4,10,17,53)	12.05	48.20%	25
20	Pedigree51 (871,1153,5,4,19,38,114)	14.64	3.30%	443
21	Pedigree7 (867,1069,4,4,19,31,119)	8.01	10.27%	78
22	Pedigree9 (935,1119,7,4,20,26,132)	7.85	25.32%	31

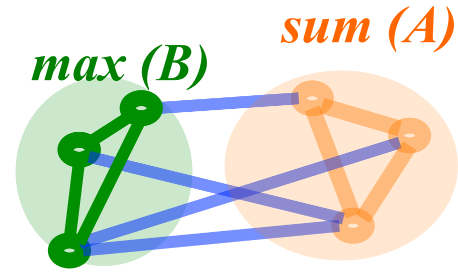
BRAOBB : Tout 30min (MBE-MM, MPLP 2ec, JGLP 2 sec, SLS 2x2 sec, Ordering from Kalev's Code 3sec)
GUROBI : Tout 30min (Default, Single Processor, Dual Simplex at the Root)
Memory Limit : Both 4 GB

Recent work:

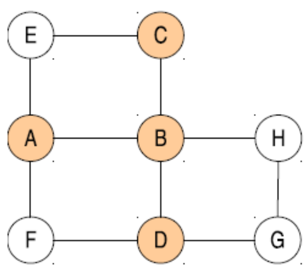
- ***Parallelism AOBB,***
- ***Using External memory***
- ***M-best search***
- ***Weighted Best-first***
- ***Marginal MAP***
- ***STLS- Tree-based SLS***

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



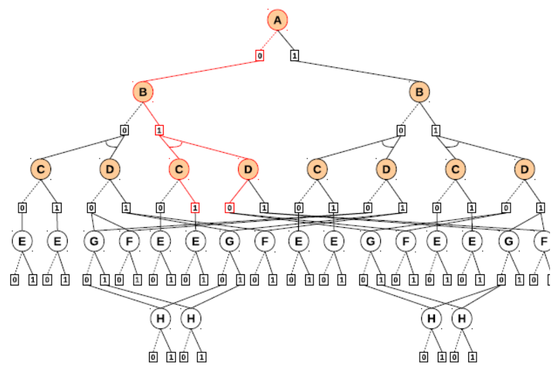


Figure 2: AND/OR search spaces for marginal MAI

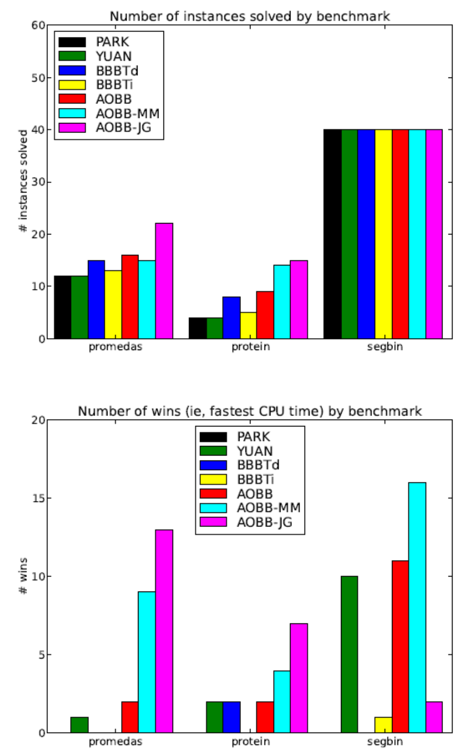


Figure 5: Number of instances solved (top) and number of wins (bottom) by benchmark.

