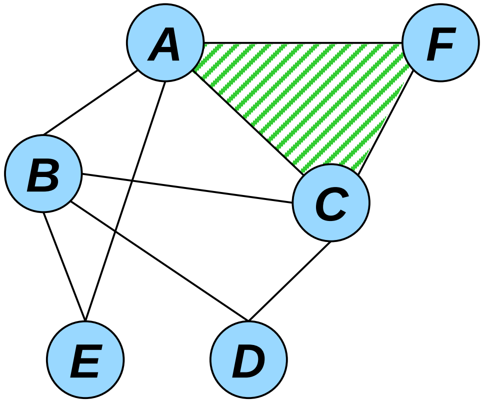
Probabilistic Reasoning Meets Heuristic Search

Rina Dechter

Bren School of Information and Computer Sciences, University of California, Irvine

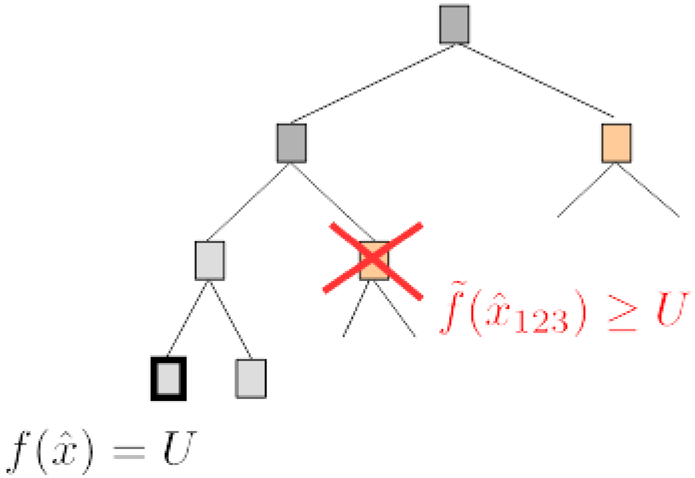


Main collaborator:
Radu Marinescu
Lars Otten
Alex Ihler
Kalev Kask
Robert Mateescu
Irina Rish

