Advances in Combinatorial Optimization Tasks over Graphical Models

Rina Dechter

Bren school of ICS, University of California, Irvine




Joint work with Radu Marinescu, Robert Mateescu and Lars Otten

BGU 2011

Outline

- Graphical models: the primary reasoning principles
- OR Search Trees
- AND/OR Search Trees
- AND/OR Branch-and-Bound and Best-First Search
- Lower Bounding Heuristics
- Experiments
- More recent work

2004-2009

2010-now

Example Constraint Networks

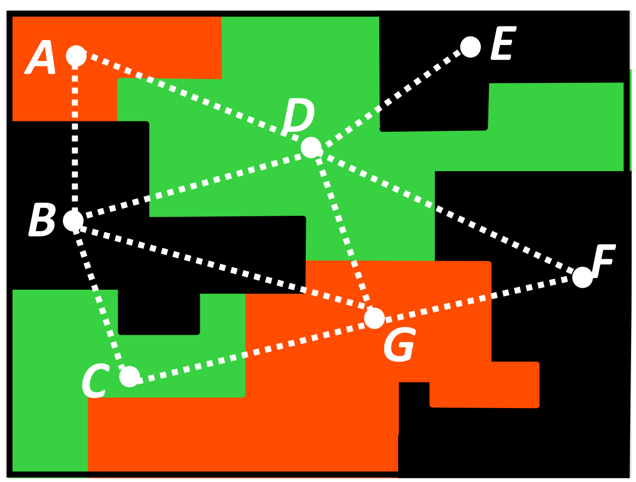
Map coloring

Variables: *countries (A B C etc.)*

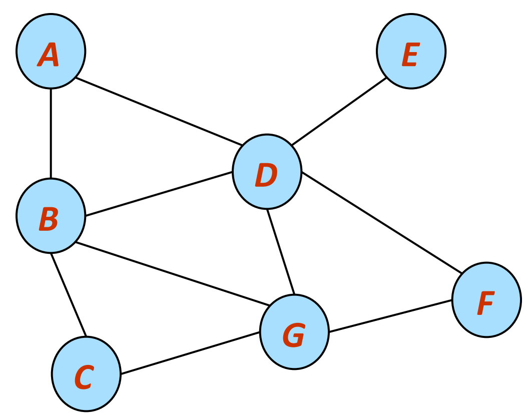
Values: *colors (red green blue)*

Constraints: **A ≠ B, A ≠ D, D ≠ E** , …

A	B
red	green
red	yellow
green	red
green	yellow
yellow	green
yellow	red



Constraint graph



Queries: Find one solution, all solutions, counting

Constraint Optimization Problems

for Graphical Models

A *finite COP* is a triple $R = \langle X, D, F \rangle$ where:

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

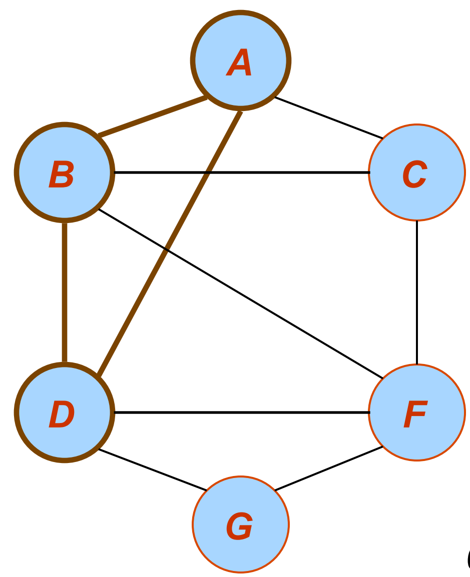
$D = \{D_1, ..., D_n\}$ - domains

$F = \{f_1, ..., f_m\}$ - cost functions

f(A,B,D) has scope {A,B,D}

A	B	D	Cost
1	2	3	3
1	3	2	2
2	1	3	∞
2	3	1	0
3	1	2	5
3	2	1	0

Primal graph =
Variables --> nodes
Functions, Constraints -→ arcs



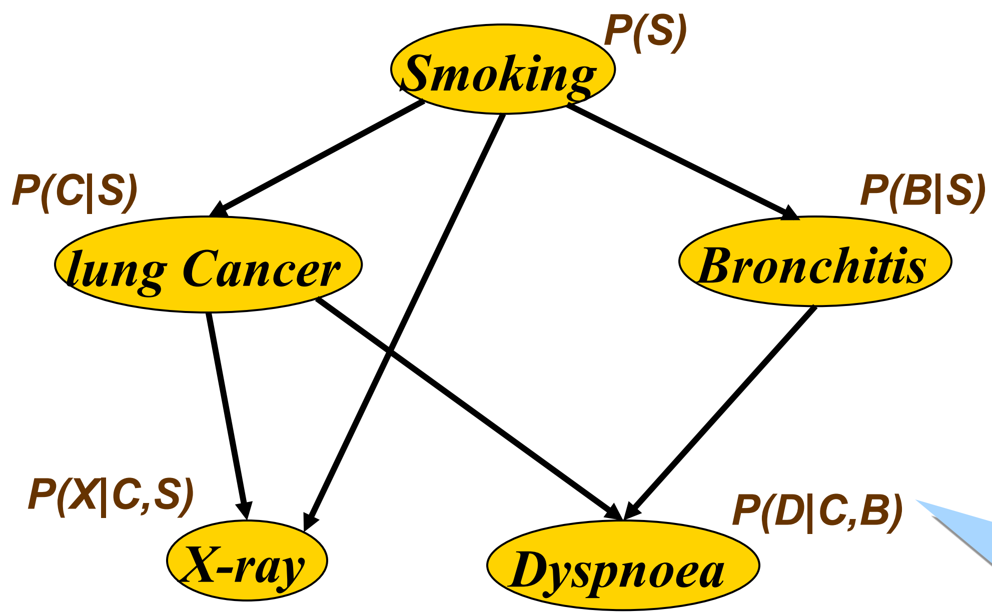
F(a,b,c,d,f,g)= f_1(a,b,d)+f_2(d,f,g)+f_3(b,c,f)

Global Cost Function

$$F(X) = \sum_{i=1}^{m} f_i(X)$$

Bayesian Networks

The MAP/MPE Problem over Bayesian Networks



BN = (G, Θ)

CPD:

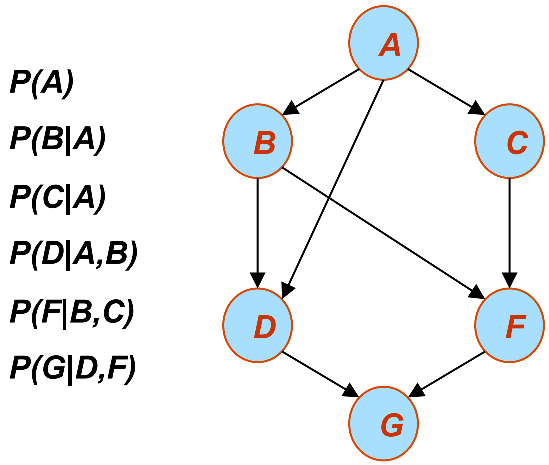
C	B	P(D|C,B)	
0	0	0.1	0.9
0	1	0.7	0.3
1	0	0.8	0.2
1	1	0.9	0.1

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

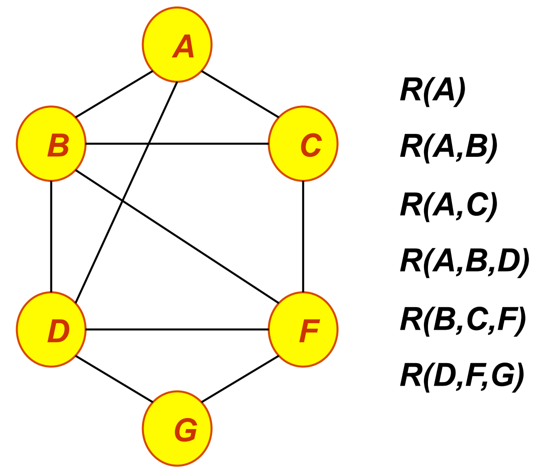
Posterior marginals, Most probable tuple (MPE)

- *MPE/MAP= find argmax $P(S)\cdot P(C|S)\cdot P(B|S)\cdot P(X|C,S)\cdot P(D|C,B)$ =?*

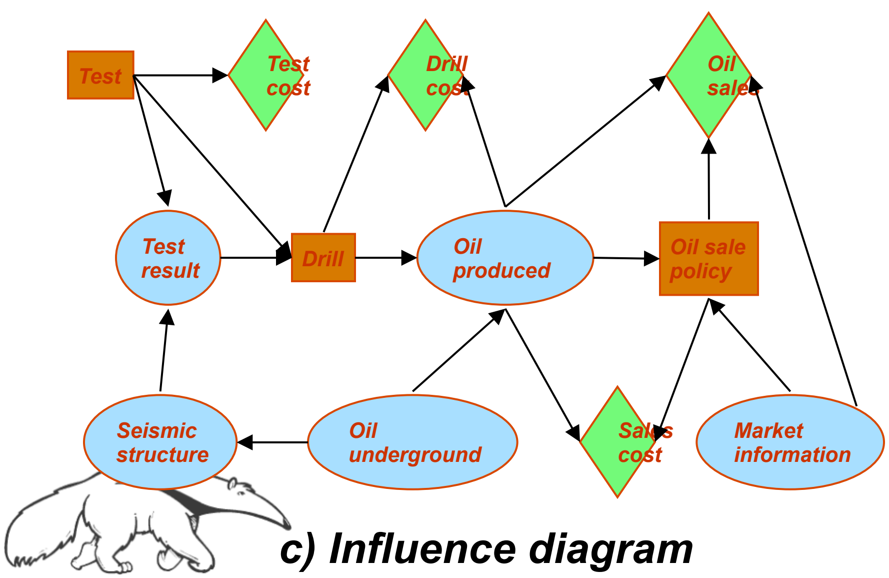
Graphical Models



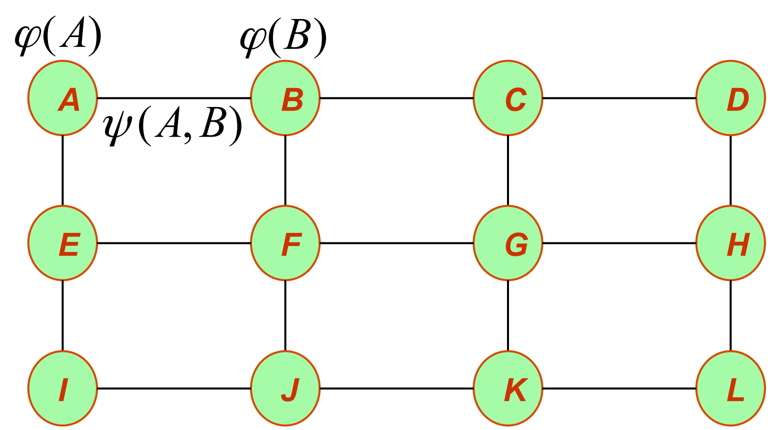
a) Belief network



b) Constraint network



c) Influence diagram



d) Markov network

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
  - elimination (projection)
- Tasks:
  - **Belief updating**: $\Sigma_{X\text{-}y} \prod_j P_i$
  - **MPE**: $\max_X \prod_j P_j$
  - **CSP**: $\prod_X \times_j C_j$
  - **Max-CSP**: $\min_X \Sigma_j F_j$

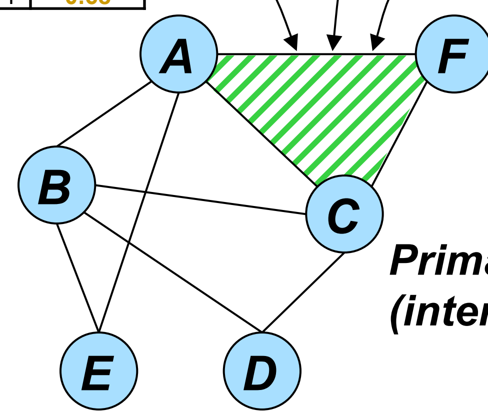
Conditional Probability Table (CPT)

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

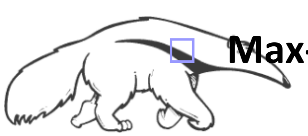
A	C	F
red	green	blue
blue	red	red
blue	blue	green
green	red	blue

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



Primal graph (interaction graph)

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

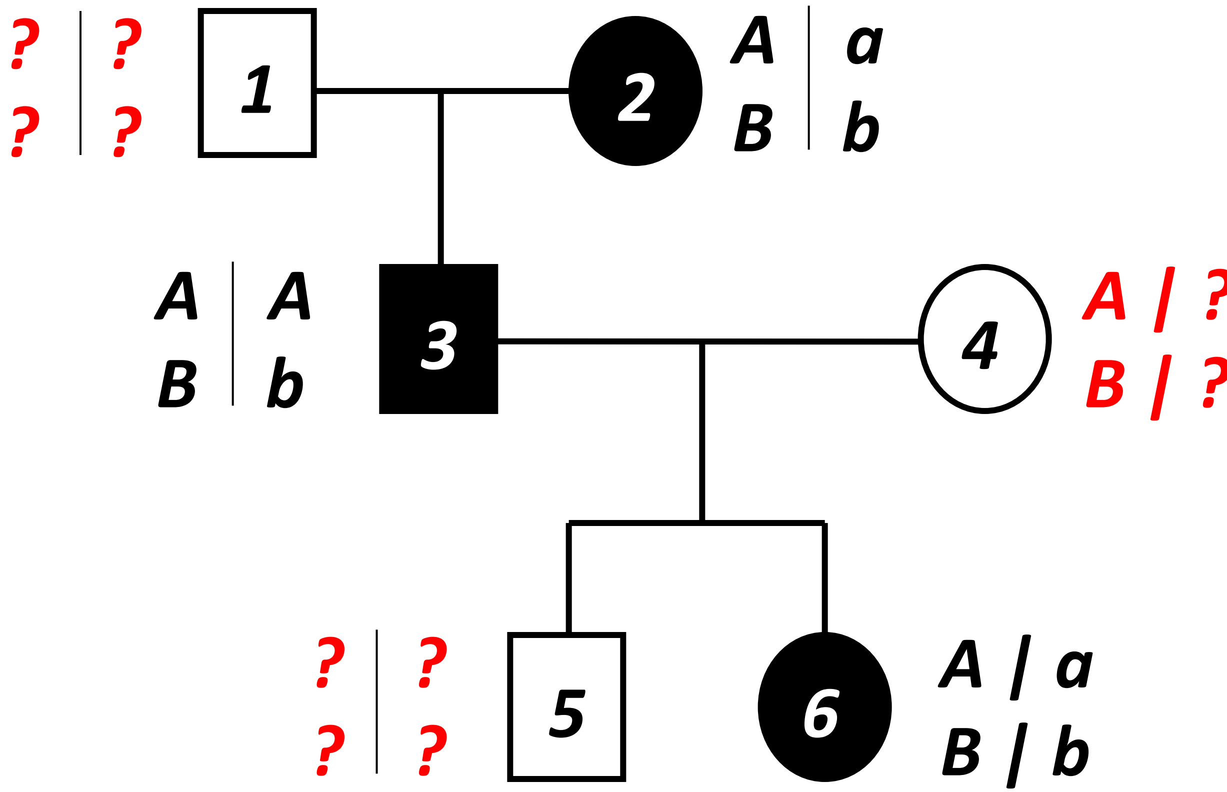


Types of Constraint Optimization

- Valued CSPs, Weighted CSPs, Max-CSPs, Max-SAT
- Most Probable Explanation (MPE/MAP)
- Linear Integer Programs

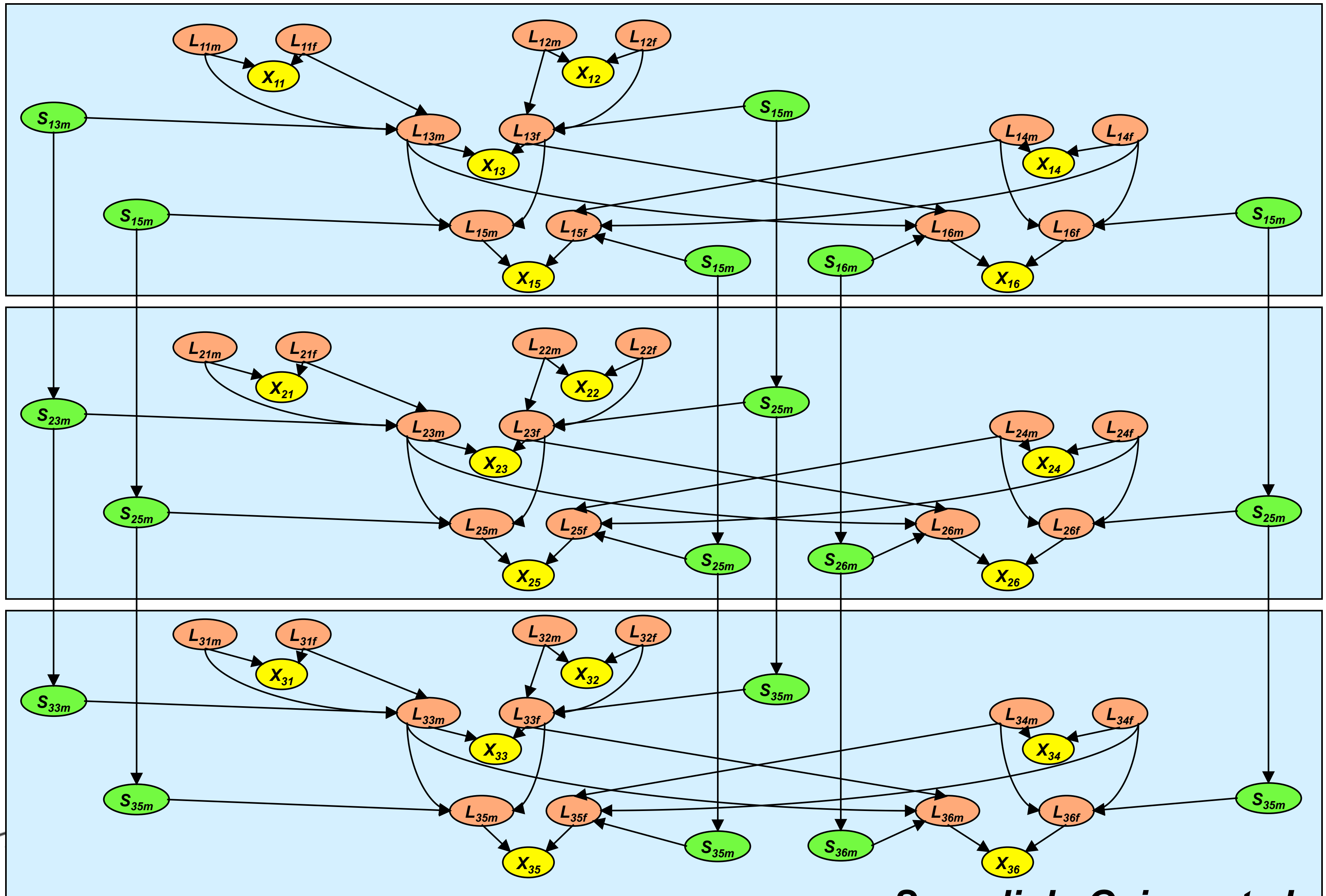
- **Examples**:
  - Problems translated from planning
  - Unit scheduling maintenance
  - Combinatorial auctions
  - Maximum-likelihood haplotypes in linkage

Linkage Analysis



- ***6 individuals***
- ***Haplotype: {2, 3}***
- ***Genotype: {6}***
- ***Unknown***

Pedigree: 6 people, 3 markers



Superlink, Geiger et.al

Solution Techniques, State of the art

AND/OR search

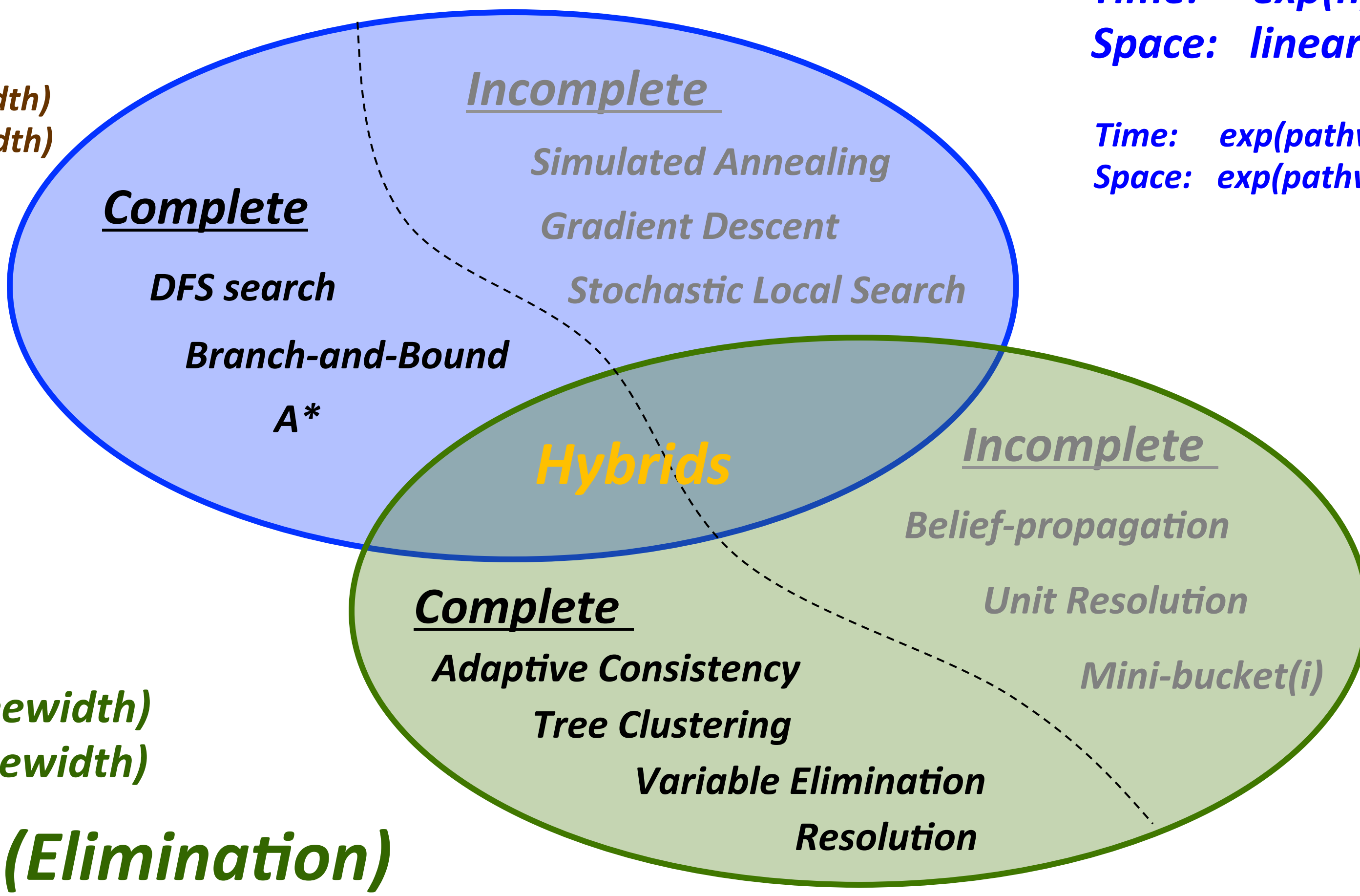
*Time: exp(treewidth * log n)*
Space: linear