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

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

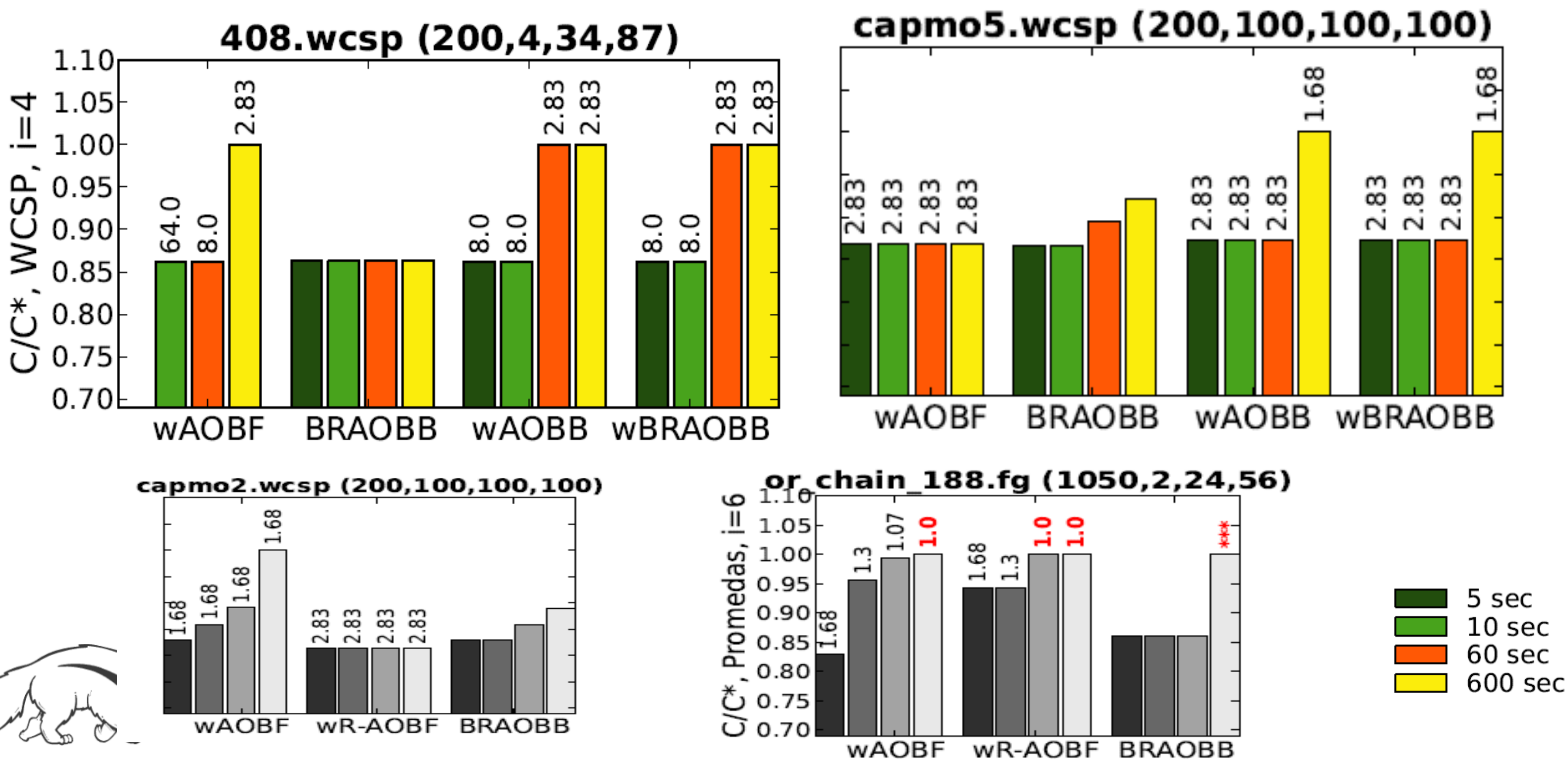
Recent work:

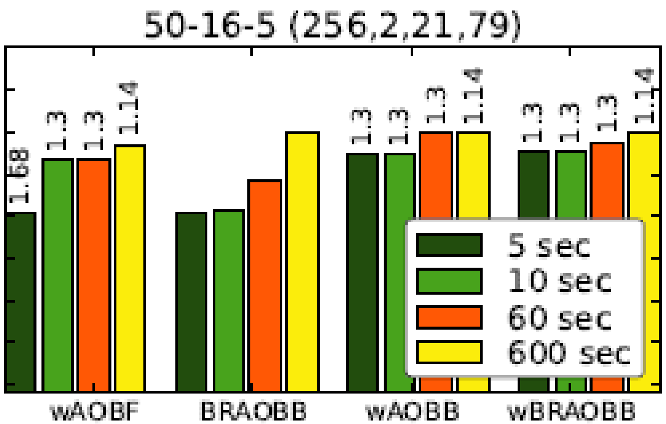
- ***Parallelism AOBB,***
- ***Using External memory***
- ***M-best search***
- ***Weighted Best-first***
- ***Marginal MAP***
- ***STLS- Tree-based SLS***

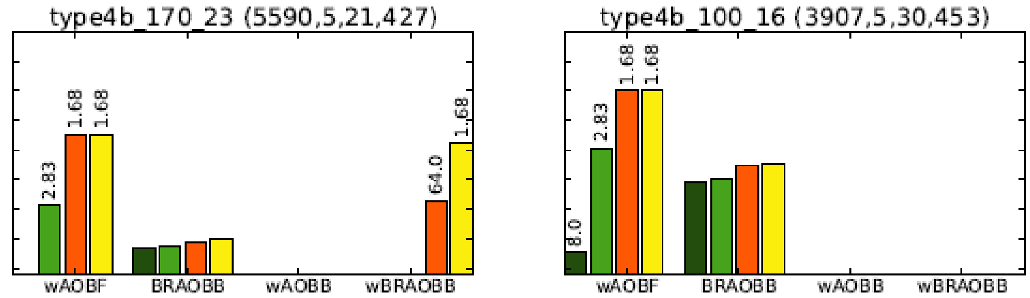
Weighted AND/OR Search

Paper submitted : "EvaluatingWeighted DFS Branch and Bound over Graphical Models" Natalia Flerova, Radu Marinescu, Rina Dechter