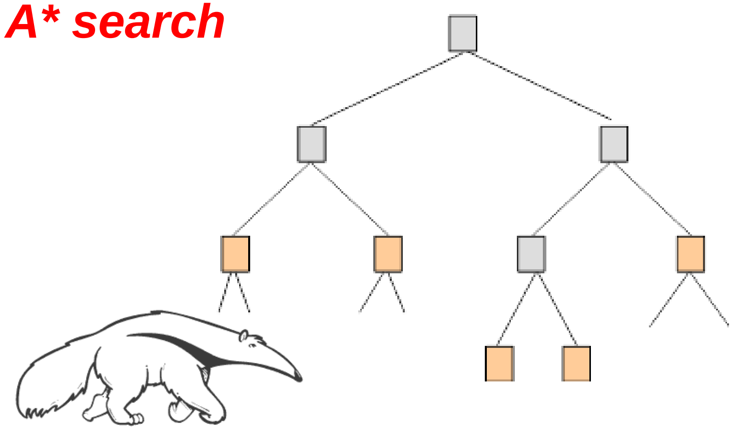
Search Swallows Inference

- Heuristic Search

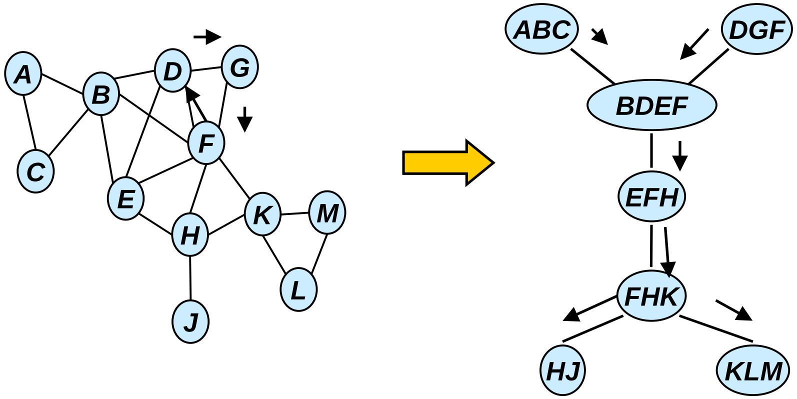
Branch-and-Bound



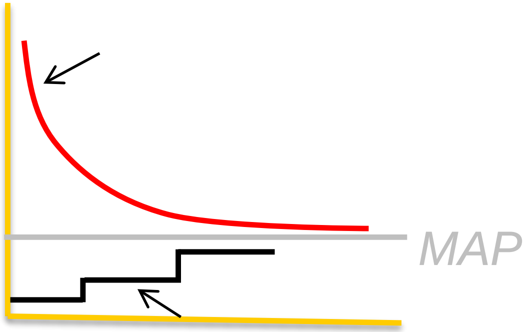
A* search



- Probabilistic reasoning, graphical models

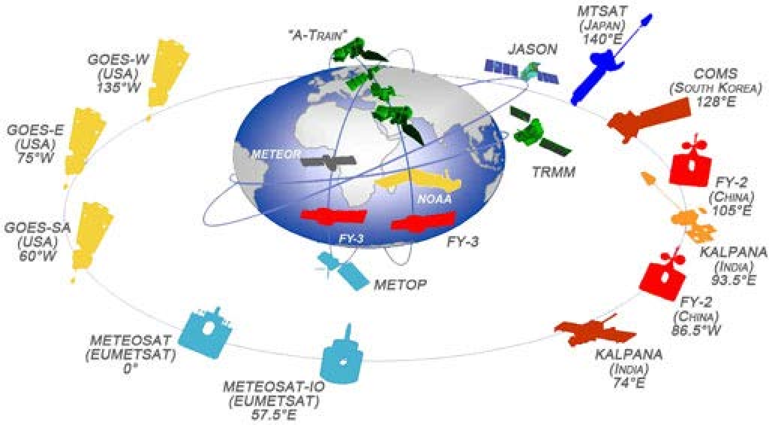