Space: exp(treewidth)
Time: exp(treewidth)

Search (Conditioning)

Time: exp(n)
Space: linear

Time: exp(pathwidth)
Space: exp(pathwidth)



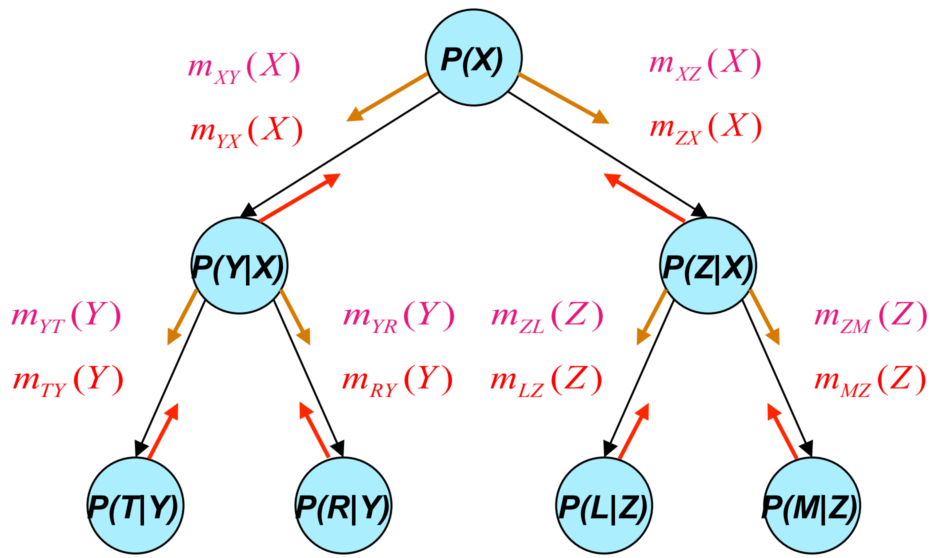
Time: exp(treewidth)
Space: exp(treewidth)

Inference (Elimination)

Tree-solving is Easy

Belief updating (sum-prod)

CSP – consistency (projection-join)



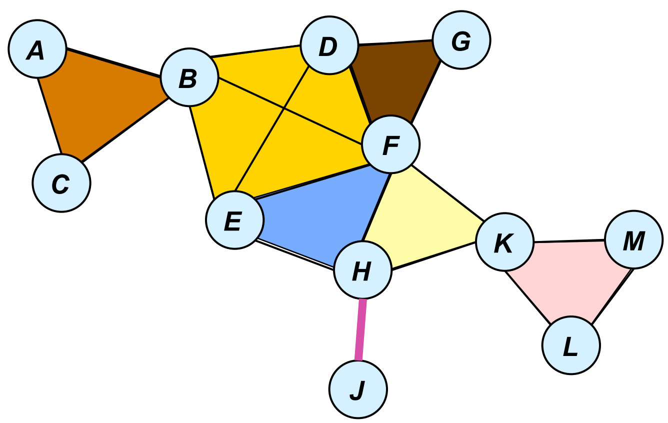
Dynamic Programming

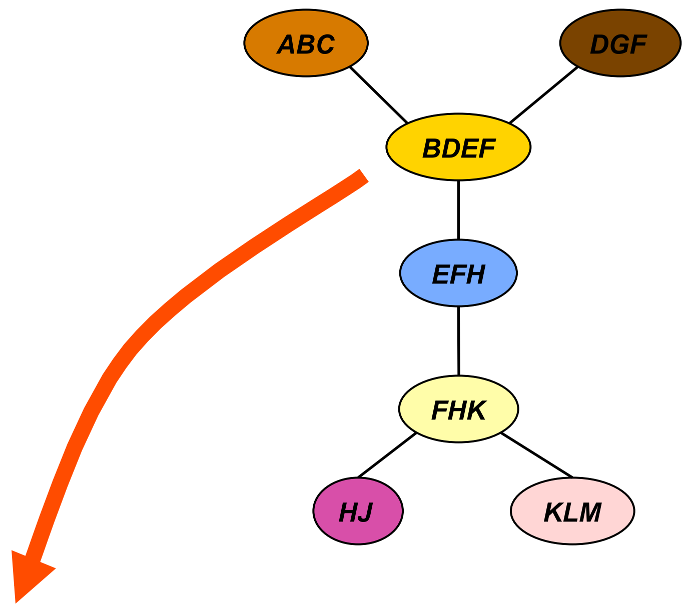
MPE (max-prod)

#CSP (sum-prod)

Trees are processed in linear time and memory
Also Acyclic graphical models

Inference and Treewidth



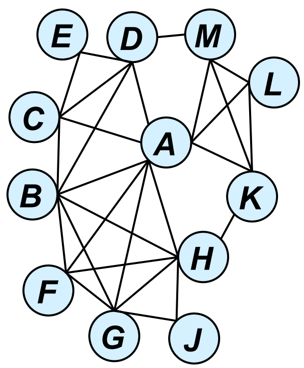


Inference algorithm:
Time: *exp(tree-width+1)*
Space: *exp(separator-width)*

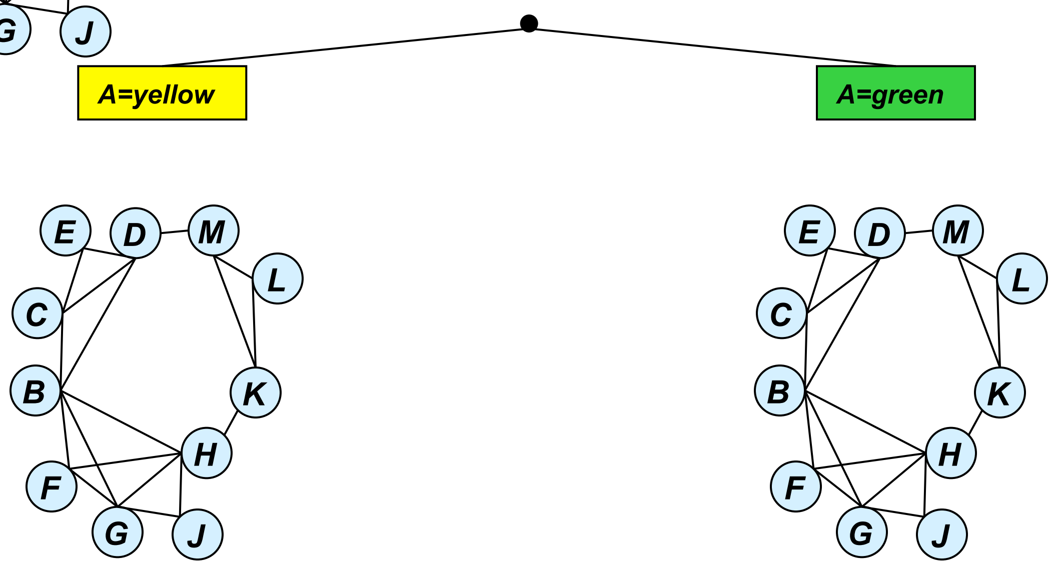
treewidth = 4 - 1 = 3
treewidth = (maximum cluster size) – 1
Separator-width=2

Search over the Cutset

Graph Coloring problem

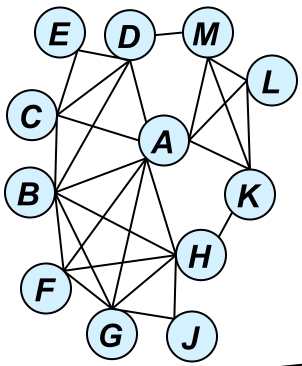


- ***Inference may require too much memory***
- ***Condition (guessing) on some of the variables***

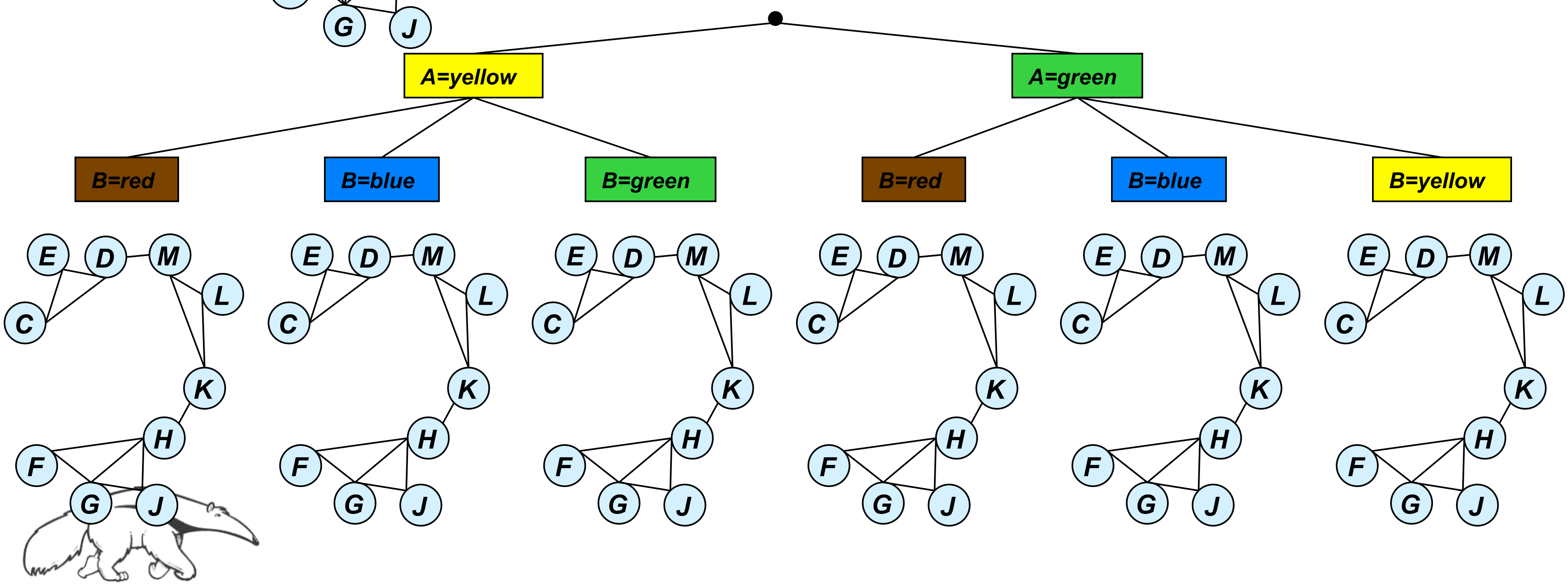


Search over the Cutset (cont)

Graph Coloring problem

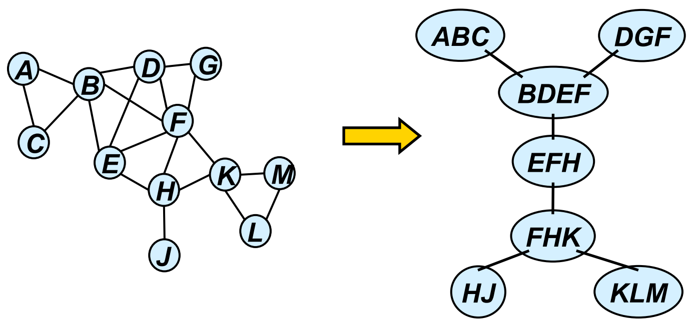


- *Inference may require too much memory*
- *Condition on some of the variables*



Inference vs. Conditioning

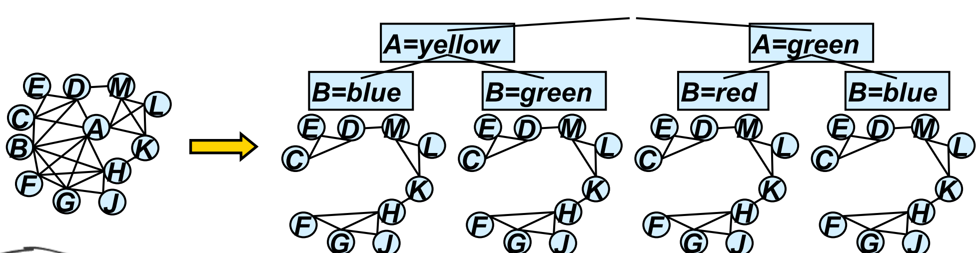
- **By Inference (thinking)**



Exponential in treewidth
Time and space

- ***By Conditioning (guessing)***

•



Exponential in cycle-cutset
Time. linear space

Inference for Optimization: Bucket Elimination

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

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

$$\max_X \prod$$

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

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

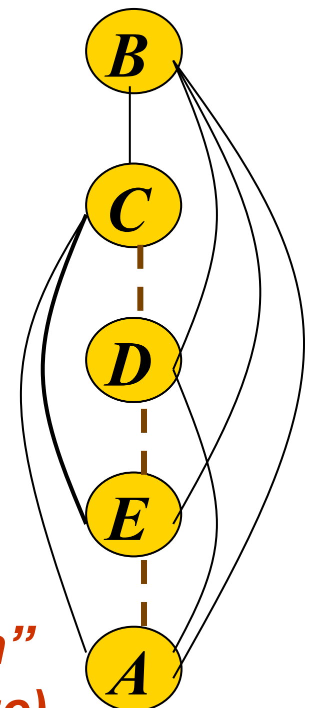
bucket D: $h^C(a,d,e)$

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')

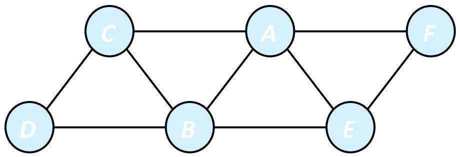
Solution methods

- Solving tree is easy
- Inference: move to trees by clustering
  - (dynamic programming, variable elimination, junction trees)
  - Exploit structure well.
- Search: move to trees by conditioning.
  - Can also exploit structure well.

Outline

- Graphical Models: reasoning principles
- OR Search Trees
- AND/OR Search Spaces for GM
- AND/OR Branch-and-Bound and Best-First Search
- Lower Bounding Heuristics
- Experimental evaluation
- Current work