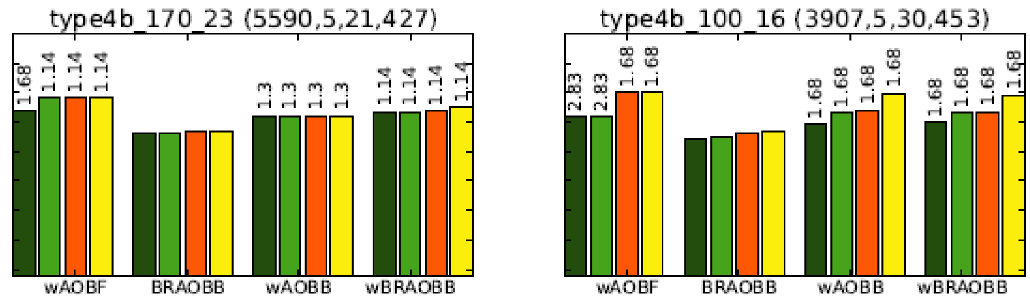
- ***Empirically evaluation proposed algorithms wAOBB and wBRAOBB against Weighted Best-First search (wAOBB) and Breadth-First AND/OR Branch and Bound (BRAOBB)***



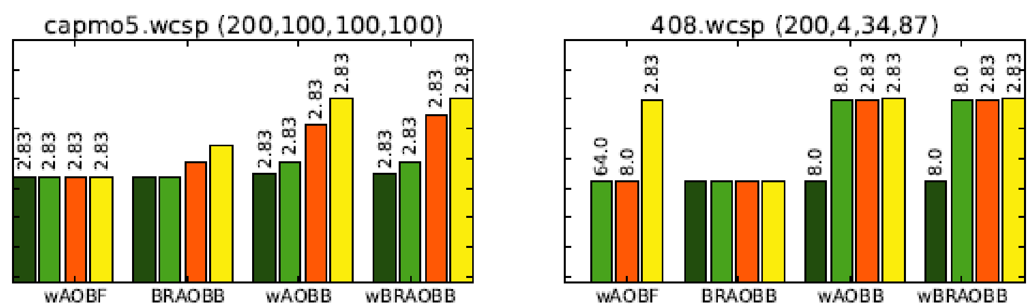

50-16-5 (256,2,21,79)
1.68
1.3
1.3
1.14
1.3
1.3
1.3
1.14
1.3
1.3
1.3
1.14
5 sec
10 sec
60 sec
600 sec
wAOBF
BRAOBB
wAOBB
wBRAOBB


type4b_170_23 (5590,5,21,427)
2.83
1.68
1.68
64.0
1.68
wAOBF
BRAOBB
wAOBB
wBRAOBB
type4b_100_16 (3907,5,30,453)
8.0
2.83
1.68
1.68
wAOBF
BRAOBB
wAOBB
wBRAOBB

(e)


type4b_170_23 (5590,5,21,427)
1.68
1.14
1.14
1.14
1.3
1.3
1.3
1.3
1.14
1.14
1.14
1.14
wAOBF
BRAOBB
wAOBB
wBRAOBB
type4b_100_16 (3907,5,30,453)
2.83
2.83
1.68
1.68
1.68
1.68
1.68
1.68
1.68
1.68
1.68
1.68
wAOBF
BRAOBB
wAOBB
wBRAOBB

(f)


capmo5.wcsp (200,100,100,100)
2.83
2.83
2.83
2.83
2.83
2.83
2.83
2.83
2.83
2.83
2.83
2.83
wAOBF
BRAOBB
wAOBB
wBRAOBB
408.wcsp (200,4,34,87)
64.0
8.0
2.83
8.0
8.0
2.83
2.83
8.0
8.0
2.83
2.83
wAOBF
BRAOBB
wAOBB
wBRAOBB

Conclusion

- Search+bounded-Inference lead to effective anytime schemes and effective approximations, with bounding guarantees.
- Heuristic evaluation function can be improved
- Static/dynamic heuristic, during search with dynamic variable-ordering (tradeoff of time overhead against pruning).
- Portfolio approach

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)*

BREN:ICS INFORMATION AND COMPUTER SCIENCES

UNIVERSITY of CALIFORNIA IRVINE

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