Anytime algorithms.



Combinatorial Optimization

Planning & Scheduling



Find an optimal schedule for the satellite that maximizes the number of photographs taken, subject to on-board recording capacity

Computer Vision

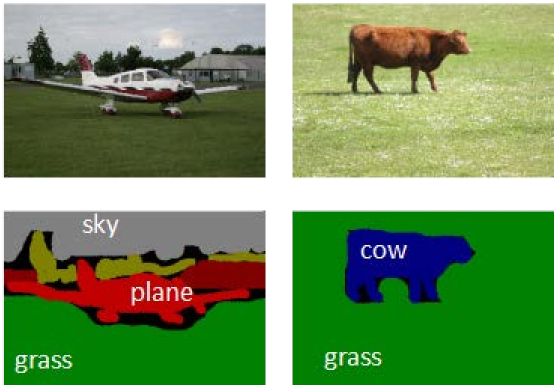
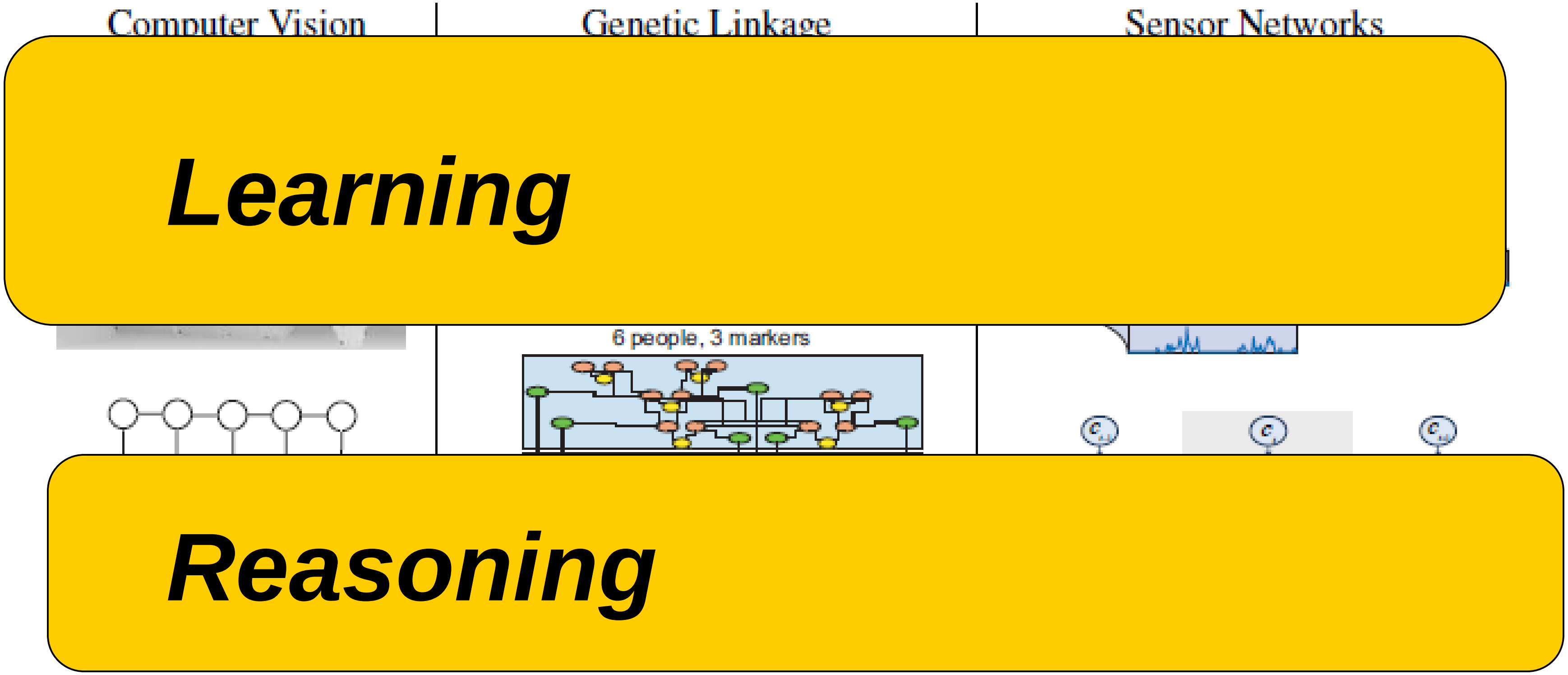