The Search Space



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

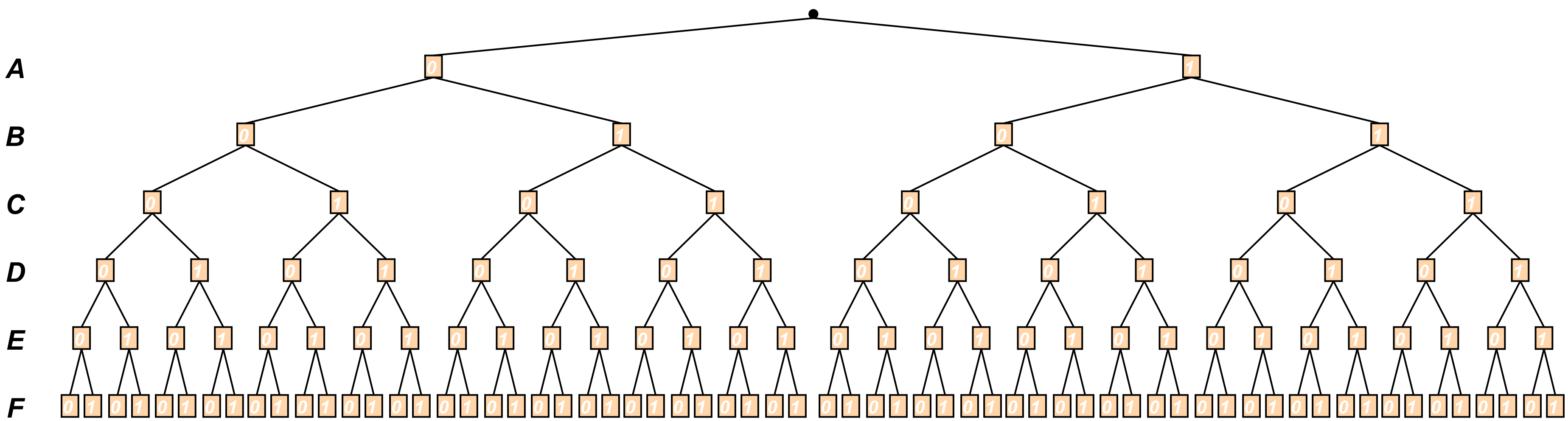
B	D	f_6
0	0	4
0	1	2
1	0	1
1	1	0

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

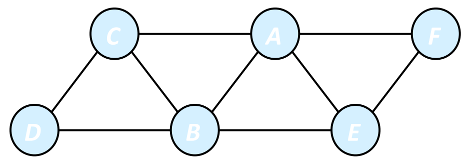
C	D	f_8
0	0	1
0	1	4
1	0	0
1	1	0

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

$$f(X) = \min_X \sum_{i=1}^{9} f_i(X)$$



The Search Space



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

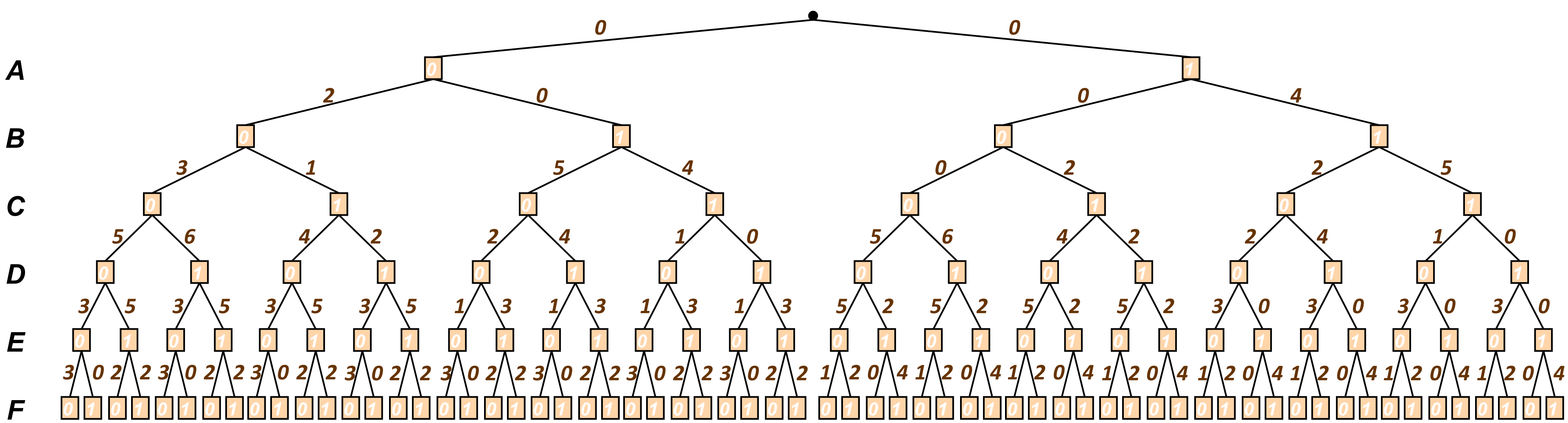
B	D	f_6
0	0	4
0	1	2
1	0	1
1	1	0

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

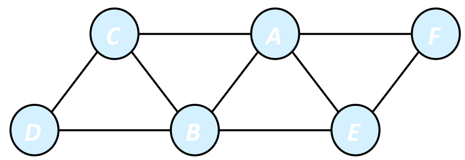
E	F	f_9
0	0	1
0	1	0
1	0	0
1	1	2

$$f(X) = \min_X \sum_{i=1}^{9} f_i(X)$$



are calculated based on cost components

The Search Space



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

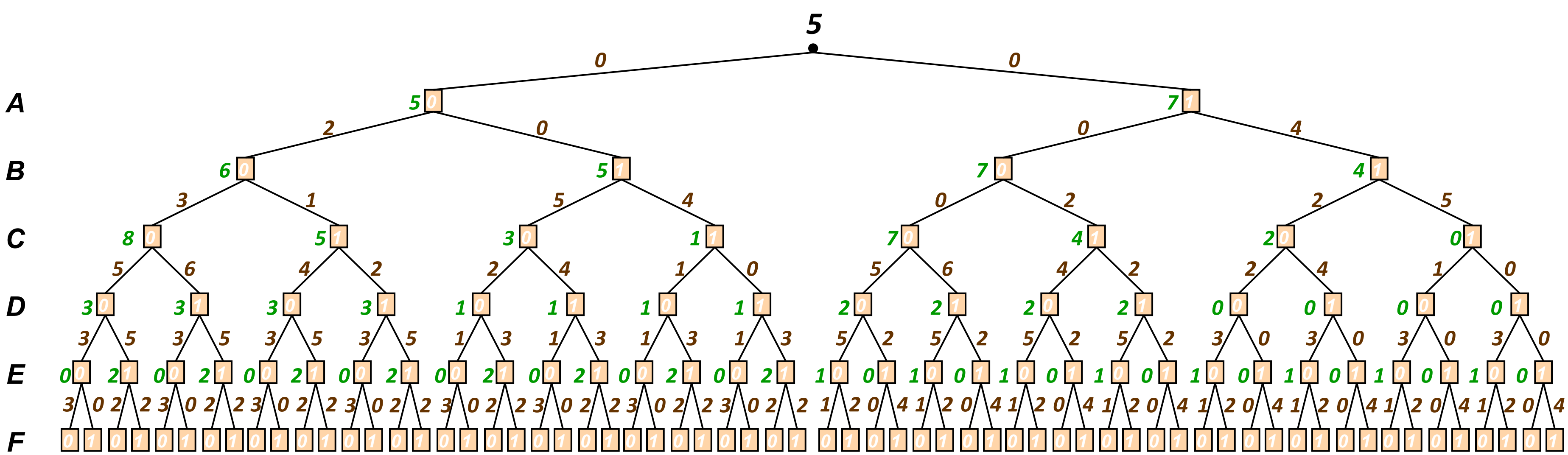
B	D	f_6
0	0	4
0	1	2
1	0	1
1	1	0

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

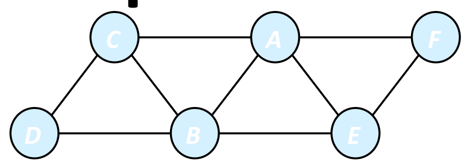
E	F	f_9
0	0	1
0	1	0
1	0	0
1	1	2

$$f(X) = \min_X \sum_{i=1}^{9} f_i(X)$$



Node value = minimal cost solution below it

An Optimal Solution



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

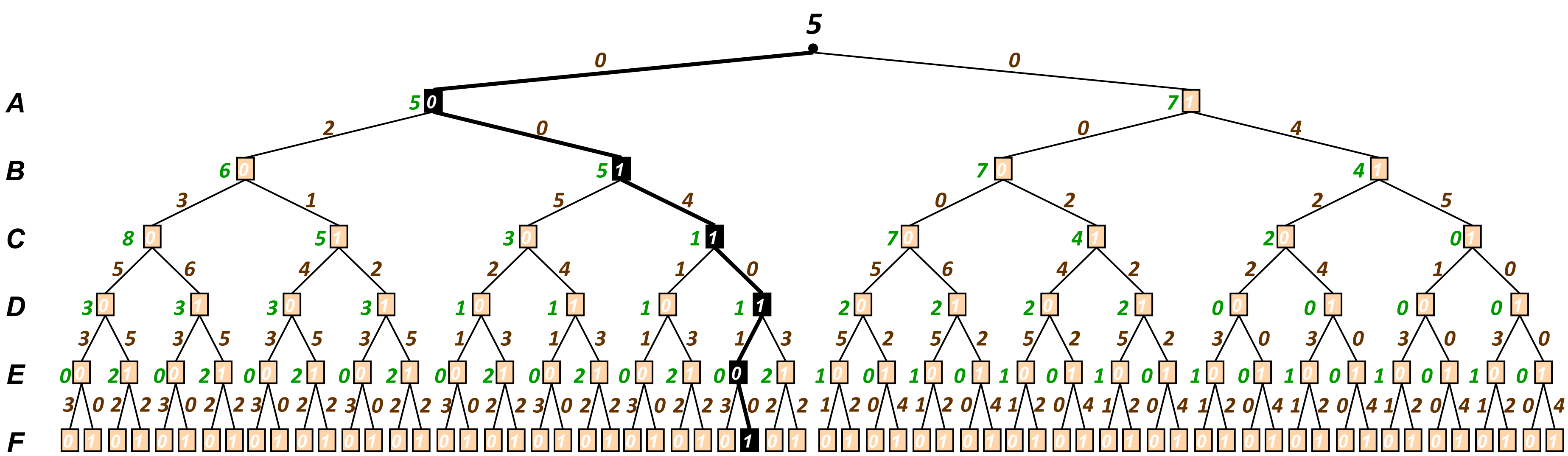
B	D	f_6
0	0	4
0	1	2
1	0	1
1	1	0

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

$$f(\mathrm{X}) = \min_X \sum_{i=1}^{9} f_i(\mathrm{X})$$

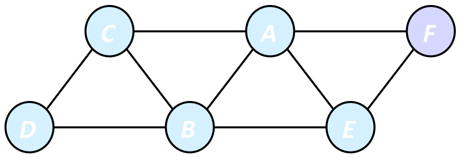


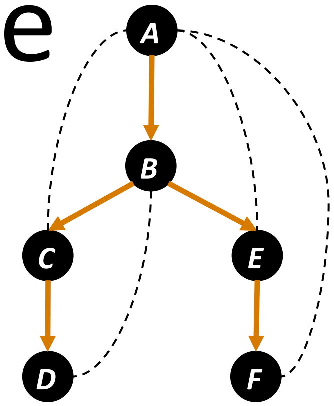
An optimal assignment is A=0, B=1, C=1, D=1, E=0, F=1 with cost 5

Outline

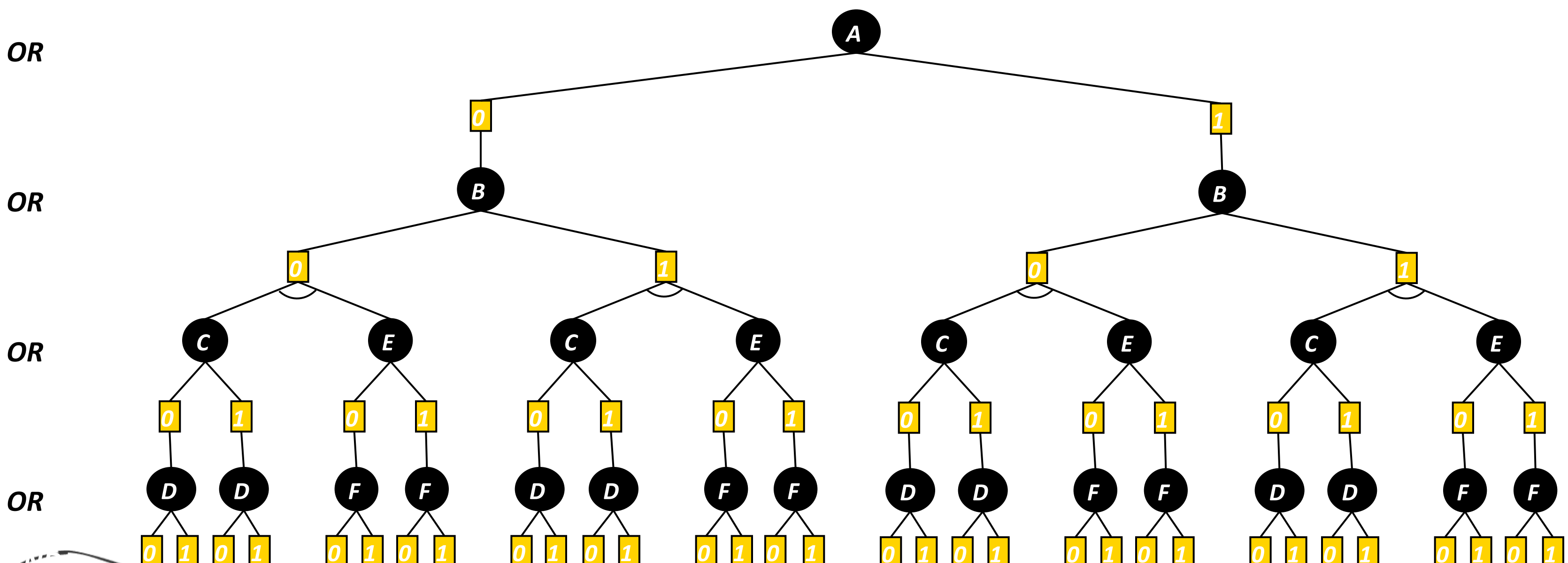
- Graphical Models: reasoning principles
- OR Search Trees
- AND/OR Search Spaces for GM
- AND/OR Branch-and-Bound and Best-First Search
- Lower Bounding Heuristics
- Experimental evaluation
- Current work

The AND/OR Search Tree

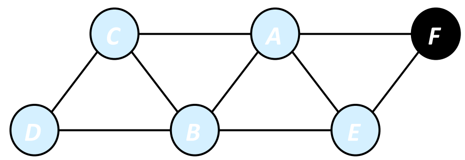


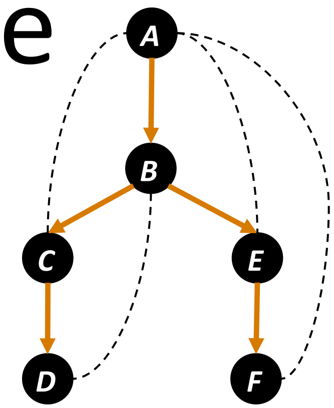


***Pseudo tree** (Freuder & Quinn85)*



The AND/OR Search Tree





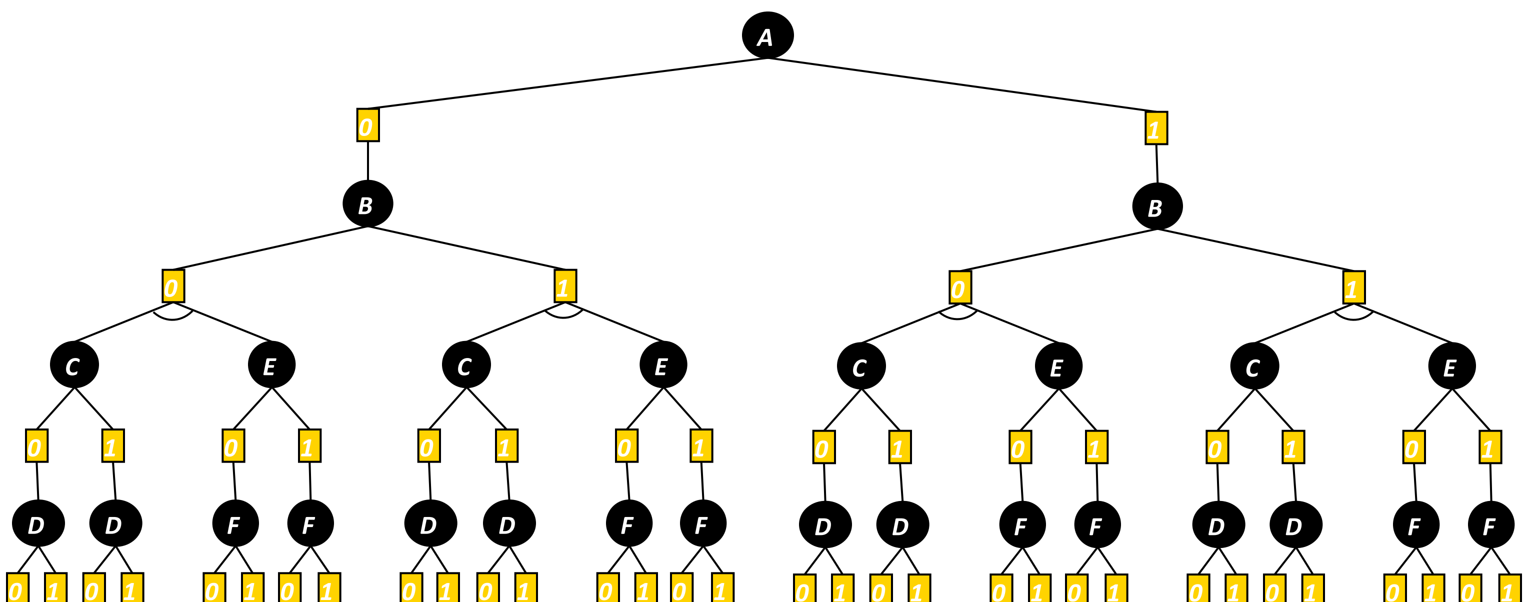
Pseudo tree

OR

OR

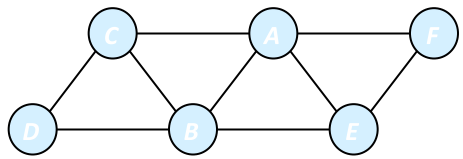
OR

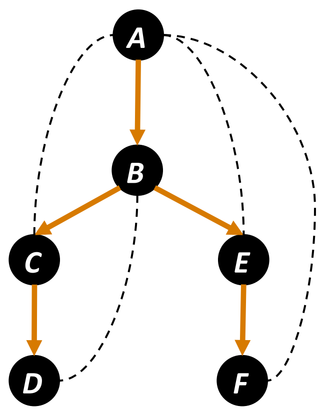
OR



A solution subtree is

Weighted AND/OR Search Tree





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

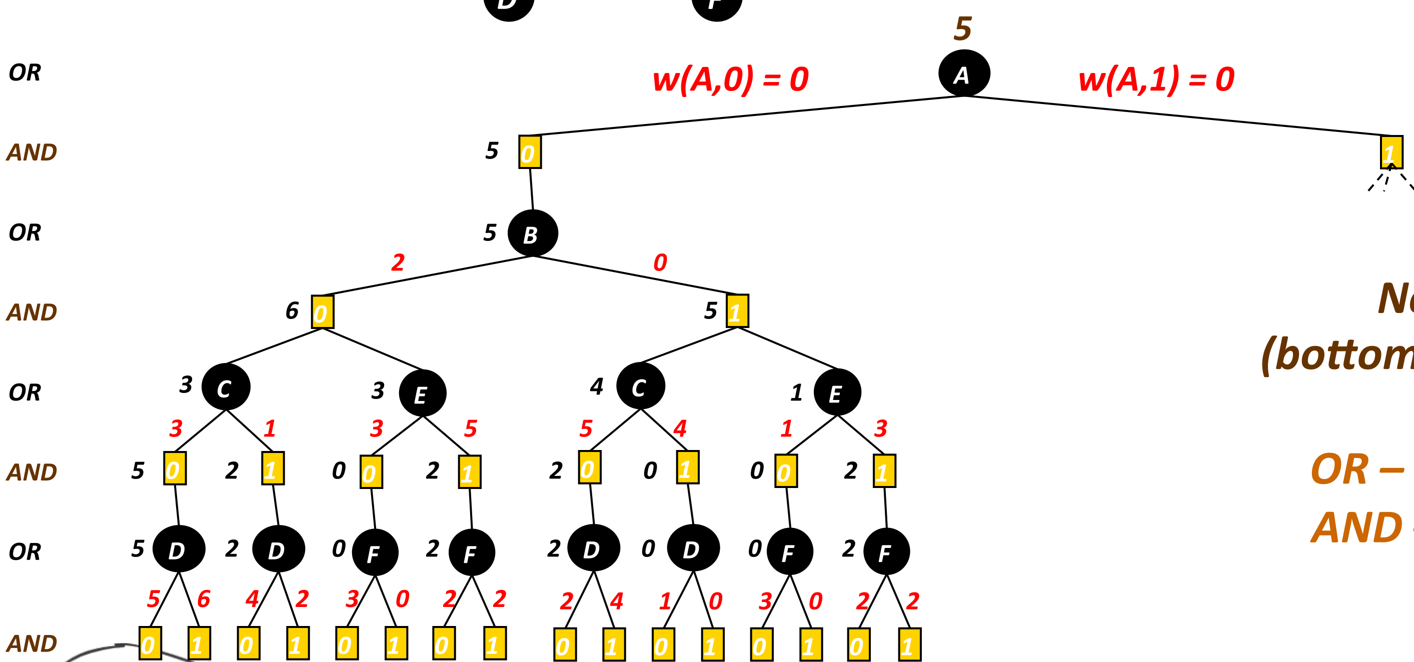
B	D	f_6
0	0	4
0	1	2
1	0	1
1	1	0

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

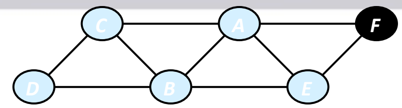
$$f(X) = \min_X \sum_{i=1}^{9} f_i(X)$$

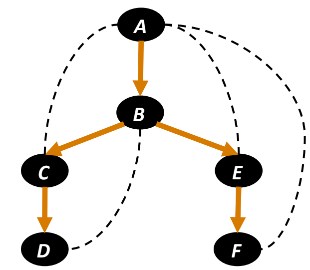


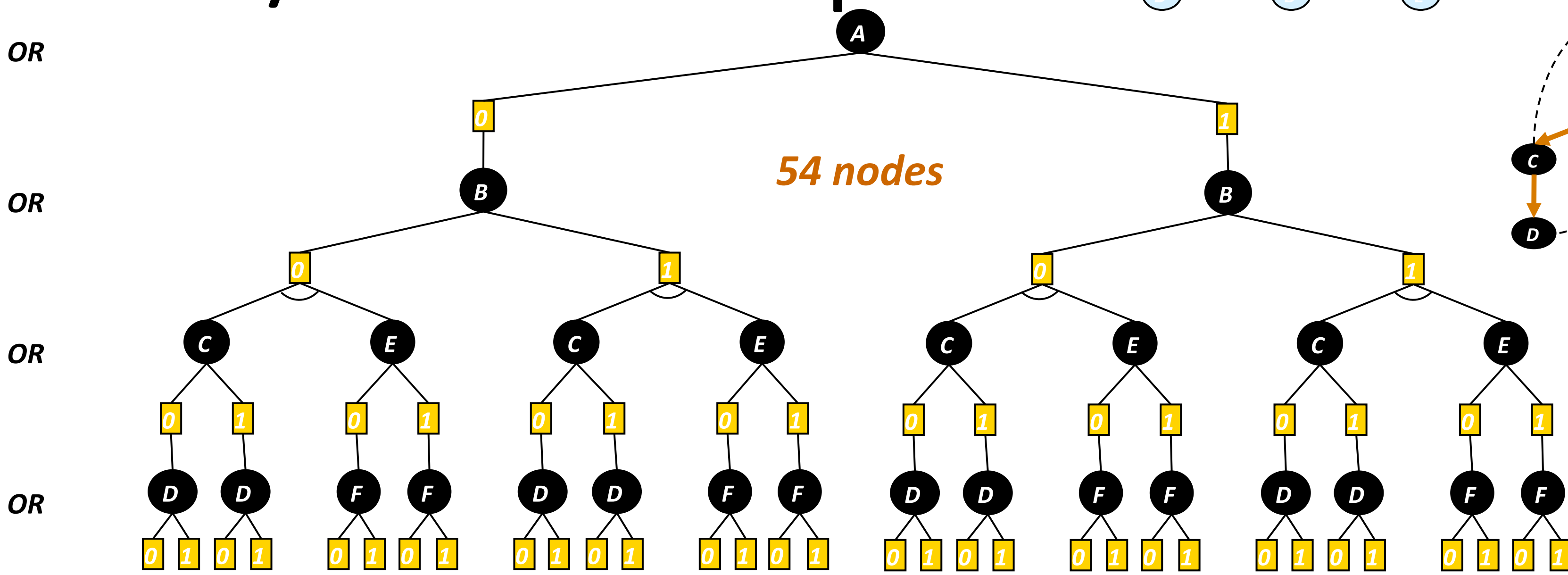
Node Value
(bottom-up evaluation)

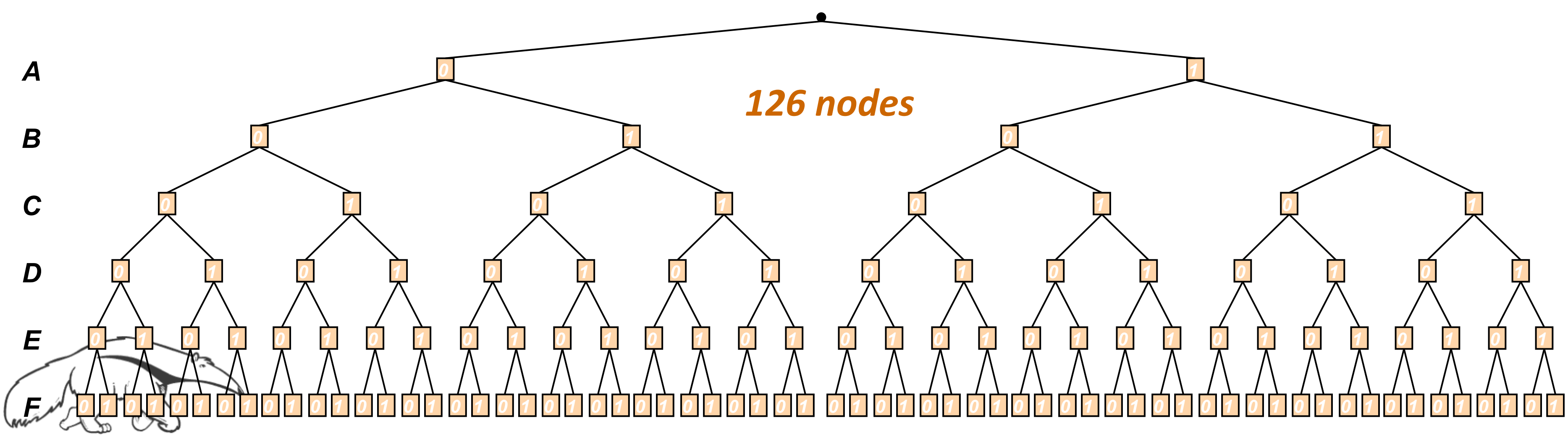
OR – minimization
AND – summation

AND/OR vs. OR Spaces









AND/OR vs. OR Spaces

width	depth	OR space		AND/OR space		
		Time (sec.)	Nodes	Time (sec.)	AND nodes	OR nodes
5	10	3.15	2,097,150	0.03	**10,494**	5,247
4	9	3.13	2,097,150	0.01	**5,102**	2,551
5	10	3.12	2,097,150	0.03	**8,926**	4,463
4	10	3.12	2,097,150	0.02	**7,806**	3,903
5	13	3.11	2,097,150	0.10	**36,510**	18,255

Random graphs with 20 nodes, 20 edges and 2 values per node

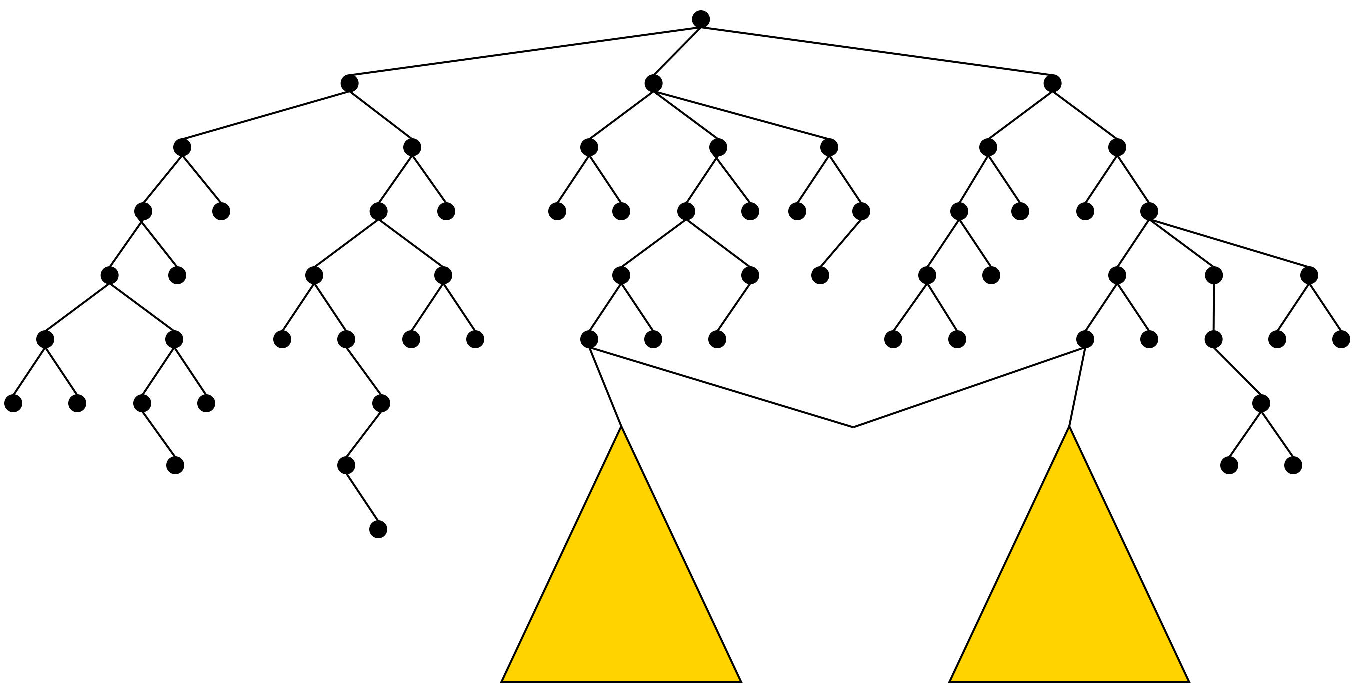
Complexity of AND/OR Tree Search

	AND/OR tree	OR tree
Space	$O(n)$	$O(n)$
Time	$O(n\ k^m)$ $O(n\ k^{w^* \log n})$ (Freuder & Quinn85), (Collin, Dechter & Katz91), (Bayardo & Miranker95), (Darwiche01)	$O(k^n)$

k = *domain size*
m = *depth of pseudo-tree*
n = *number of variables*
*w** = *treewidth*

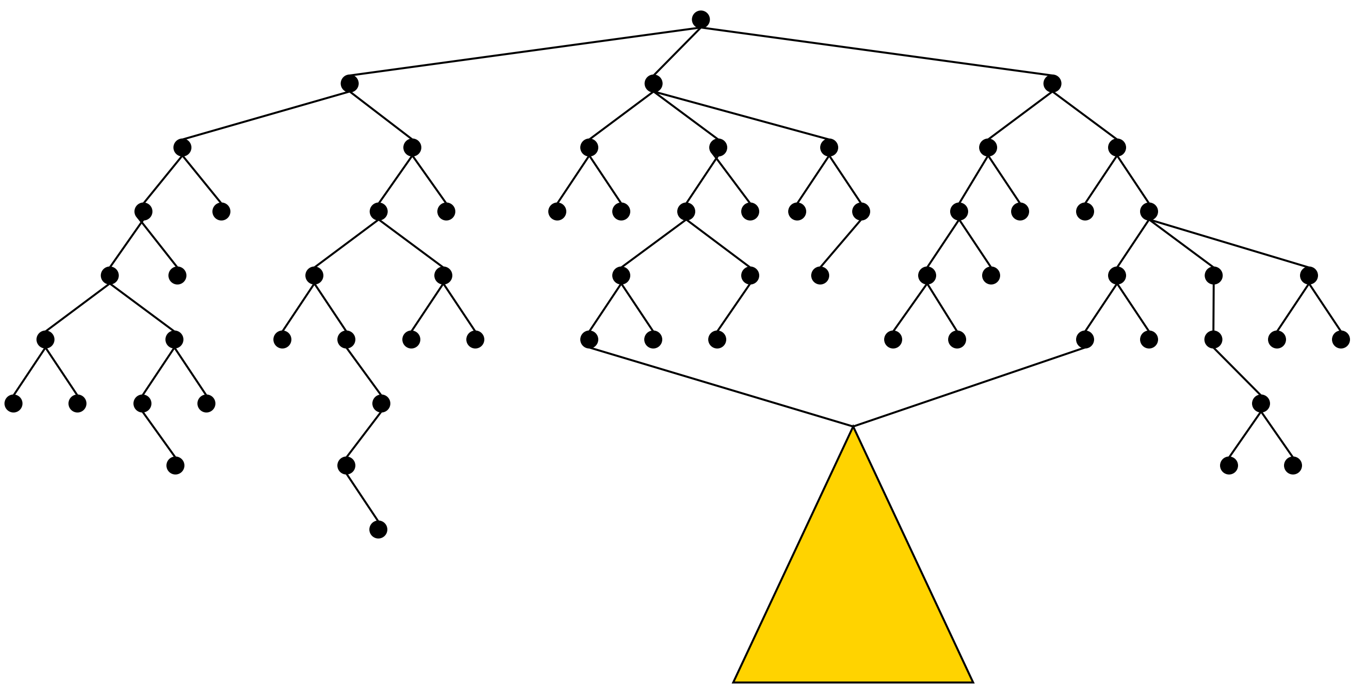
From Search Trees to Search Graphs

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

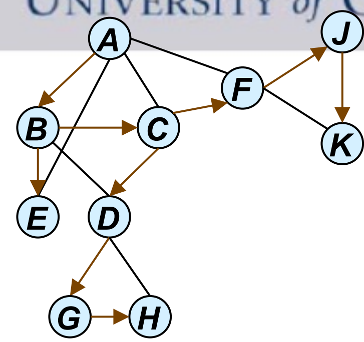


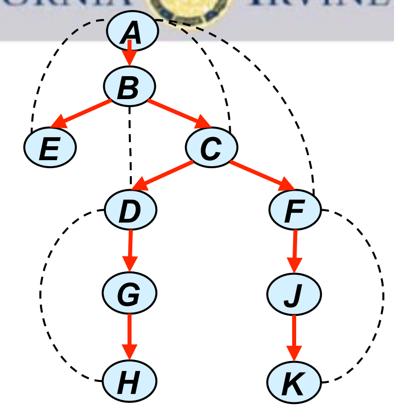
From Search Trees to Search Graphs

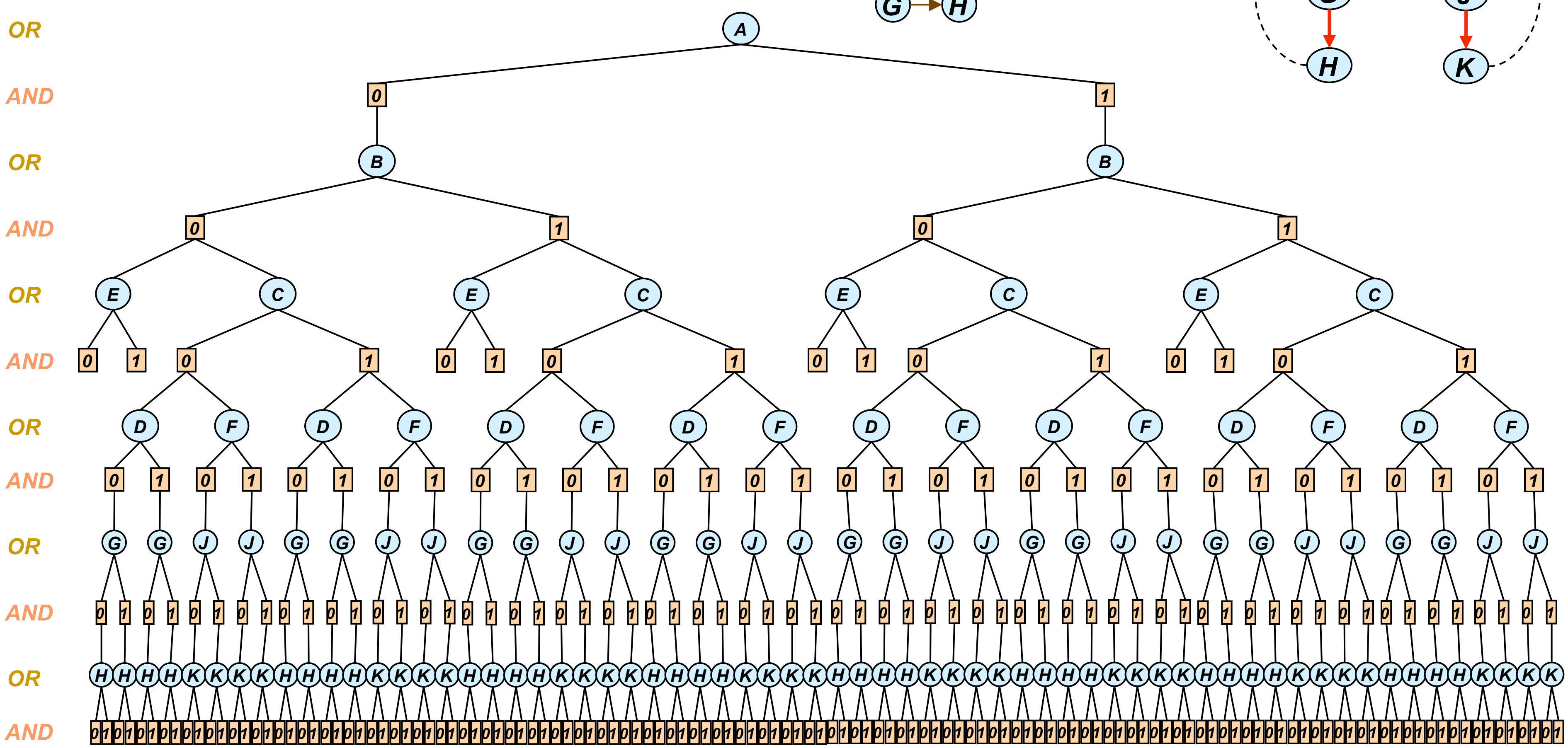
- Any two nodes that root identical sub-trees or sub-graphs can be merged



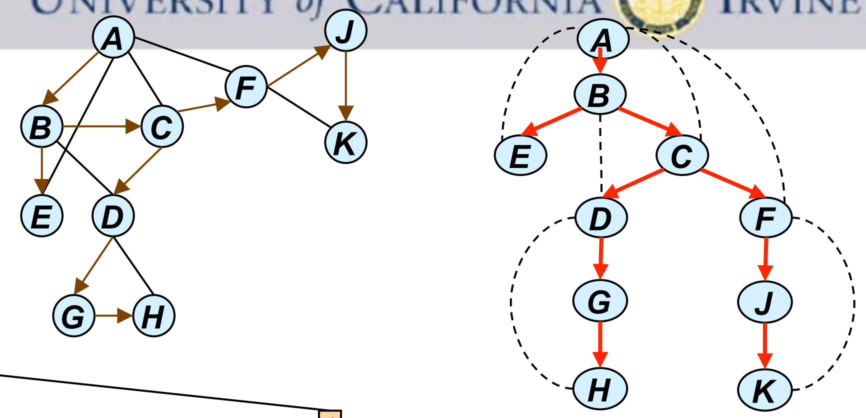
From AND/OR Tree

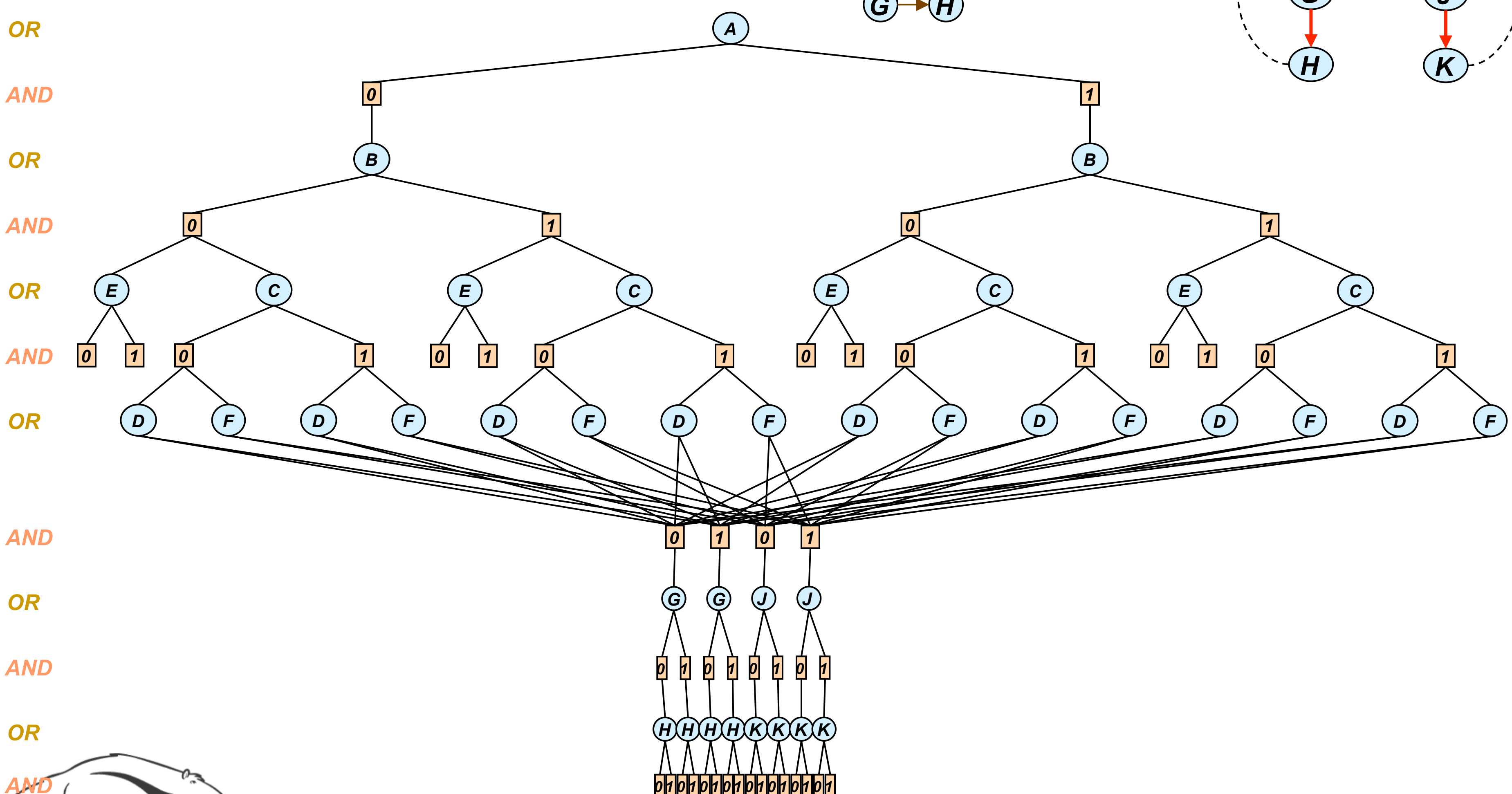






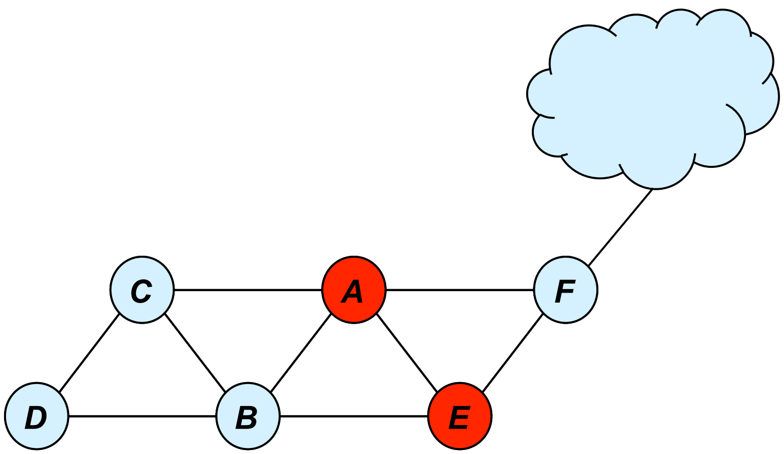
An AND/OR Graph

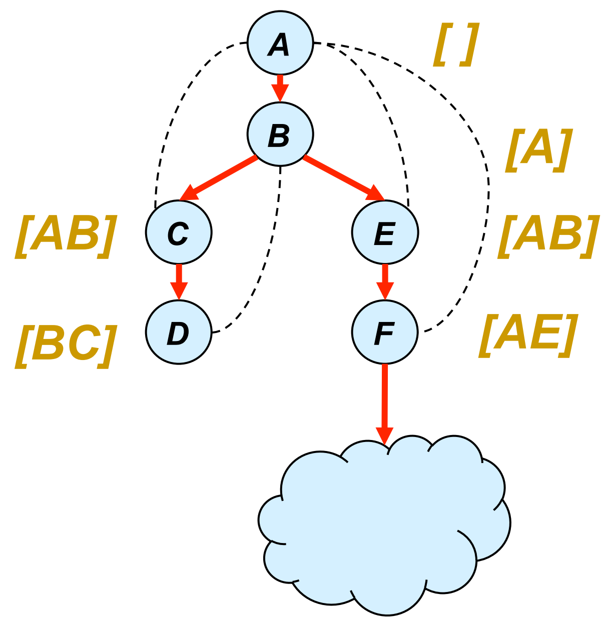




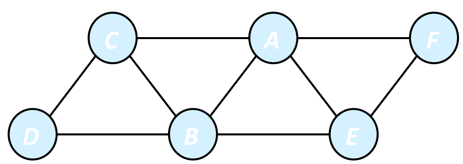
Merging based on context

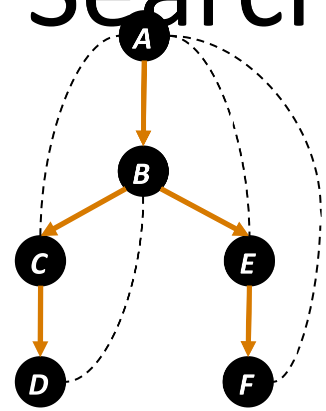
context (X) = ancestors of X connected to → *X* / → *descendants of X*





AND/OR Search Graph





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

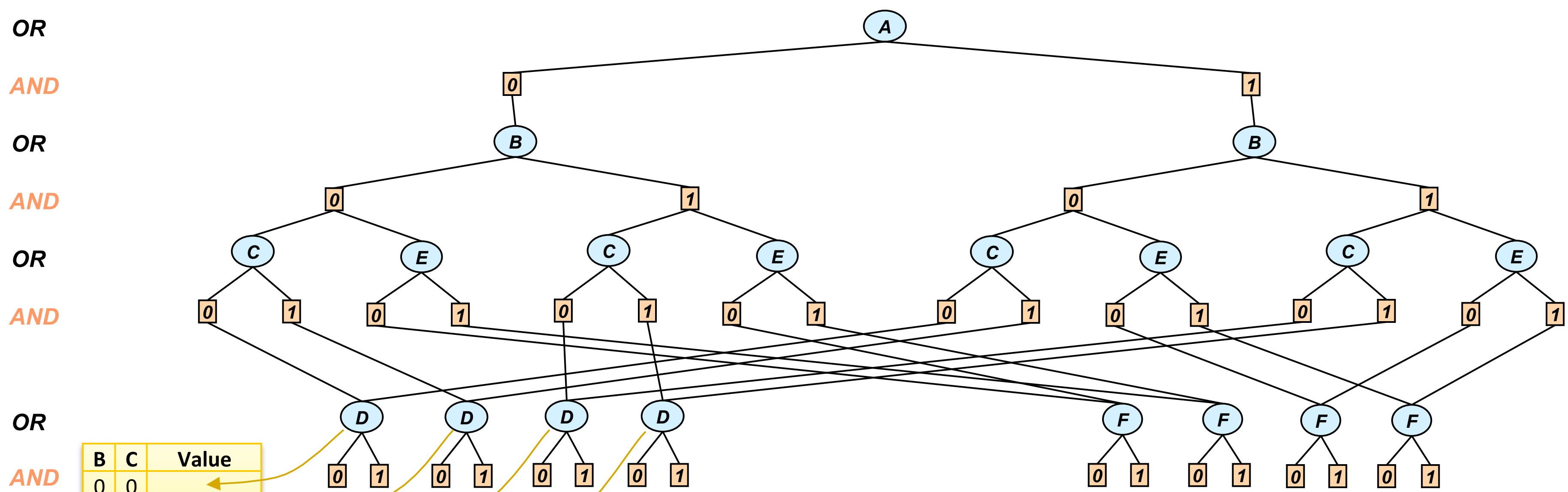
B	D	f_6
0	0	4
0	1	2
1	0	1
1	1	0

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

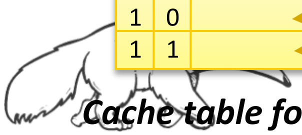
C	D	f_8
0	0	1
0	1	4
1	0	0
1	1	0

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

$$f(X) = \min_X \sum_{i=1}^{9} f_i(X)$$



B	C	Value
0	0	
0	1	
1	0	
1	1	

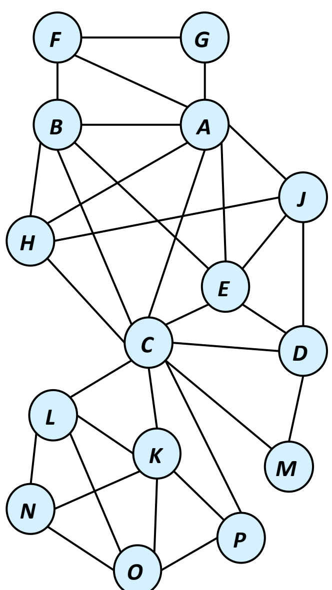


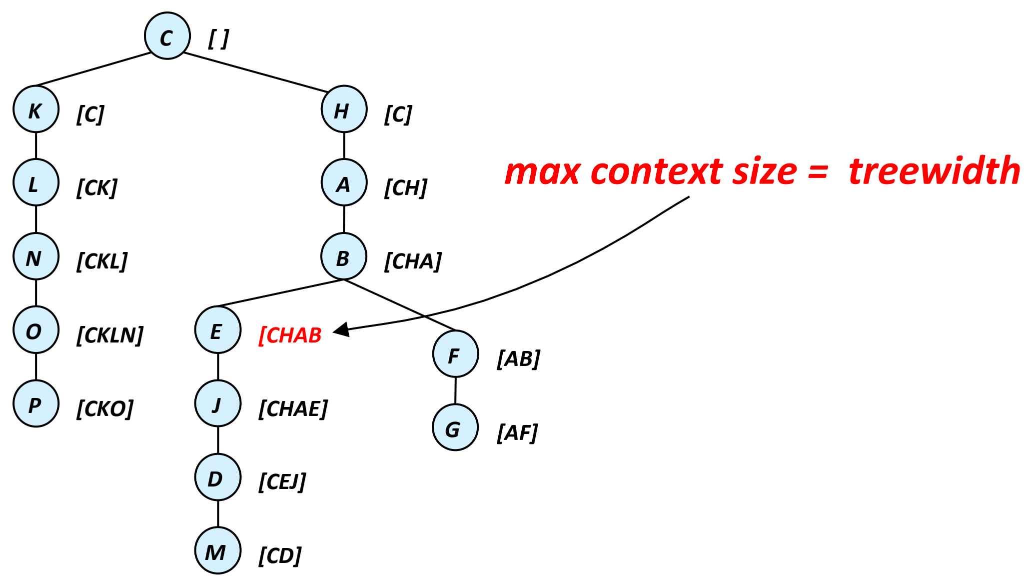
Cache table for D

context minimal graph

How Big Is The Context?