Image classification: label pixels in an image by their associated object class

[He et al. 2004; Winn et al. 2005]

Sample Applications for Graphical Models



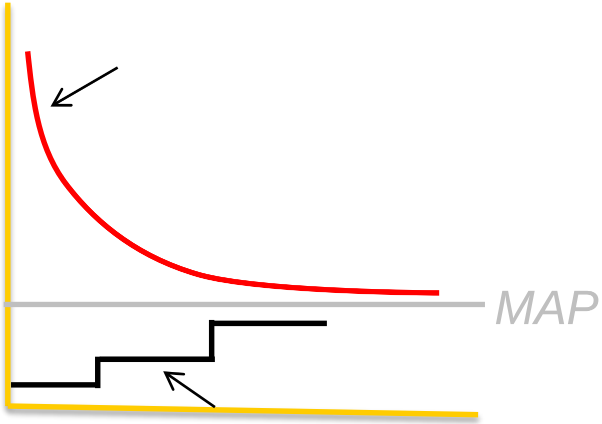
Learning

Reasoning

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

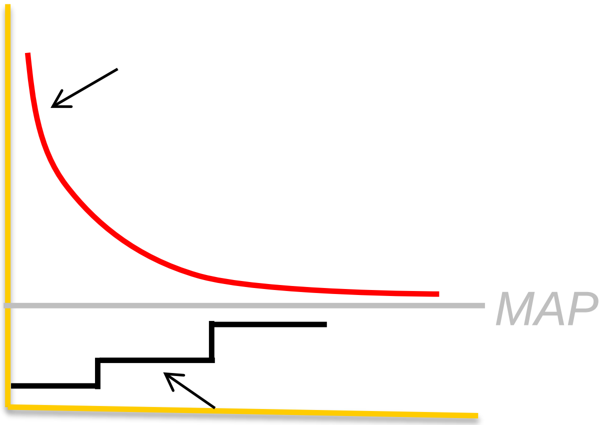
Outline

- Graphical models, Queries, Inference vs search
- Inference Algorithms: bucket-elimination
- AND/OR search spaces
- Bounded Inference: a) mini-bucket, b) cost-shifting
- Generating heuristics using mini-bucket elimination
- AND/OR Heuristic Search for Map and Marginal Map
- Conclusion



Outline

- Graphical models, Queries, Inference vs search
- Inference Algorithms: bucket-elimination
- AND/OR search spaces
- Bounded Inference: a) mini-bucket, b) cost-shifting
- Generating heuristics using mini-bucket elimination
- AND/OR Heuristic Search for Map and Marginal Map
- Conclusion