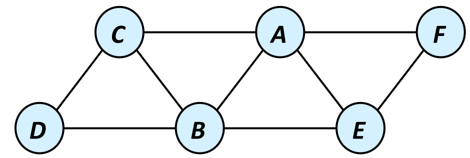
Theorem: The maximum context size for a pseudo tree is equal to the treewidth of the graph along the pseudo tree.

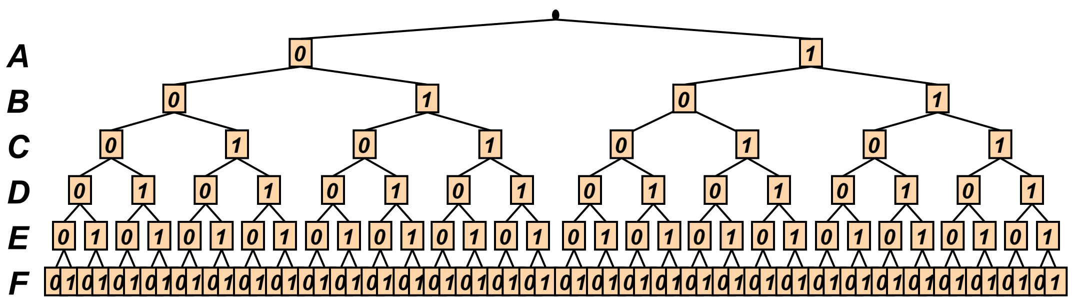




(C K H A B E J L N O D P M F G)

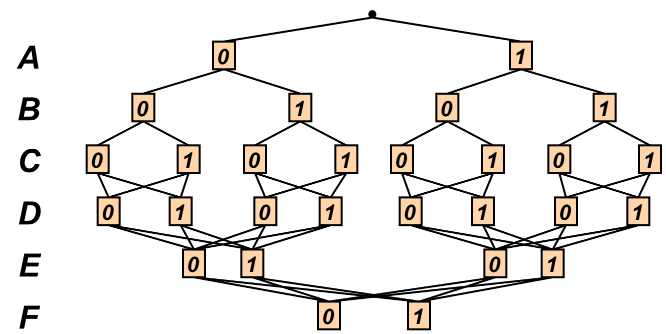
All Four Search Spaces





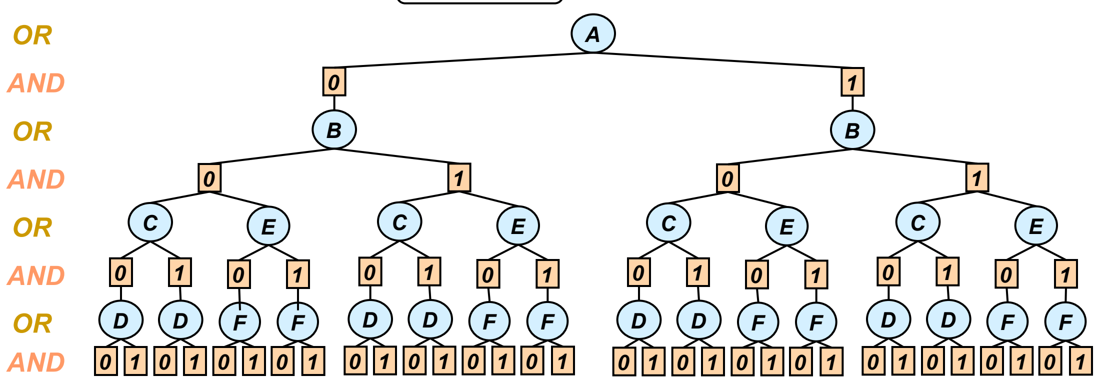
Full OR search tree

126 nodes



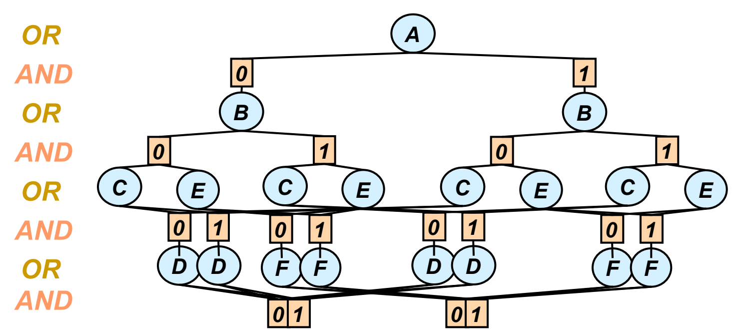
Context minimal OR search graph

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 space

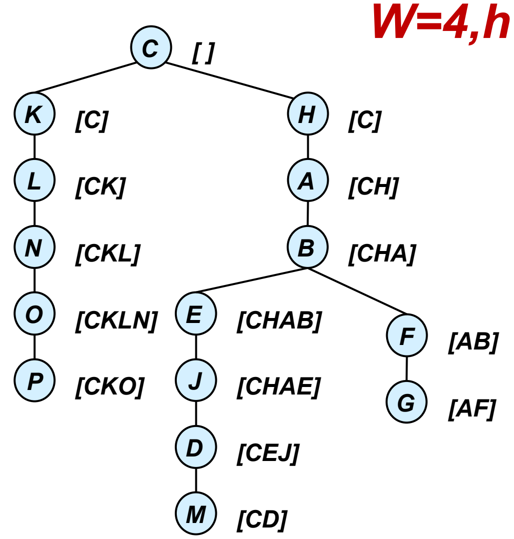
Complexity of AND/OR Graph Search

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

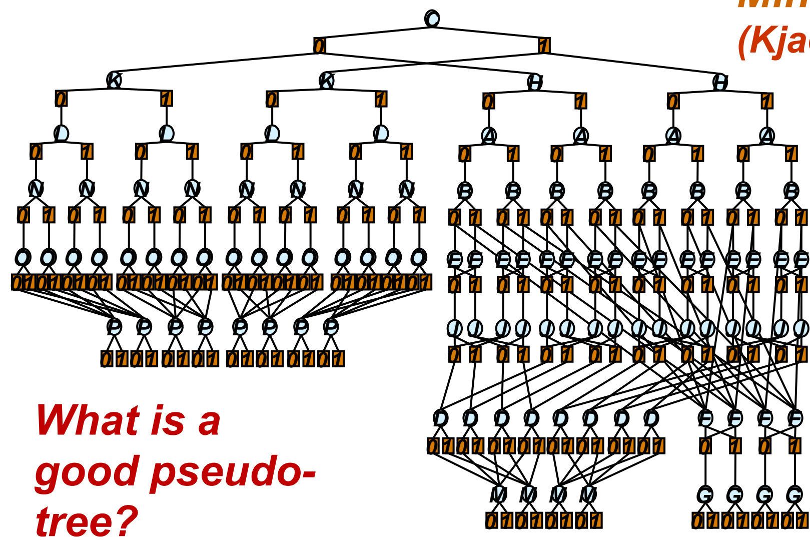
***k* = domain size**
***n* = number of variables**
w*= treewidth
pw*= pathwidth

The impact of the pseudo-tree

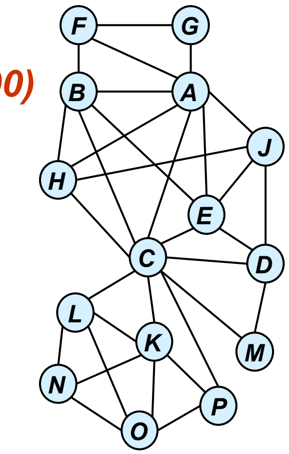
W=4,h=8



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



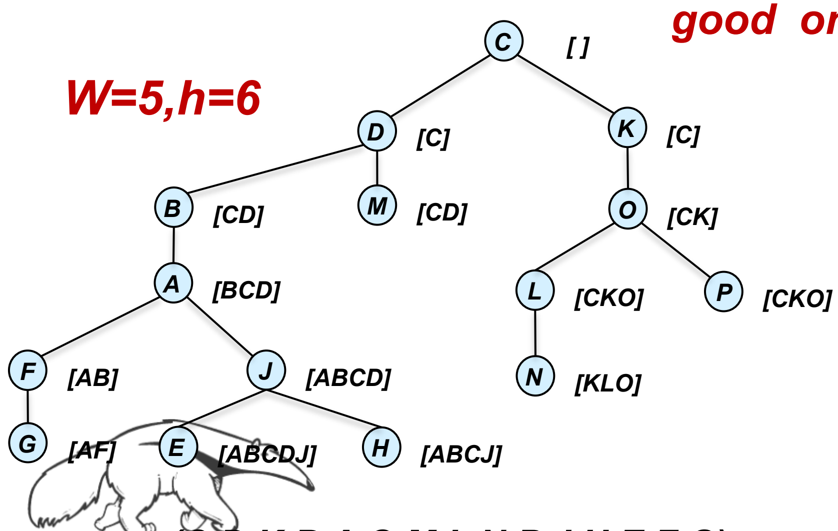
Min-Fill (Kjaerulff90)



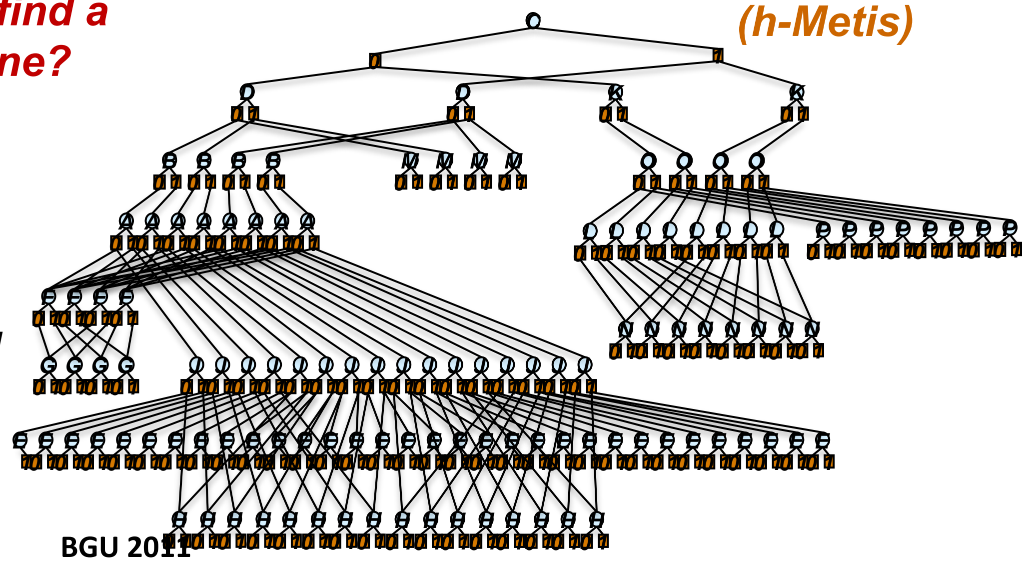
What is a good pseudo-tree?
How to find a good one?

Hypergraph Partitioning (h-Metis)

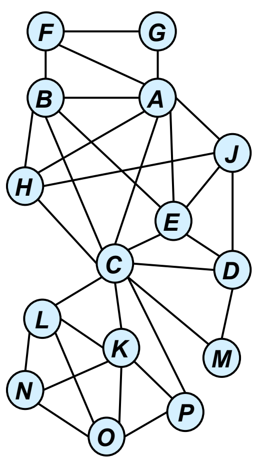
W=5,h=6

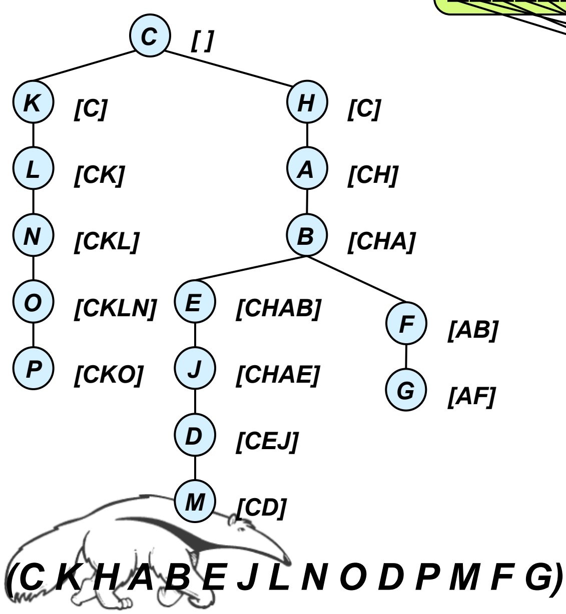


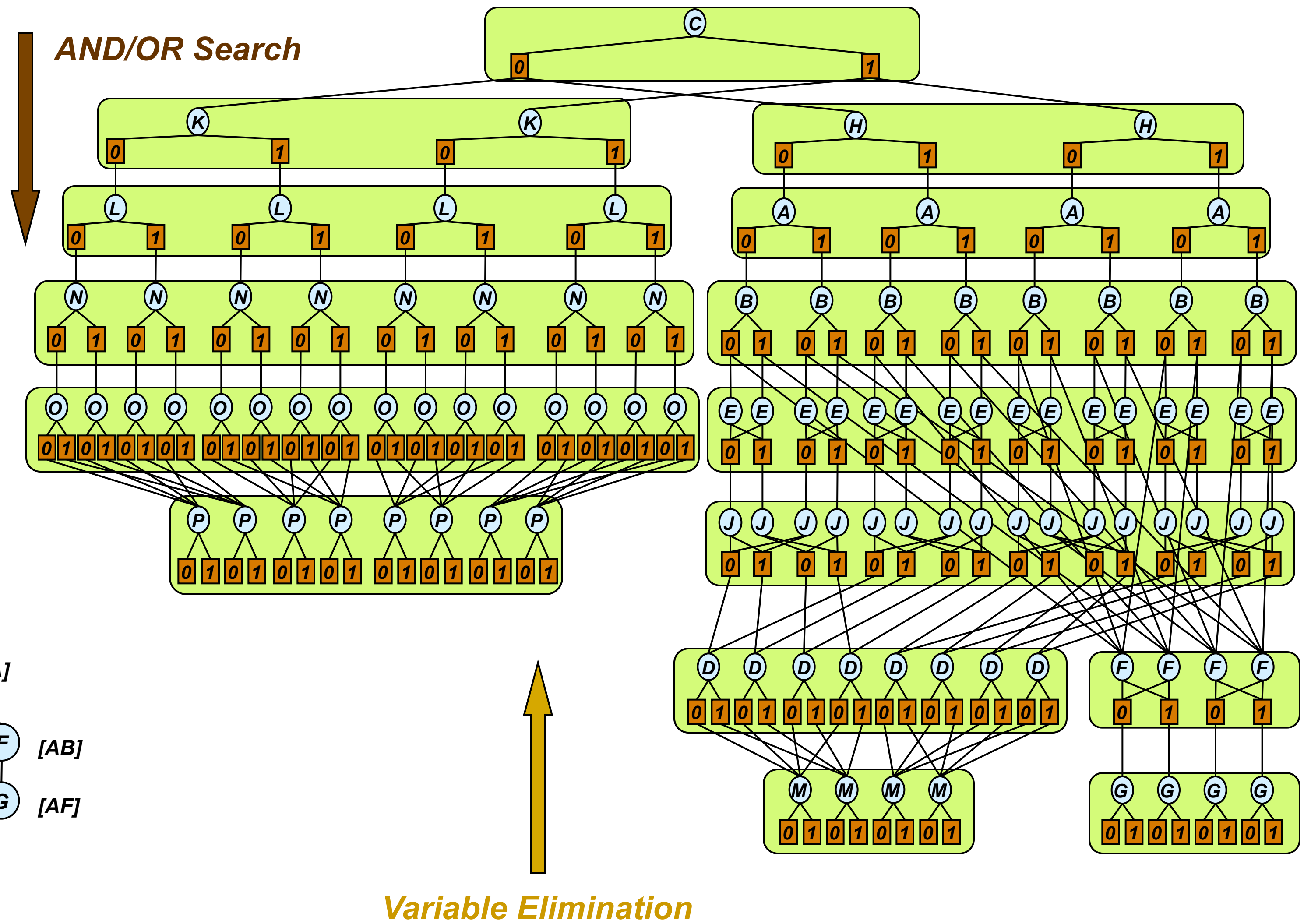
(C D K B A O M L N P J H E F G)



AND/OR Context Minimal Graph







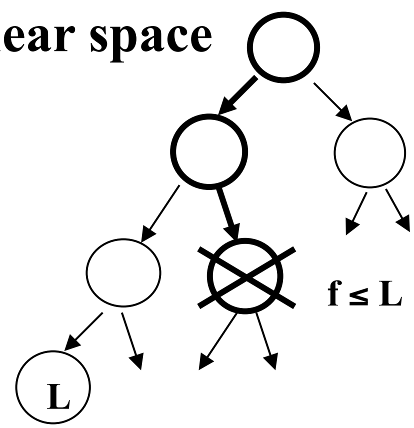
Outline

- Graphical Models: reasoning principles
- OR Search Trees
- AND/OR Search Spaces for GM
- AND/OR Branch-and-Bound and Best-First Search
- Lower Bounding Heuristics
- Experimental evaluation
- Current work

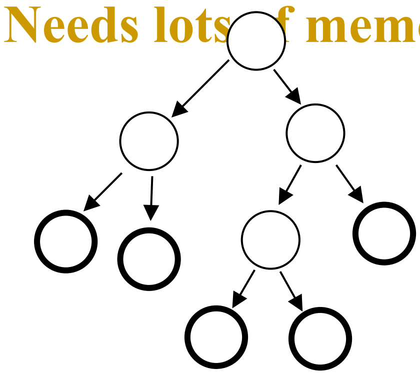
Basic Heuristic Search Schemes

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

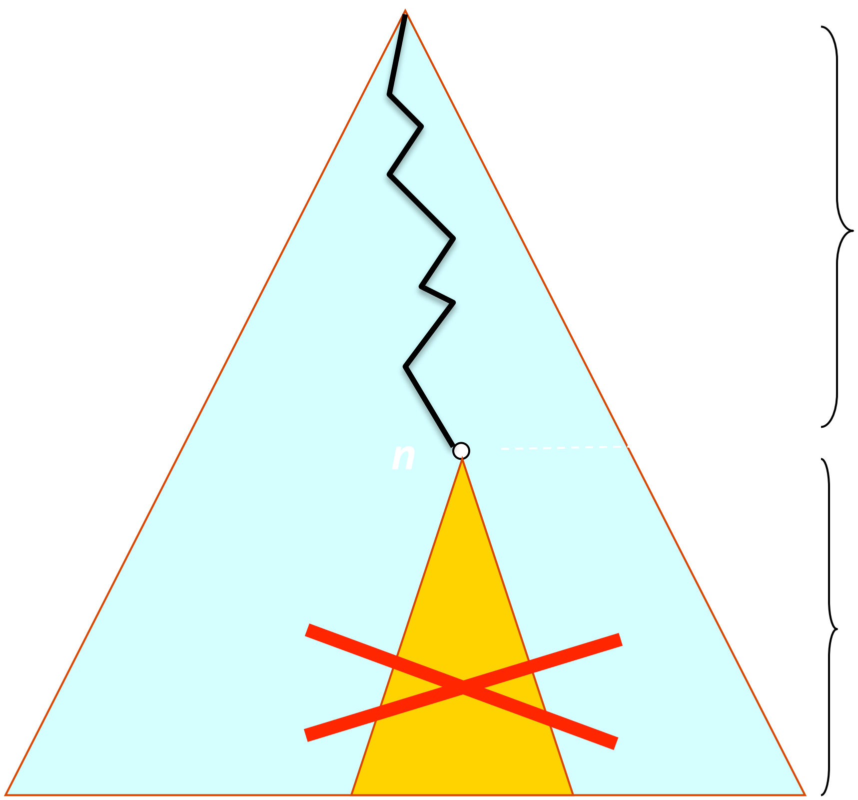
1.Branch and Bound
Use heuristic function $f(x^p)$ to prune the depth-first search tree.
Linear space



2.Best-First Search
Always expand the node with the highest heuristic value $f(x^p)$.
Needs lots of memory



Branch-and-Bound Search



g(n)=cost of the search path to n

H(n) = estimates the optimal cost below n

OR Search Tree

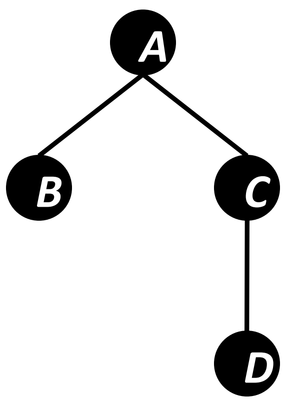
Upper Bound UB

Lower Bound LB(n)

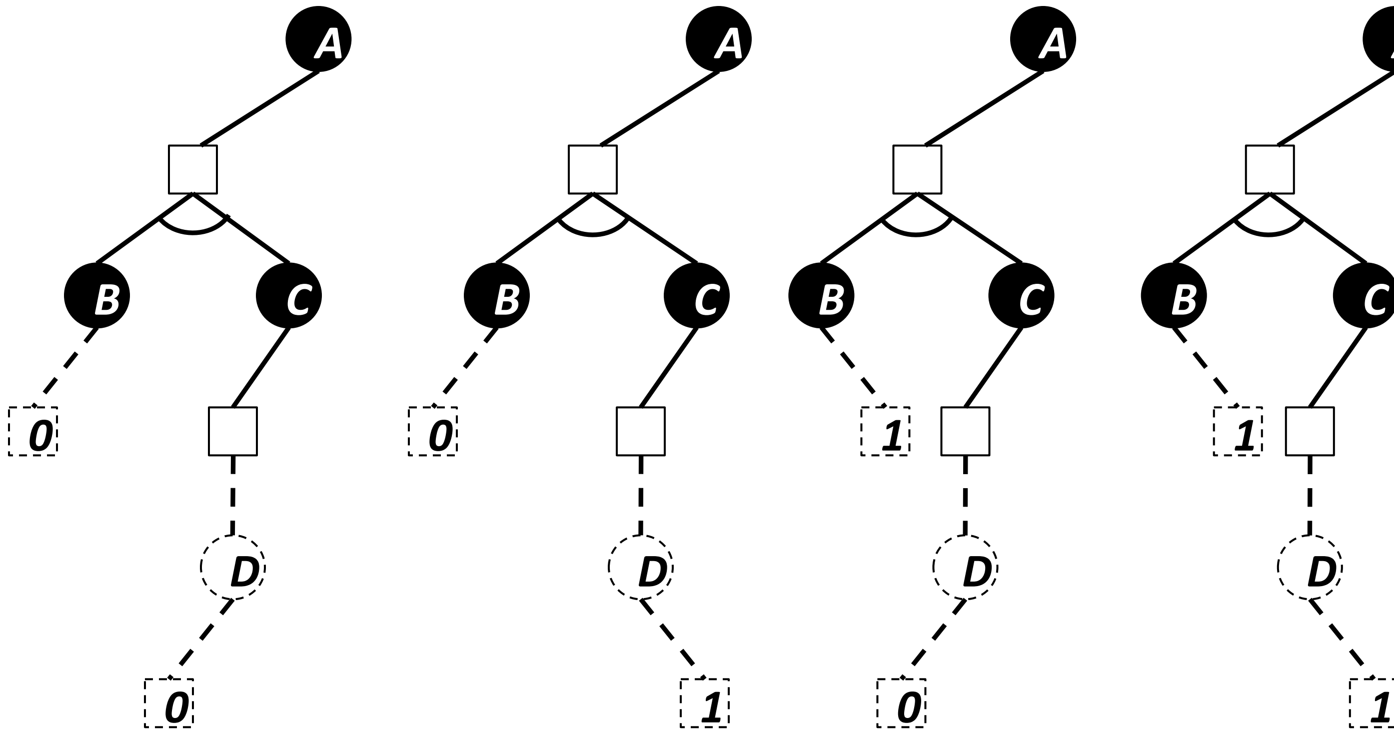
Prune if LB(n) ≥ UB

(Lawler & Wood66)

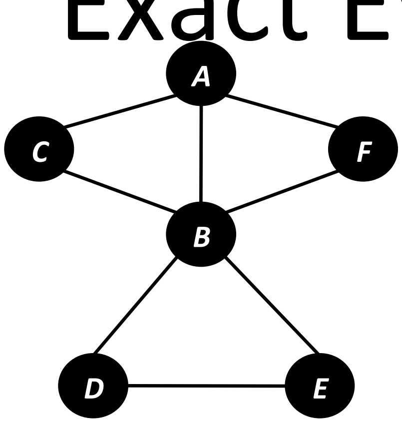
Partial Solution Tree



Pseudo tree



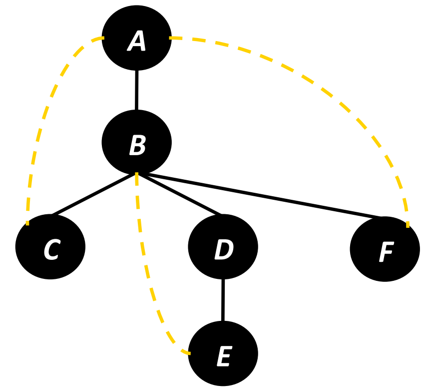
Exact Evaluation Function

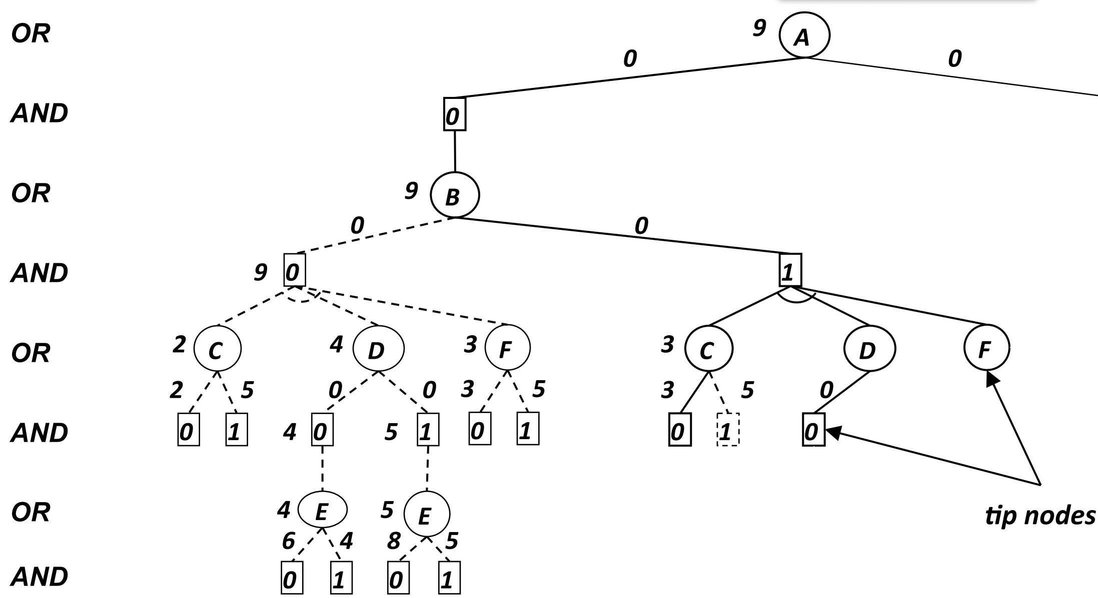


A	B	C	$f_1(ABC)$
0	0	0	2
0	0	1	5
0	1	0	3
0	1	1	5
1	0	0	9
1	0	1	3
1	1	0	7
1	1	1	2

A	B	F	$f_2(ABF)$
0	0	0	3
0	0	1	5
0	1	0	1
0	1	1	4
1	0	0	6
1	0	1	5
1	1	0	6
1	1	1	5

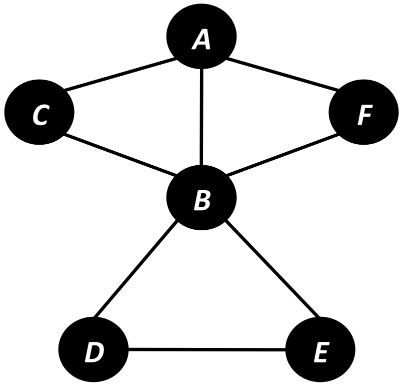
B	D	E	$f_3(BDE)$
0	0	0	6
0	0	1	4
0	1	0	8
0	1	1	5
1	0	0	9
1	0	1	3
1	1	0	7
1	1	1	4





$f^*(T') = w(A,0) + w(B,1) + w(C,0) + w(D,0) + v(D,0) + v(F)$

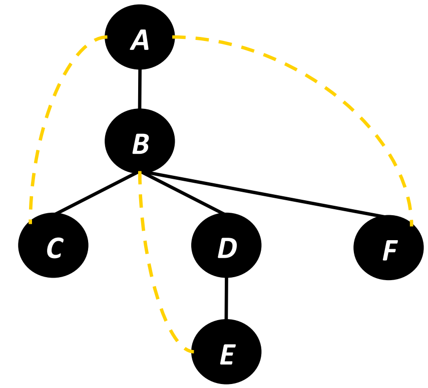
Heuristic Evaluation Function

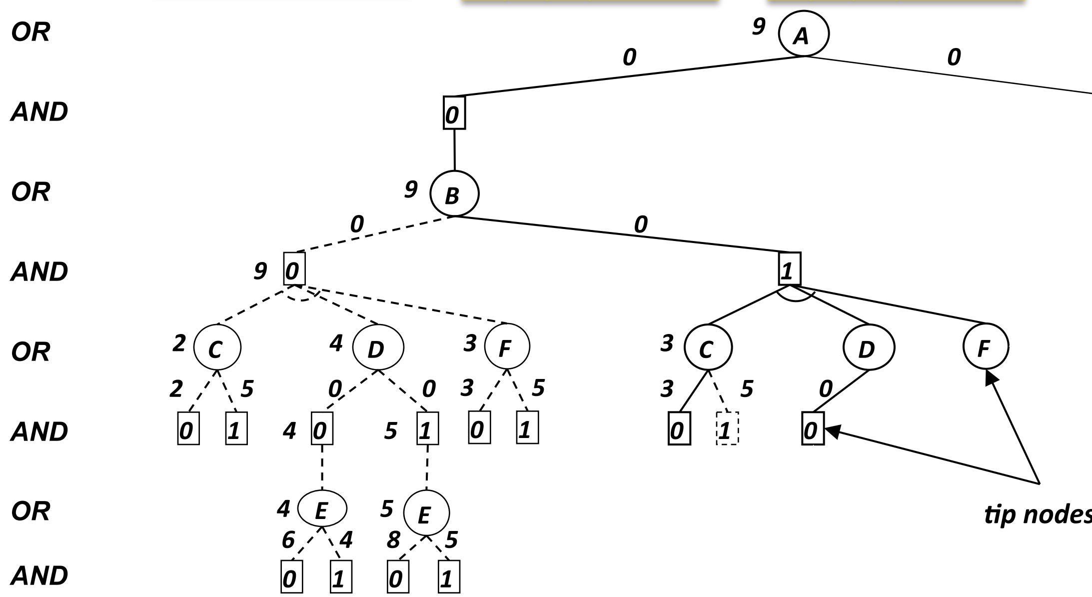


A	B	C	$f_1(ABC)$
0	0	0	2
0	0	1	5
0	1	0	3
0	1	1	5
1	0	0	9
1	0	1	3
1	1	0	7
1	1	1	2

A	B	F	$f_2(ABF)$
0	0	0	3
0	0	1	5
0	1	0	1
0	1	1	4
1	0	0	6
1	0	1	5
1	1	0	6
1	1	1	5

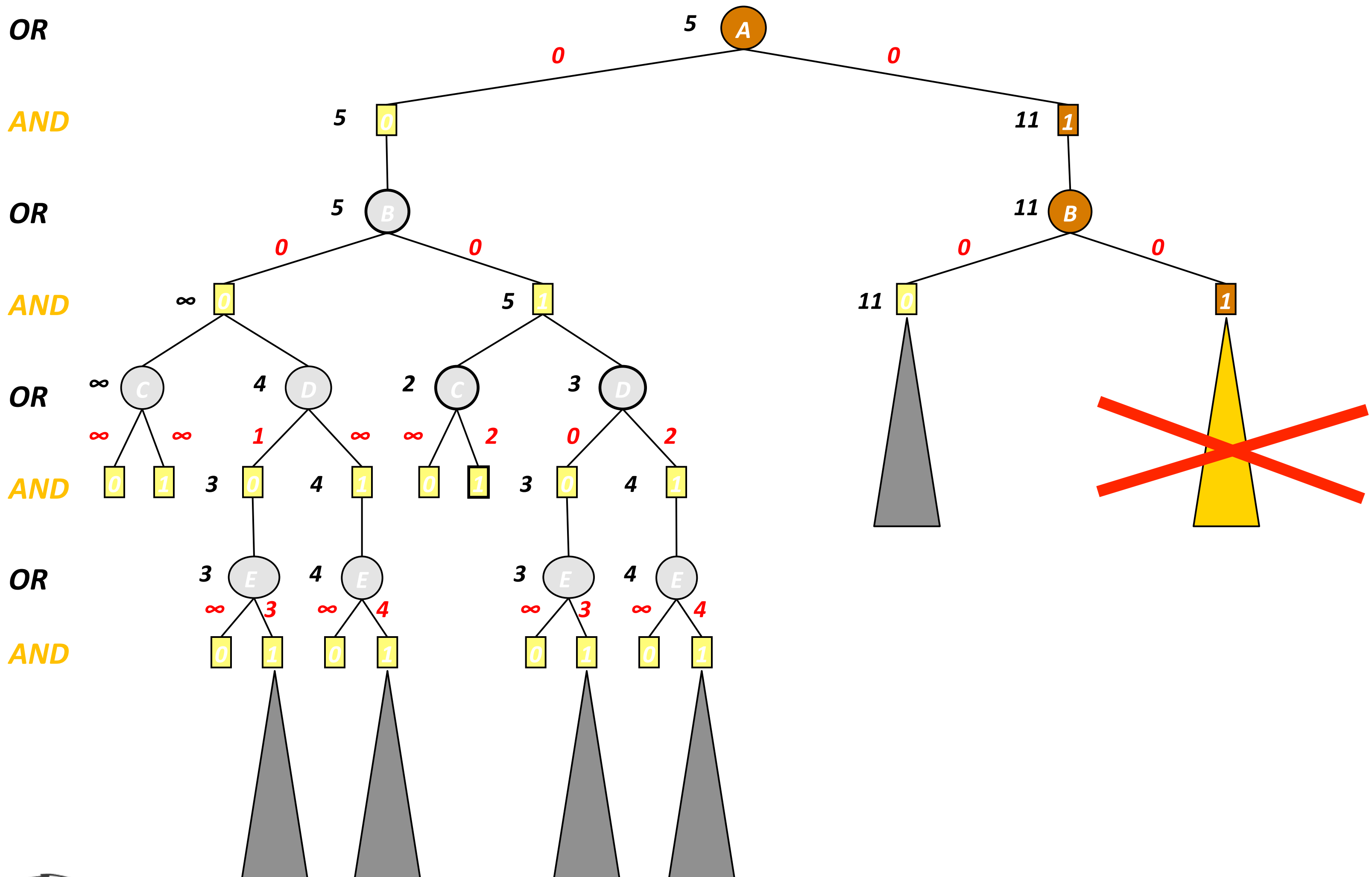
B	D	E	$f_3(BDE)$
0	0	0	6
0	0	1	4
0	1	0	8
0	1	1	5
1	0	0	9
1	0	1	3
1	1	0	7
1	1	1	4





$$f(T') = w(A,0) + w(B,1) + w(C,0) + w(D,0) + h(D,0) + h(F) = 12 \le f^*(T')$$

AND/OR Branch-and-Bound Search



Best-First Search Principle

- Best-first search expands first the node with the best heuristic evaluation function among all nodes encountered so far
- It never expands nodes whose evaluation function is beyond the optimal one, unlike depth-first search algorithms (Dechter & Pearl85)
- Superior among memory intensive algorithms employing the

Best-First AND/OR Search

- Maintains the set of best partial solution trees
- **EXPAND** (top-down)
  - Traces down marked connectors from root (best partial solution tree)
  - Expands a tip node by generating its successors **n’**
  - Associate each successor with heuristic estimate **h(n’)**
    - Initialize **v(n’) = h(n’)**
- **REVISE** (bottom-up)
  - Updates node values **v(n)**
    - OR nodes: **minimization**
    - AND nodes: **summation**
  - Marks the most promising solution tree from the root
  - Label the nodes as SOLVED:
    - OR is SOLVED if marked child is SOLVED
    - AND is SOLVED if all children are SOLVED
- Terminate when root node is SOLVED

[specializes Nilsson’s AO* to graphical models (Nilsson80)]

Outline

- Graphical Models: reasoning principles
- OR Search Trees
- AND/OR Search Spaces for GM
- AND/OR Branch-and-Bound and Best-First Search
- Lower Bounding Heuristics
- Experimental evaluation
- Current work

How to Generate Heuristics *(Pearl86)*

- The principle of relaxed models
  - Linear relaxation for integer programs
  - Mini-Bucket Elimination for graphical models
  - Bounded directional consistency ideas
  - Pattern databases

Inference for Optimization: Bucket Elimination

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

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

$$\max_X \prod$$

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

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

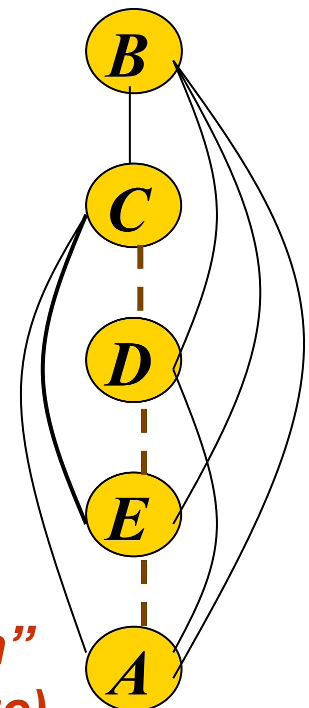
bucket D: $h^C(a,d,e)$

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

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

$P(a|e=0)$

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



Mini-bucket approximation: MPE task

Split a bucket into mini-buckets =>bound complexity

bucket (X) =

$$\{ h_1, \ldots, h_r, h_{r+1}, \ldots, h_n \}$$

$$h^X = \max_X \prod_{i=1}^{n} h_i$$

$$\{ h_1, \ldots, h_r \} \qquad \{ h_{r+1}, \ldots, h_n \}$$

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

$$h^X \le g^X$$

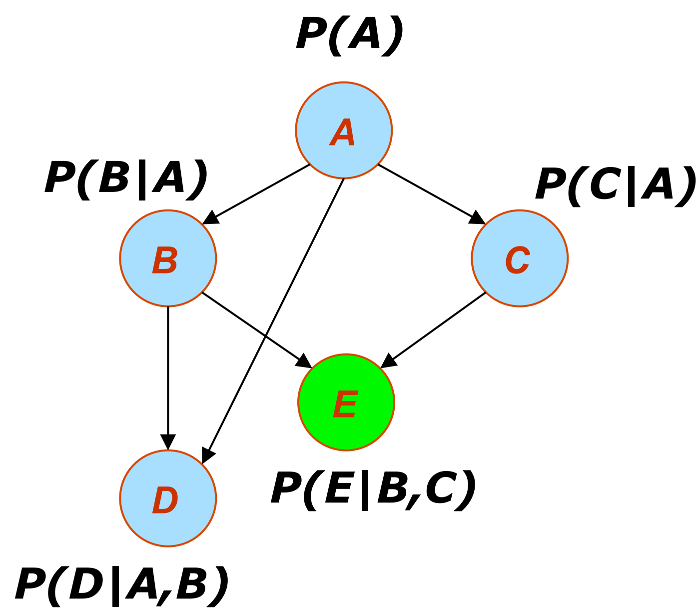
Exponential complexity decrease : $O(e^n) \rightarrow O(e^r) + O(e^{n-r})$