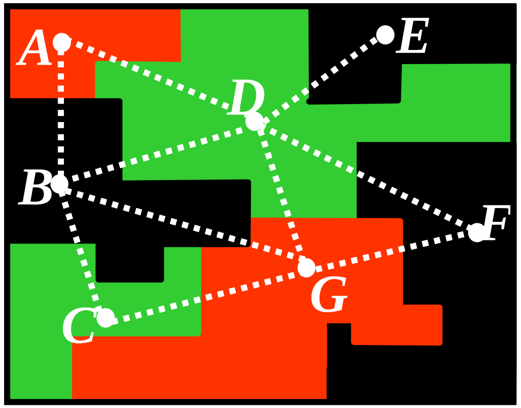
Graphical Models, Queries, Algorithms

***Any collection of local functions over a subset of variable
Is a graphical model***

Constraint Satisfaction/Satisfiability

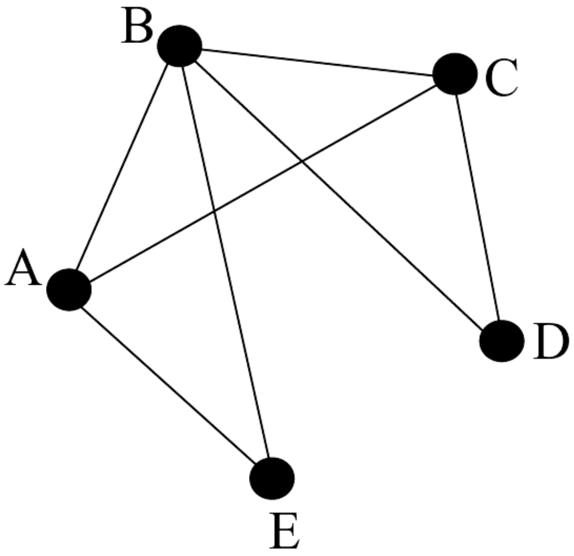
Constraint Networks

- Variables - countries (A,B,C,etc.)
- Values - colors (red, green, blue)
- Constraints: **A ≠ B, A ≠ D, D ≠ E,** *etc.*



Propositional Satisfiability

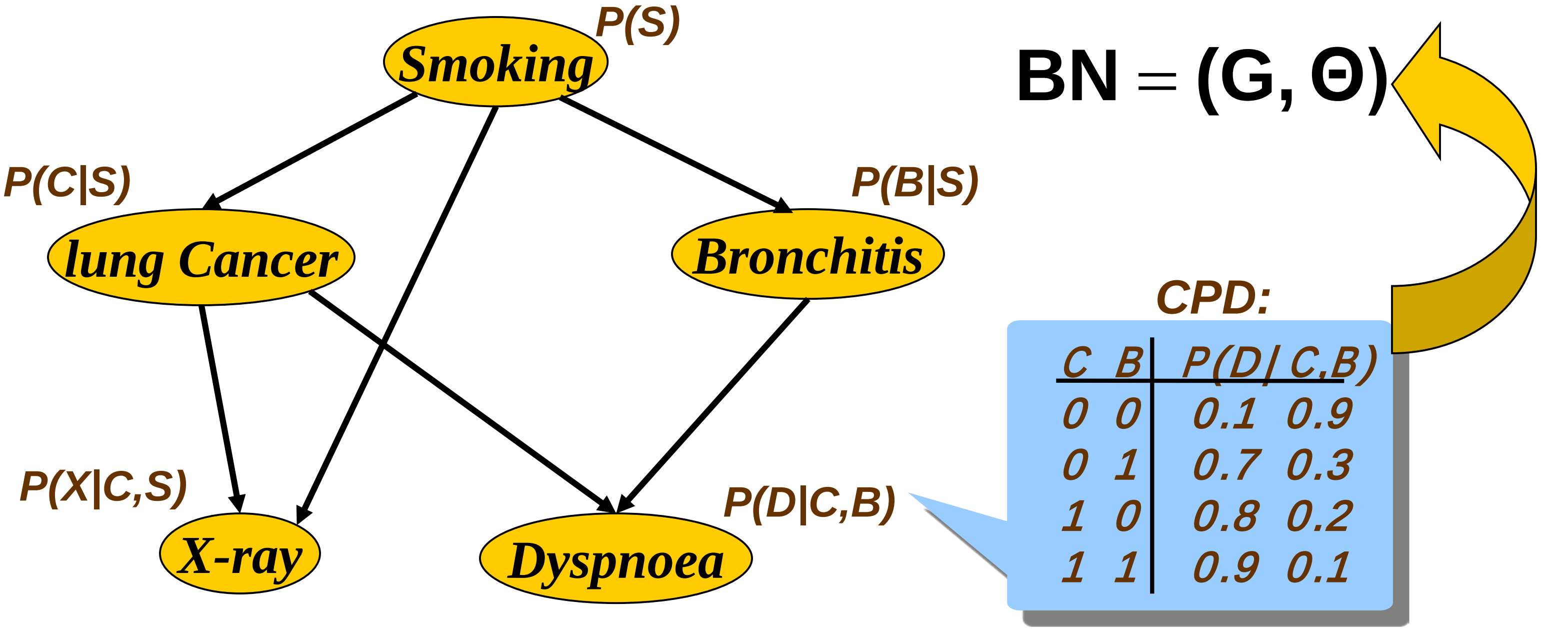
- φ = {(¬*C*), (*A* v *B* v *C*), (¬*A* v *B* v *E*), (¬*B* v *C* v *D*)}.