Mini-Bucket Elimination

Node duplication, renaming

	$max_B\Pi$	$max_B\Pi$
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)**	

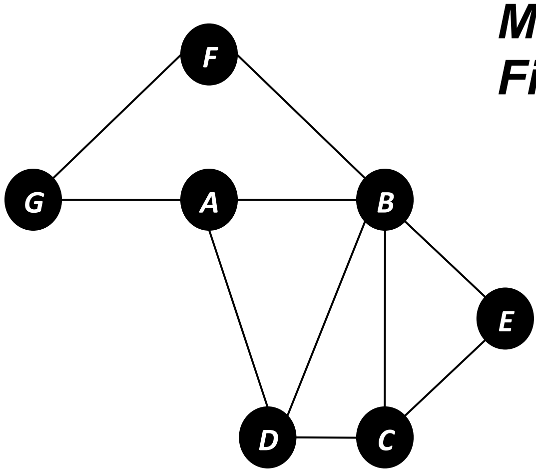
W=2

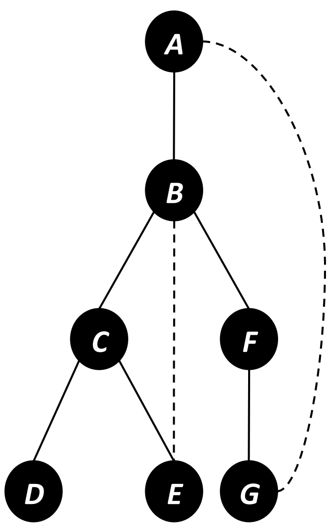


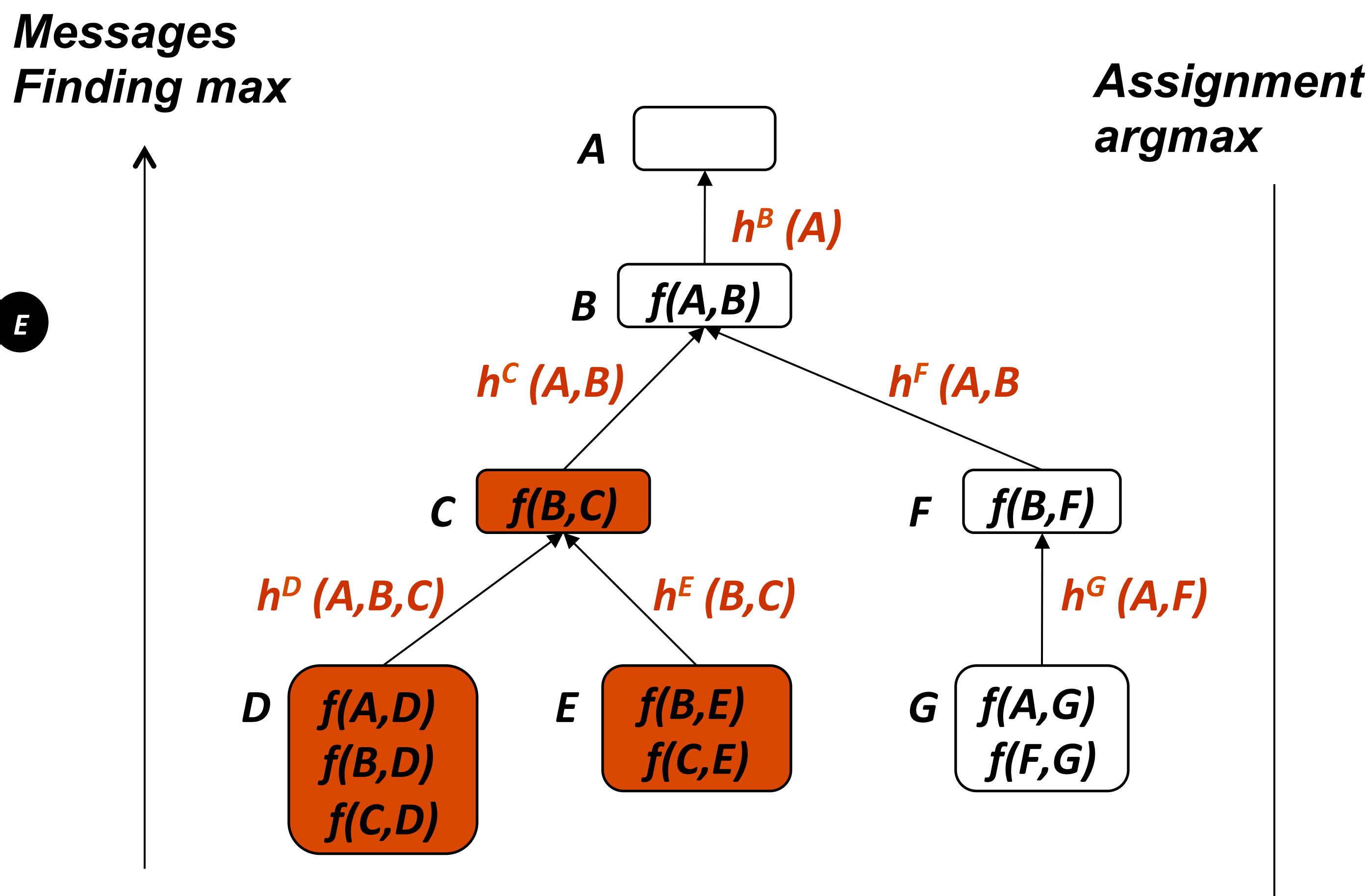
MPE* is an upper bound on MPE --U
Generating a solution yields a lower bound--L

Bucket Elimination

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



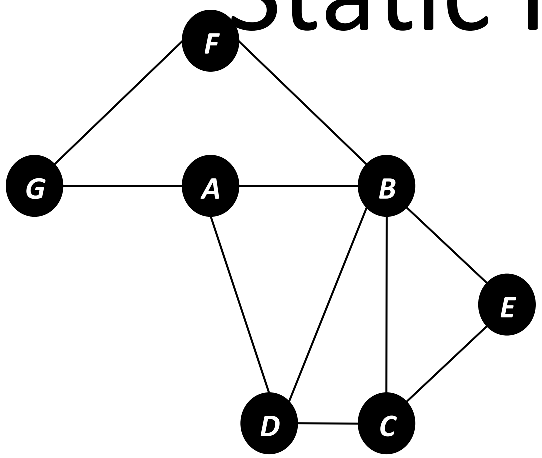


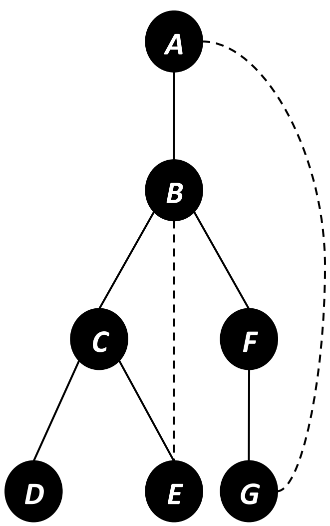


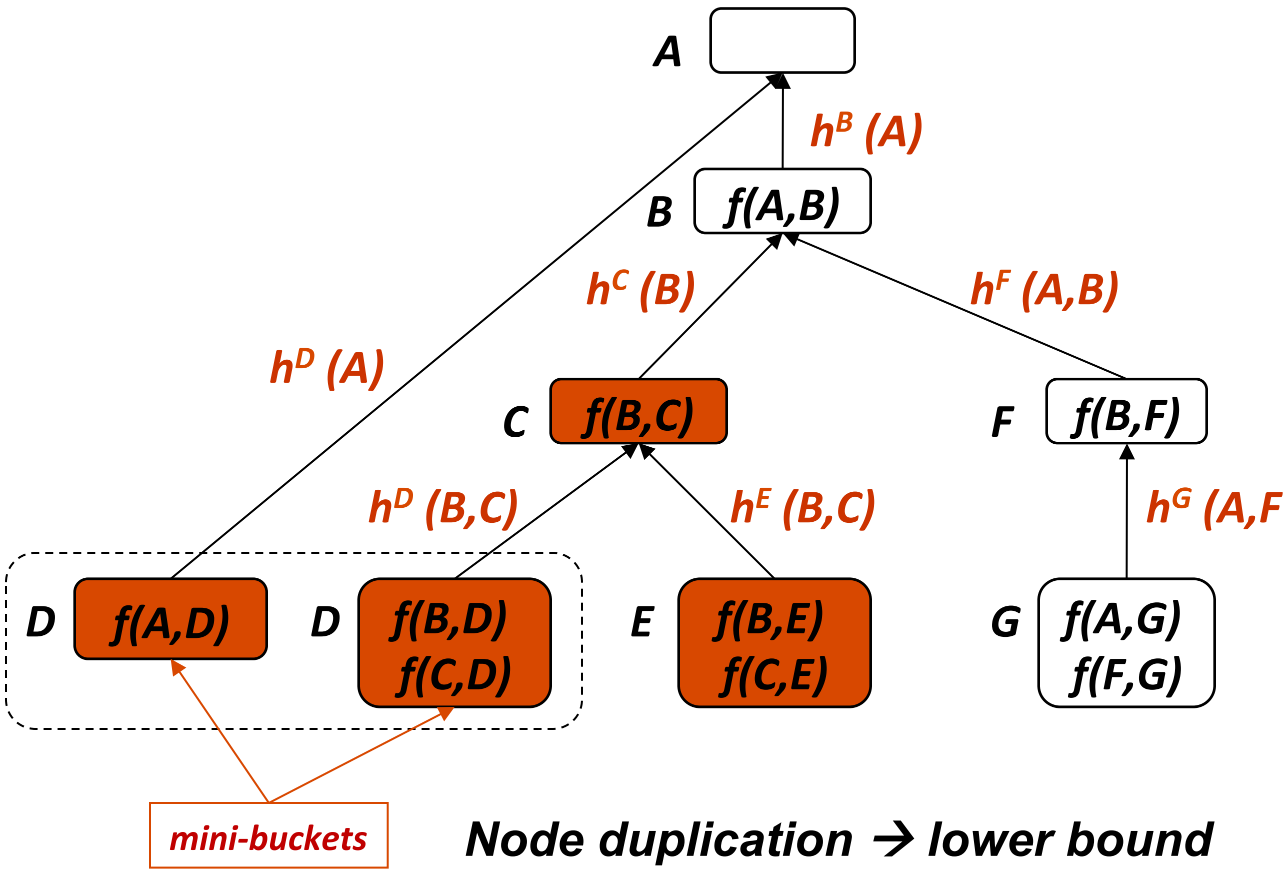
$$h^*(a, b, c) = h^D(a, b, c) + h^E(b, c)$$

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

Static Mini-Bucket Heuristics







Node duplication → lower bound

$$h(a, b, c) = h^D(a) + h^D(b, c) + h^E(b, c)$$
$$\leq h^*(a, b, c)$$

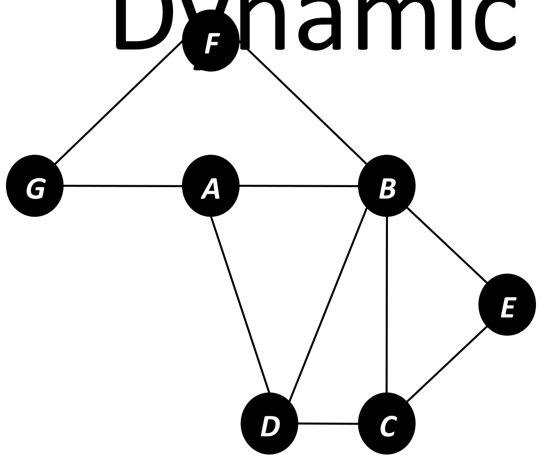
Ordering: (A, B, C, D, E, F, G)

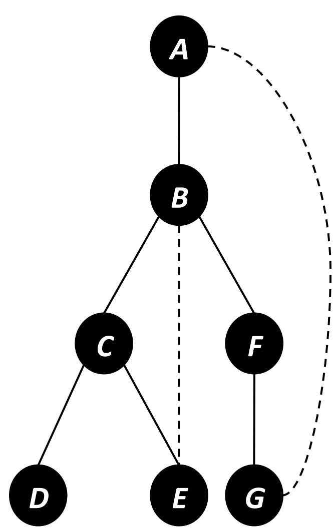
Mini-Bucket Elimination

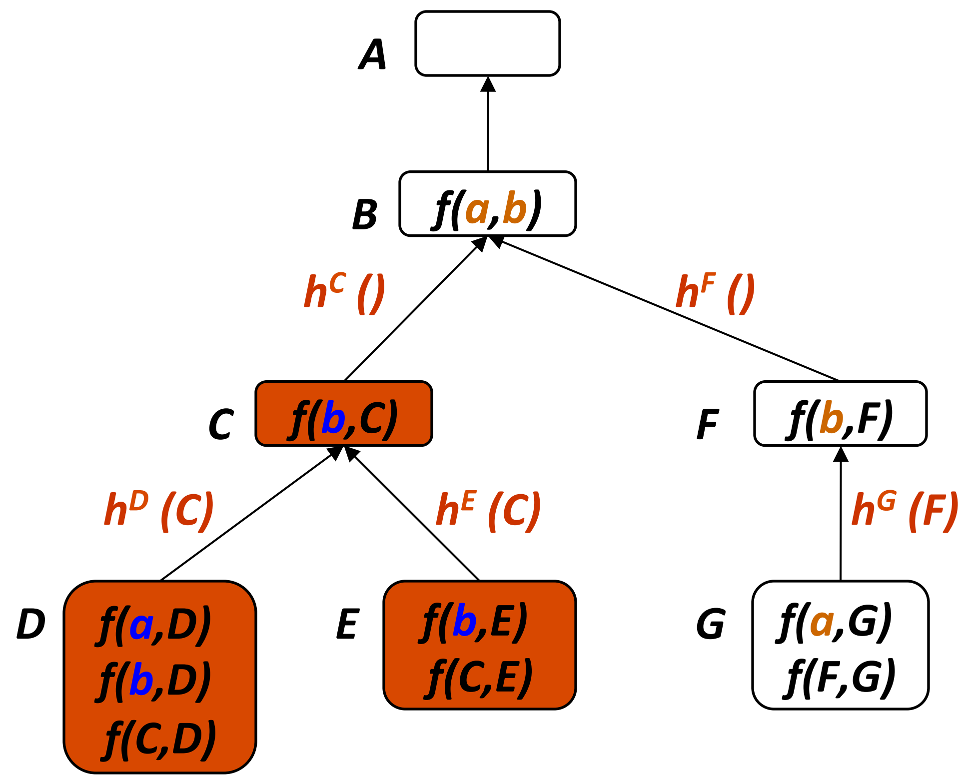
- Approximation of the Bucket Elimination algorithm by partitioning large buckets into "mini-buckets" which are processed separately (Dechter & Rish97)

- Properties
  - Parameterized by i-bound (controls complexity)
  - Computes a lower bound on the exact solution
  - Approximation improves with the i-bound

Dynamic Mini-Bucket Heuristics







$$h(a, b, c) = h^D(c) + h^E(c)$$
$$= h^*(a, b, c)$$

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

Outline

- Graphical Models: reasoning principles
- OR Search Trees
- AND/OR Search Spaces for GM
- AND/OR Branch-and-Bound and Best-First Search
- Lower Bounding Heuristics
- Experimental evaluation
- Current work

What are the empirical questions?

- What is the
  - Impact of AND/OR decomposition?
  - Impact of caching (Tree search vs graph search)?
  - The heuristic strength vs. search tradeoff.
  - impact of depth-first BB vs Best-first
  - The (w,h) of the pseudo-tree

Experiments

- Optimization task
  - Most Probable Explanation in belief networks
  - Optimal wcsp
- Algorithms
  - AOBB-C+SMB(i): AOBB w/ full caching and mini-buckets
  - AOBF-C+SMB(i): AOBF w/ full caching and mini-buckets
  - Samlam (Recursive Conditioning) (Darwiche01)
  - Superlink (genetic linkage analysis) (Fishelson&Geiger02)
  - ILOG CPLEX 11.0
- Benchmarks
  - ISCAS'89 circuits
  - Grid networks
  - Genetic linkage analysis
  - Mastermind games

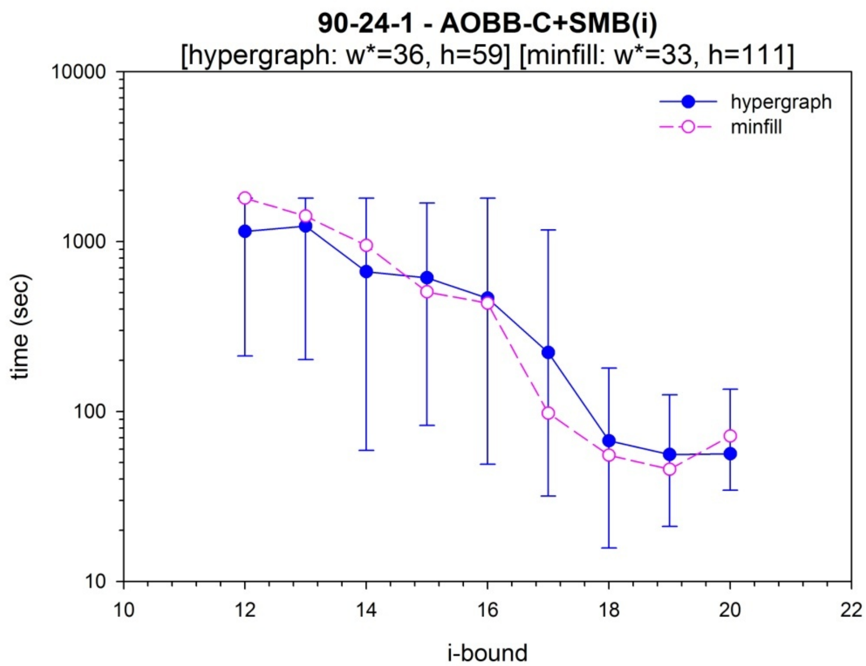
Grid Networks (BN)

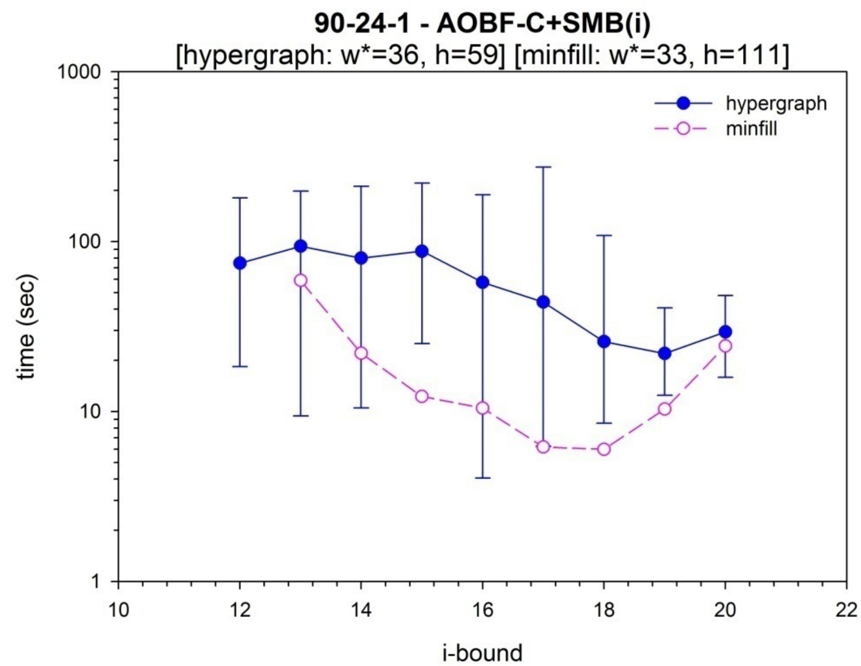
(Sang et al.05)

grid (w*, h) (n, e)	**Samlam**	**MBE(i)** **BB-C+SMB(i)** **AOBB+SMB(i)** **AOBB-C+SMB(i)** **AOBF-C+SMB(i)** i=12		**MBE(i)** **BB-C+SMB(i)** **AOBB+SMB(i)** **AOBB-C+SMB(i)** **AOBF-C+SMB(i)** i=14		**MBE(i)** **BB-C+SMB(i)** **AOBB+SMB(i)** **AOBB-C+SMB(i)** **AOBF-C+SMB(i)** i=16		**MBE(i)** **BB-C+SMB(i)** **AOBB+SMB(i)** **AOBB-C+SMB(i)** **AOBF-C+SMB(i)** i=18	
		time	nodes	time	nodes	time	nodes	time	nodes
		0.28		0.64		1.69		4.60	
90-24-1		-	-	-	-	-	-	-	-
(33, 111)	out	-	-	2338.67	24,117,151	1548.09	18,238,983	138.67	1,413,764
(576, 20)		-	-	1273.09	9,047,518	596.27	4,923,760	70.42	473,675
		out		21.94	75,637	10.59	33,770	**6.06**	5,144
		0.63		1.25		3.72		11.66	
90-34-1		-	-	-	-	-	-	-	-
(45, 153)	out	-	-	-	-	-	-	-	-
(1154, 80)		-	-	-	-	-	-	-	-
		out		out		**243.63**	596,978	270.88	667,013
		0.78		1.67		4.20		12.36	
90-38-1		-	-	-	-	-	-	-	-
(47, 163)	out	2032.33	6,835,745	-	-	807.38	2,850,393	568.69	2,079,146
(1444, 120)		969.02	2,623,971	1753.10	3,794,053	203.67	614,868	165.45	488,873
		101.69	174,786	103.80	146,237	54.00	95,511	**53.44**	78,431

Min-fill pseudo tree. Time limit 1 hour.

Impact of the Pseudo Trees and heuristic power





Runtime distribution for AOBB and AOBF with different i-bound for
The mini-bucket heuristics for pseudo-trees created by hypergraph and min-fill
over 20 independent runs for 90-24-1 grid instance.

Genetic Linkage Analysis

(Fishelson & Geiger02)

pedigree (w*, h) (n, d)	Samlam Superlink	MBE(i) BB-C+SMB(i) AOBB+SMB(i) AOBB-C+SMB(i) AOBF-C+SMB(i) i=12		MBE(i) BB-C+SMB(i) AOBB+SMB(i) AOBB-C+SMB(i) AOBF-C+SMB(i) i=14		MBE(i) BB-C+SMB(i) AOBB+SMB(i) AOBB-C+SMB(i) AOBF-C+SMB(i) i=16		MBE(i) BB-C+SMB(i) AOBB+SMB(i) AOBB-C+SMB(i) AOBF-C+SMB(i) i=18	
		time	nodes	time	nodes	time	nodes	time	nodes
		0.42		0.83		1.78		5.75	
ped30		-	-	-	-	-	-	-	-
(23, 118)	out	-	-	-	-	-	-	214.10	1,379,131
(1016, 5)	13095.83	10212.70	93,233,570	8858.22	82,552,957	-	-	34.19	193,436
		out		out		out		**30.39**	72,798
		0.58		2.31		7.84		33.44	
ped33		-	-	-	-	-	-	-	-
(37, 165)	out	2804.61	34,229,495	737.96	9,114,411	3896.98	50,072,988	159.50	1,647,488
(581, 5)	-	1426.99	11,349,475	307.39	2,504,020	1823.43	14,925,943	86.17	453,987
		out		140.61	407,387	out		**74.86**	134,068
		4.20		31.33		96.28		out	
ped42		-	-	-	-	-	-		
(25, 76)	out	-	-	-	-	-	-		
(448, 5)	561.31	-	-	-	-	2364.67	22,595,247		
		out		out		**133.19**	93,831		

Min-fill pseudo tree. Time limit 3 hours.