***Semantics**: set of all solutions*

***Primary task**: find a solution*

Bayesian Networks (Pearl, 1988)



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

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

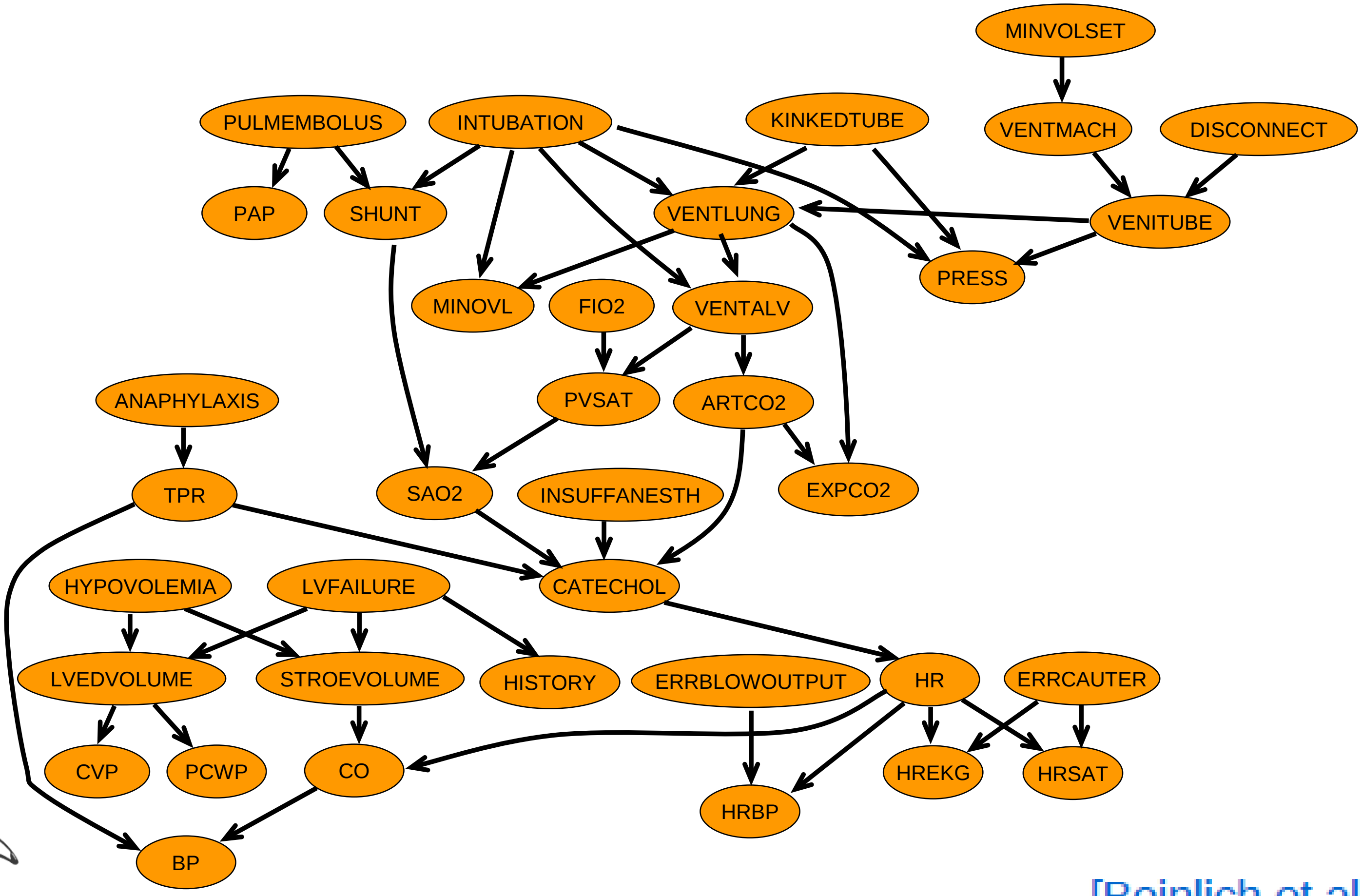
***Belief Updating*:**

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

***MAP* = argmax P(S)· P(C|S)· P(B|S)· P(X|C,S)· P(D|C,B)**

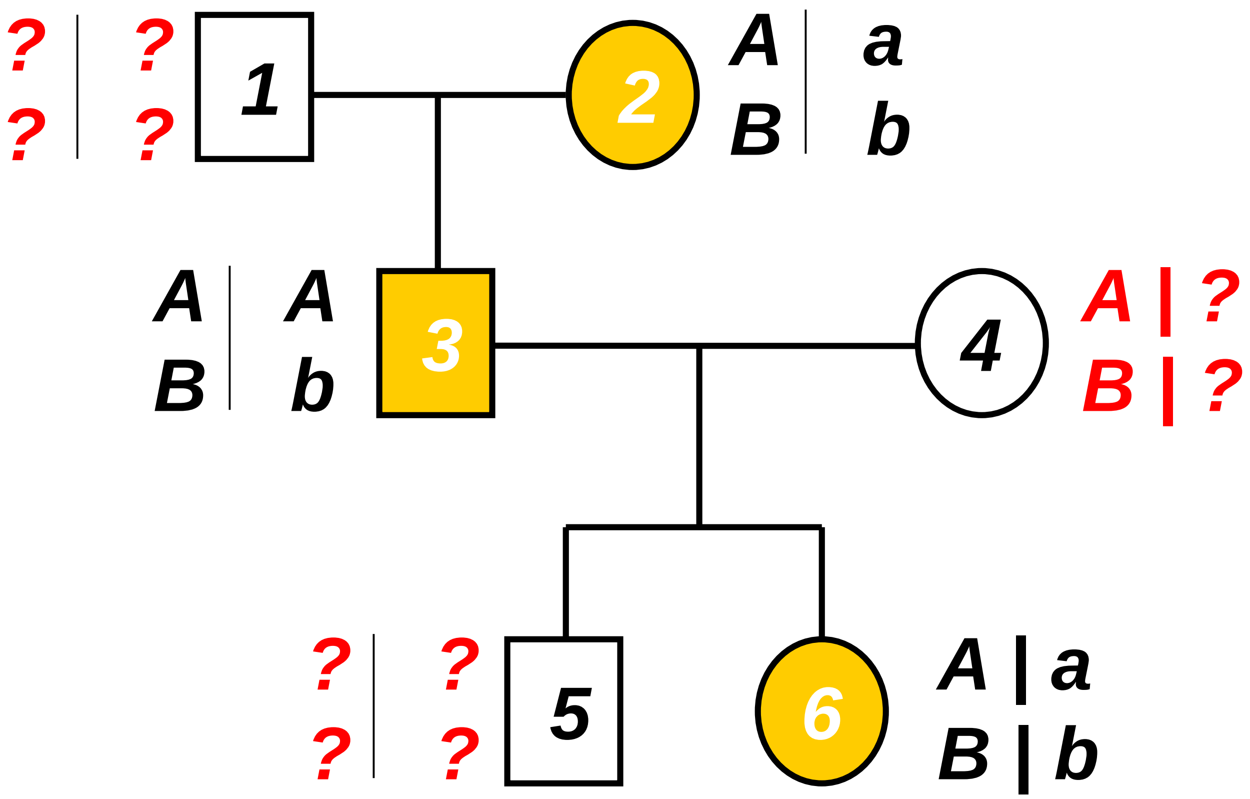
Monitoring Intensive-Care Patients

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



[Beinlich et al., 1989]