ISCAS'89 Benchmark (WCSP)

iscas (w*, h) (n, d)	MBE(i) BB-C+SMB(i) AOBB+SMB(i) AOBB-C+SMB(i) AOBF+SMB(i) i=8		MBE(i) BB-C+SMB(i) AOBB+SMB(i) AOBB-C+SMB(i) AOBF+SMB(i) i=10		MBE(i) BB-C+SMB(i) AOBB+SMB(i) AOBB-C+SMB(i) AOBF+SMB(i) i=12		MBE(i) BB-C+SMB(i) AOBB+SMB(i) AOBB-C+SMB(i) AOBF+SMB(i) i=14		toolbar toolbar-BTD	
	time	nodes	time	nodes	time	nodes	time	nodes	time	nodes
	0.08		0.08		0.14		0.28			
c499	-	-	-	-	1.53	4,495	6.20	35,314	-	-
(23, 55)	96.46	1,265,425	39.65	526,517	1.42	18,851	37.26	486,656	100.96	1,203,734
(499, 2)	19.28	99,906	7.36	40,285	0.47	2,401	5.83	34,708		
	3.91	14,049	2.45	8,816	**0.34**	1,032	2.52	8,755		
	0.16		0.19		0.38		0.94			
s1196	-	-	-	-	-	-	-	-	-	-
(54, 97)	-	-	1347.95	12,392,442	-	-	1949.37	15,775,180	376.35	1,276,514
(562, 2)	3347.38	13,554,137	503.30	2,425,152	2299.72	11,488,366	734.66	3,524,780		
	22.67	72,075	**2.89**	9,336	13.02	40,210	7.27	21,989		
	0.16		0.22		0.38		0.92			
s1238	-	-	-	-	-	-	-	-	-	-
(59, 94)	-	-	-	-	1722.53	18,302,873	1394.86	14,213,319	-	-
(541, 2)	1897.37	8,386,634	1682.99	7,431,223	281.05	1,350,933	248.27	1,220,658		
	34.09	137,960	29.41	111,205	12.31	53,095	**6.64**	26,101		

Min-fill pseudo tree. Time limit 1 hour.

UAI 2010 evaluation, 2008, 2006

- Toulbar2: INRA
  Summary: Toulbar2 is an open source exact anytime Weighted CSP solver using Branch and Bound and soft local consistency
  Team members: S. de Givry, D. Allouche, A. Favier, T. Schiex
  Additional contributors: M. Sanchez, S. Bouveret, H. Fargier, F. Heras, P. Jegou, J. Larrosa, K. L. Leung, S. N'diaye, E. Rollon, C. Terrioux, G. Verfaillie, M. Zytnicki
  Contact person: Thomas Schiex, Thomas.Schiex@toulouse.inra.fr
  Detailed description
- Daoopt: UCI Irvine
  Summary: "daoopt" and "daoopt.anytime" are based on AND/OR branch and bound graph search, with mini bucket heuristics and LDS (Limited Discrepancy Search) initialization.
  Team members: Lars Otten, Rina Dechter
  Additional Contributor: Radu Marinescu
  Contact person: Lars Otten, lotten@ics.uci.edu
  Detailed description
  Web-site: http://graphmod.ics.uci.edu

3rd in all 3 categorie
After Toolbar, Joris

What did we learn

- Use the highest heuristic memory allows
- Generate heuristic in pre-processing
- Use AOBB with caching (best-first run out of memory). We also have adaptive-caching
- Try to get the best induced-width pseudo-tree
- Use some good initial upper-bound from local search

Outline

- Graphical Models: reasoning principles
- OR Search Trees
- AND/OR Search Spaces for GM
- AND/OR Branch-and-Bound and Best-First Search
- Lower Bounding Heuristics
- Experimental evaluation
- Current work

Recent related highlights

- Improving Heuristics due to “moment matching” or soft arc-consistency. (Flerova, Ihler,Dechter, Otten, 2011)
- Anytime behavior of AOBB (Otten and Dechter, SOCS 2011)
- Improving treewidth (Kask et. Al. 2011)
- Improving mini-bucket parttiotning (Rollon 2010)

Bounding algorithms

non-iterative message-passing schemes

e.g. ***MBE*** *[Dechter, Rish 2003]*

iterative schemes using re-parametrization

e.g. ***MPLP*** *[Globerson et al. 2007],*

Max-sum diffusion *[Kovalevsky et al. 1975]*

Soft arc-consistency

[Schiex 2000, Bistarelli et al. 2000]

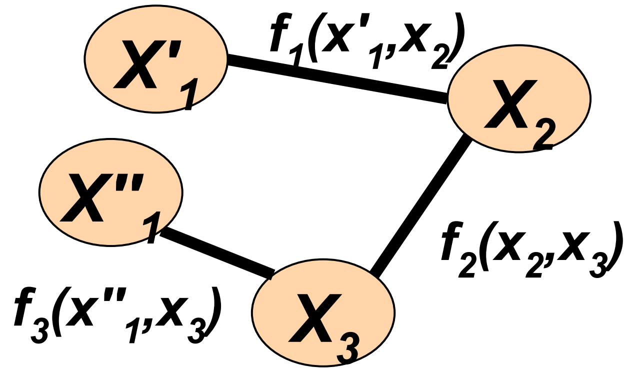
↓ ↓

Mini-Bucket with moment-matching

Intuition behind MBE-MM

How to make max-marginals equal?

Do cost shifting!

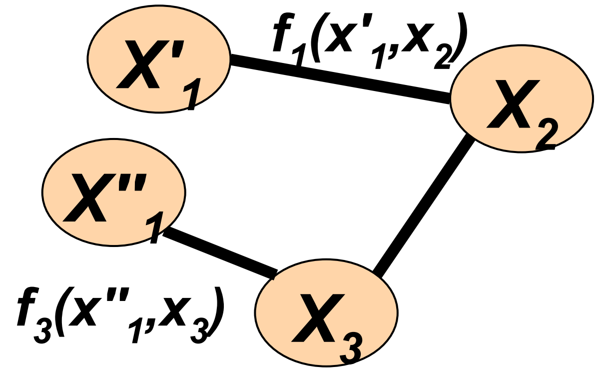


$$C_1(x_2, x_3) = \max_{x_1} f_1(x_1, x_2) \cdot f_3(x_1, x_3)$$

$$C_1(x_2, x_3) = \max_{x_1} f_1(x_1, x_2) g(x_1) \cdot f_3(x_1, x_3) / g(x_1)$$

Intuition behind MBE-MM

$$x_1^{*\prime} = \underset{x_1'}{argmax} \max_{x_2}[f_1(x_1', x_2) \cdot g(x_1')]$$



$$x_1^{*\prime\prime} = \underset{x_1''}{argmax} \max_{x_3}[f_3(x_1'', x_3)/g(x_1'')]$$

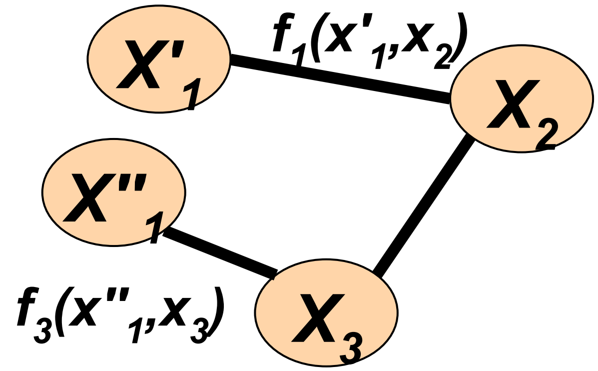
$$x_1^{*} = x_1^{*\prime} = x_1^{*\prime\prime}$$

$$\max_{x_2} f_1(x_1, x_2) g(x_1) = \max_{x_3} f_3(x_1, x_3)/g(x_1)$$

Intuition behind MBE-MM

$$F'_1(x_1, x_2) = f_1(x_1, x_2)\sqrt{\frac{\max\limits_{x_3} f_3(x_1, x_3)}{\max\limits_{x_2} f_1(x_1, x_2)}}$$

$$F''_1(x_1, x_3) = f_3(x_1, x_3)\sqrt{\frac{\max\limits_{x_2} f_1(x_1, x_2)}{\max\limits_{x_3} f_3(x_1, x_3)}}$$



Linear relaxation-based schemes (MPLP class, Globerson and Jakkola)

Find: $\boldsymbol{x} = (x_1, \ldots, x_n)$ to all the variables which maximizes the sum of the factors:

$$\mathrm{MAP}(\boldsymbol{\theta}) = \max_{\boldsymbol{x}} \sum_{i \in V} \theta_i(x_i) + \sum_{f \in F} \theta_f(\boldsymbol{x}_f). \tag{1.1}$$

Best upper bound by Equivalence preserving transformations:

$$\min_{\boldsymbol{\delta}} L(\boldsymbol{\delta}), \tag{1.2}$$

$$L(\boldsymbol{\delta}) = \sum_{i \in V} \max_{x_i} \Big(\theta_i(x_i) + \sum_{f: i \in f} \delta_{fi}(x_i) \Big) + \sum_{f \in F} \max_{\boldsymbol{x}_f} \Big(\theta_f(\boldsymbol{x}_f) - \sum_{i \in f} \delta_{fi}(x_i) \Big).$$

$\delta_{fi}(x_i)$ ***Is the cost shifted from f to value x_i of X_i.***

There are several variations of scheme computing the optimizing shifts based on partial gradient descent, which differ by what is being kept constant. The 1.2 tas Dual of a linear relaxation of the original problem.

Grids

The dash '-' indicates that the was no solution within 24 hours

Time in seconds for grid instances

All grids are binary

Instances (n,w,h)	AOBB-MBE(z) AOBB-MBE-MM(z) AOBB-MPLP(z) z-bound=10 time	AOBB-MBE(z) AOBB-MBE-MM(z) AOBB-MPLP(z) z-bound=5 time	AOBB-MBE(z) AOBB-MBE-MM(z) AOBB-MPLP(z) z-bound=3 time
50-12-5	0	2	9
144,15,48	0	0	1
	0	1	0
50-14-5	1	64	211
196,18,64	0	6	25
	0	6	24
50-15-5	1	42	431
225,19,76	0	22	24
	0	29	24
50-16-5	97	6759	14047
256,21,79	1	257	11918
	1	209	11872
50-17-5	18	2674	19951
289,22,84	0	70	2293
	0	44	2302
50-18-5	1131	—	—
324,24,84	10	47196	—
	10	19697	—
50-19-5	4664	—	—
361,25,93	9	8808	—
	11	5375	864000
50-20-5	3589	—	—
400,27,97	11	28985	—
	26	6529	—
75-16-5	7	245	2457
256,21,73	0	32	511
	1	37	521
75-17-5	8	279	568
289,22,78	1	186	292
	1	133	301

The dash '-' indicates that the was no solution within 24 hours

Time in seconds for grid instances

All grids are binary

Instances (n,w,h)	AOBB-MBE(z) AOBB-MBE-MM(z) AOBB-MPLP(z) z-bound=10 time	AOBB-MBE(z) AOBB-MBE-MM(z) AOBB-MPLP(z) z-bound=5 time	AOBB-MBE(z) AOBB-MBE-MM(z) AOBB-MPLP(z) z-bound=3 time
90-20-5 400,27,99	14 1 0 9	1199 585 389	7281 5575 5317
90-21-5 441,28,106	15 0 1	1585 861 593	7722 9064 9054
90-22-5 484,30,109	50 6 1	2327 1172 604	27283 17130 17060
90-23-5 529,31,116	564 10 17	70188 29635 25511	— — —
90-24-5 576,33,110	2393 73 68	— — 27818	— — —
90-25-5 625,34,132	2496 223 277	— — —	— — —
90-26-5 676,36,136	386 21 16	36469 7077 4000	— 70798 70290
90-30-5 900,42,151	— 25439 21895	— — —	— — —

Runtime for pedigree instances for AOBB

Instances (n,k,w,h)	AOBB-MBE(z) AOBB-MBE-MM(z) AOBB-MPLP(z) z-bound=10	AOBB-MBE(z) AOBB-MBE-MM(z) OBB-MPLP(z) z-bound=8	AOBB-MBE(z) AOBB-MBE-MM(z) OBB-MPLP(z) z-bound=6	AOBB-MBE(z) AOBB-MBE-MM(z) OBB-MPLP(z) z-bound=4
pedigree1 298,4,15,48	0 1 4	1 0 7	0 0 11	0 0 42
pedigree13 88,3,32,102	— 57583 70328	— — —	— — —	— — —
pedigree19 693,5,26,95	— — —	— — —	— — —	— — —
pedigree20 387,5,22,60	44 25 87	137 112 262	167 378 582	4460 10805 11203
pedigree23 309,5,25,51	4 0 9	13 2 24	22 3 20	45 31 89
pedigree25 993,5,25,69	58 0 1	145 4 3	1303 36 48	— 13321 4670
pedigree30 1015,5,21,108	109 21 34	246 36 99	1690 442 508	13198 — —
pedigree31 1006,5,30,85	— — —	— — —	— — —	— — —

The dash '-' indicates that the was no solution within 24 hours

Runtime Runtime for pedigree instances for AOBB

Instances (n,k,w,h)	AOBB-MBE(z) AOBB-MBE-MM(z) AOBB-MPLP(z) z-bound=10	AOBB-MBE(z) AOBB-MBE-MM(z) OBB-MPLP(z) z-bound=8	AOBB-MBE(z) AOBB-MBE-MM(z) OBB-MPLP(z) z-bound=6	AOBB-MBE(z) AOBB-MBE-MM(z) OBB-MPLP(z) z-bound=4
pedigree33 581,4,28,98	89 5 8	177 7 8	201 260 287	24142 1958 2076
pedigree37 726,5,21,56	6 1 0	13 0 1	33 8 6	298 174 145
pedigree39 953,5,21,76	136 15 17	732 29 26	2871 294 377	30724 8315 9100
pedigree50 478,6,17,47	6 17 46	16 11 886	39 47 12146	25440 — —
pedigree51 871,5,39,98	— — —	— — —	— — —	— — —
pedigree7 867,4,32,90	— 1987 4975	— 5993 13211	— 46261 54000	— — —
pedigree9 935,7,27,100	46434 1206 2161	— 7086 9397	— — —	— — —

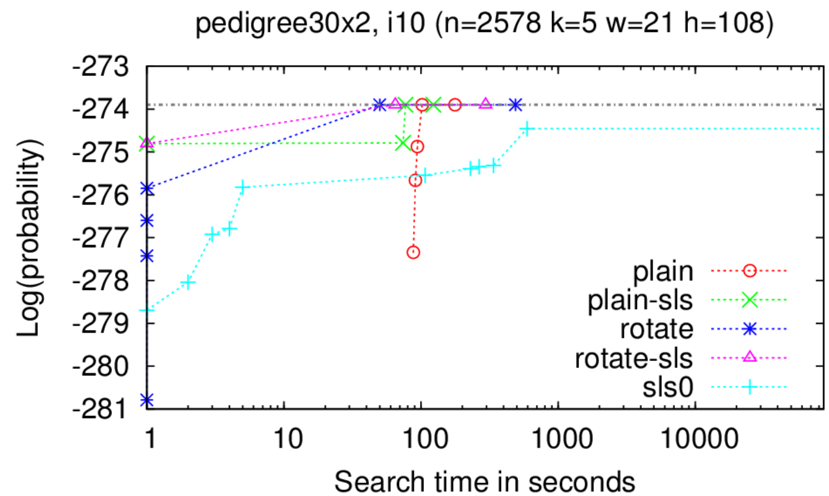
The dash '-' indicates that the was no solution within 24 hours

Recent related highlights

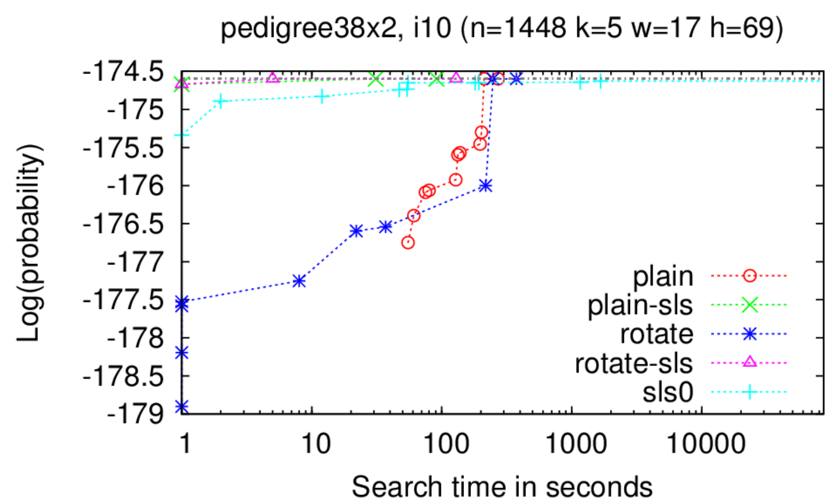
- Improving Heuristics due to “moment matching” or soft arc-consistency. (Flerova, Ihler,Dechter, Otten, 2011)
- Anytime behavior of AOBB (Otten and Dechter, SOCS 2011)
- Improving treewidth (Kask et. Al. 2011)
- Improving mini-bucket parttiotning (Rollon 2010)

Select Anytime Results

- AOBB+SLS receives initial boost

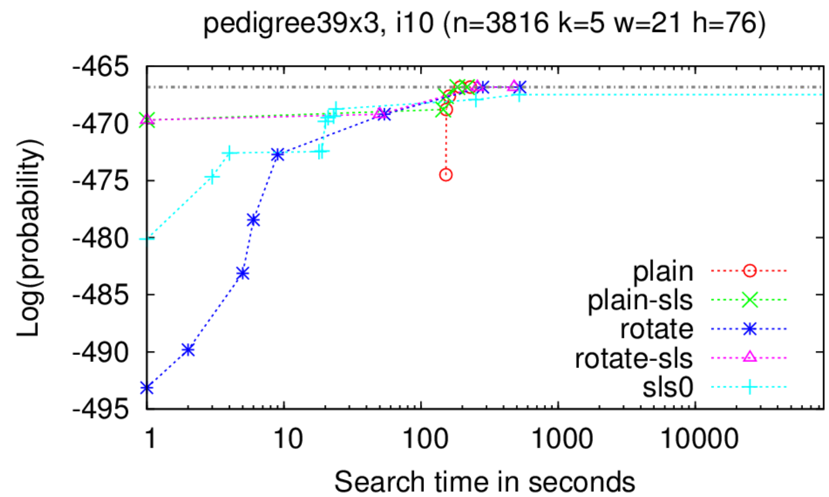


- AOBB+SLS gets to optimality faster

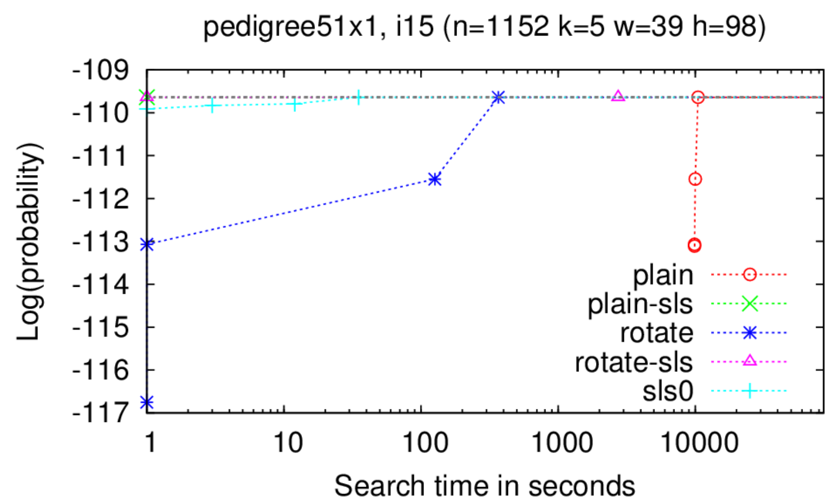


Select Detailed Results

- AOBB+SLS finds optimal solution, SLS doesn't

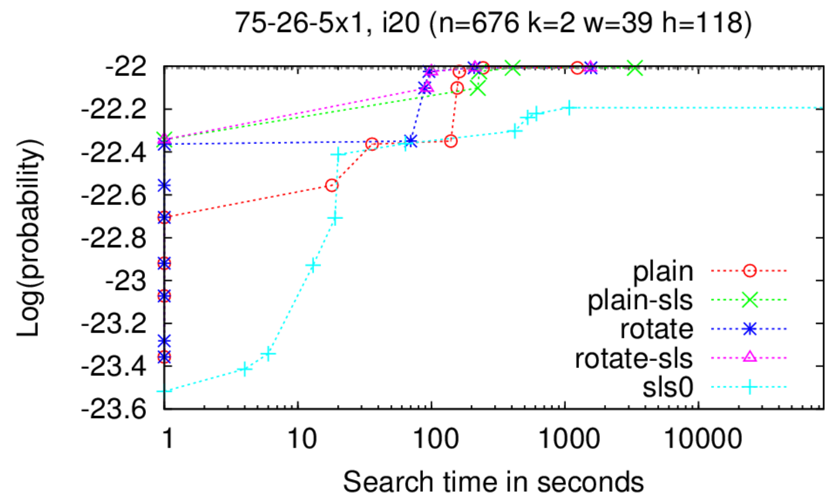


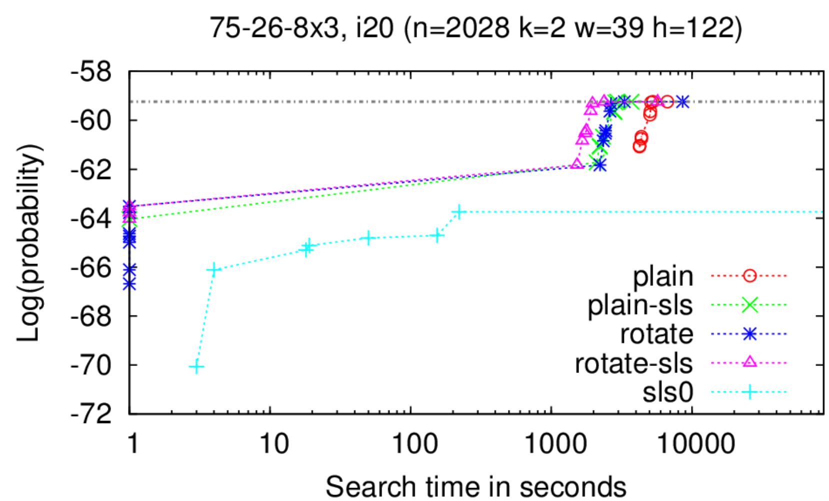
- AOBB+SLS finds optimum right away



Select Detailed Results

- All AOBB variants outperform SLS, reach optimality
- Rotating AOBB receives boost from initial SLS





Software

- AND/OR search algorithms
- Bucket-tree elimination
- Generalized belief propagation
- Samplesearch sampling

are available at:

- http://graphmod.ics.uci.edu/group/Software

Thank you!