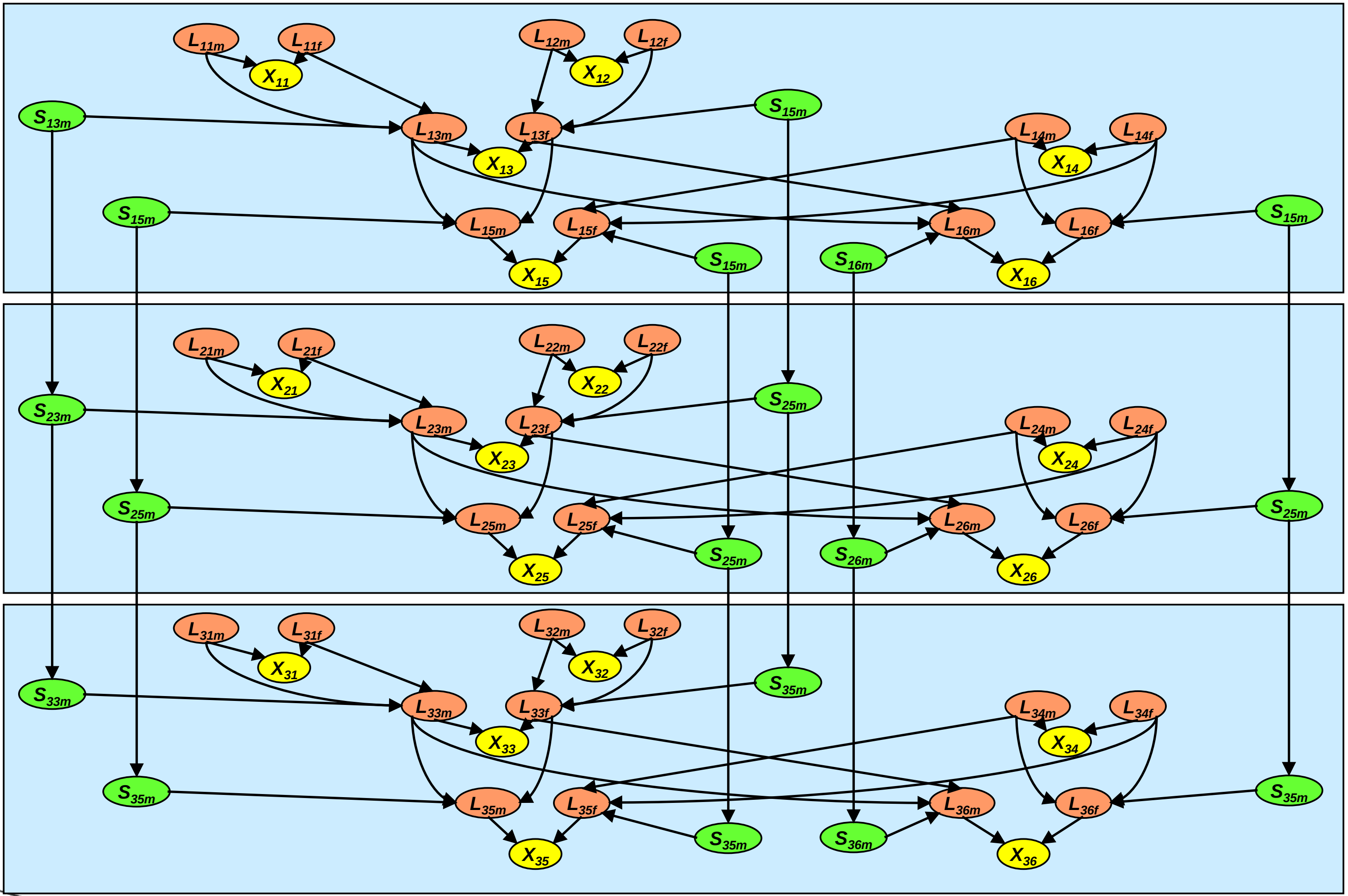
Genetic Linkage Analysis



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

e.g., [Lauritzen & Sheehan, 2003]

Pedigree: 6 people, 3 markers

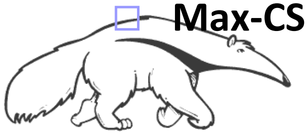


Graphical Models

- A graphical model **(X,D,F)**:
  - □ **X** = $\{X_1,\ldots X_n\}$ variables
  - □ **D** = $\{D_1, \ldots D_n\}$ domains
  - □ **F** = $\{f_1,\ldots,f_r\}$ functions
    (constraints, CPTS, CNFs …)

- Operators:
  - □ combination : Sum, product, join
  - □ Elimination: projection, sum, max/min

- Tasks:
  - □ **Belief updating**: $\Sigma_{X\setminus Y}\prod_j P_i$
  - □ **MPE\MAP**: $\max_X \prod_j P_j$
  - □ **Marginal MAP**: $\max_Y \Sigma_{X\setminus Y}\prod_j P_j$
  - □ **CSP**: $\prod_X \times_j C_j$
  - □ **Max-CSP**: $\min_X \Sigma_j F_j$

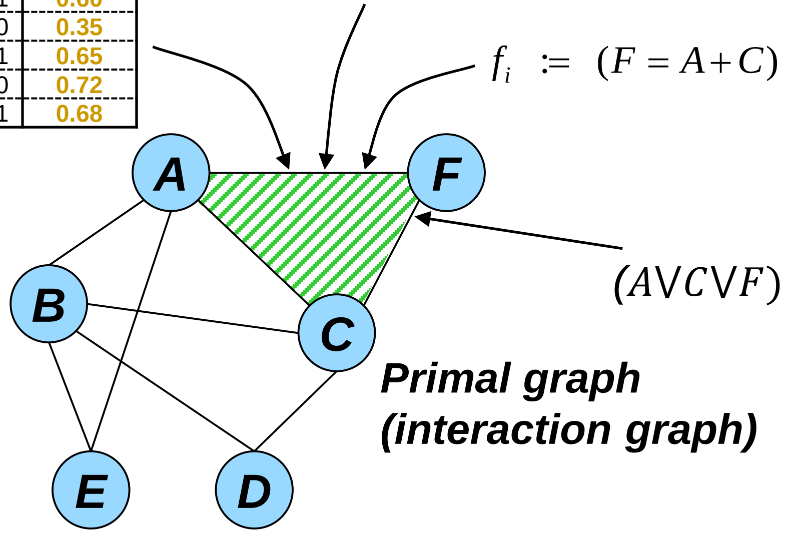


Conditional Probability Table (CPT)

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

Relation

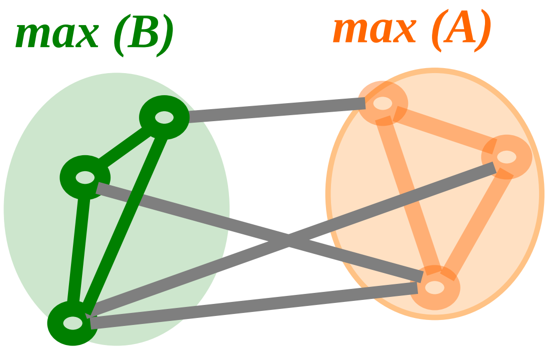
A	C	F
red	green	blue
blue	red	red
blue	blue	green
green	red	blue



Primal graph (interaction graph)

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

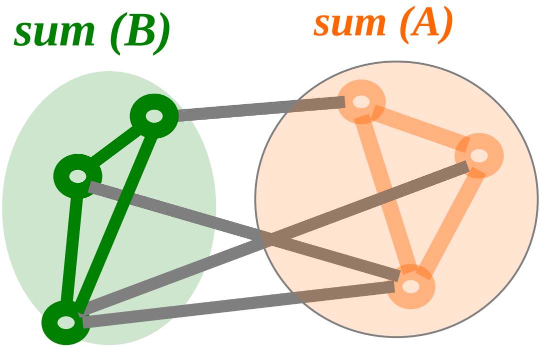
Queries



$$X = A \cup B$$

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

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

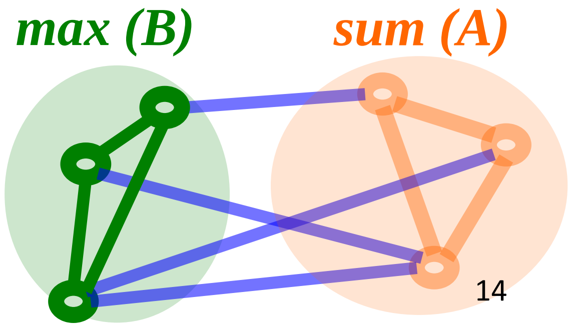


- **Likelihood queries: Partition function**

$$P(x) = \frac{1}{Z} \prod_{x_\alpha} \varphi_\alpha(x) \qquad Z = \sum_{x_A, x_B} \prod_{x_\alpha} \varphi_\alpha$$

- **Marginal MAP**

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



Also Satisfiability and Expected utility

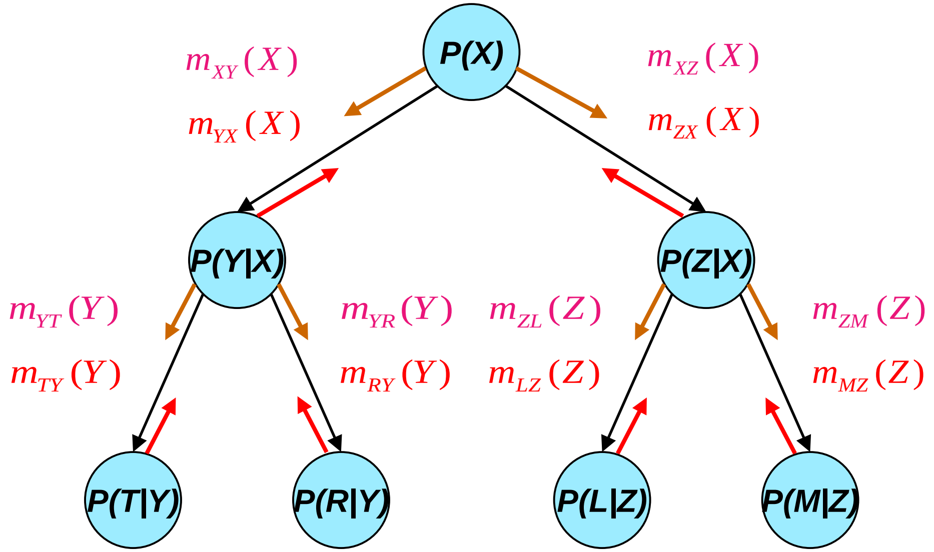
Example Domains for Graphical Models

- Natural Language Processing
  - □ Information extraction, semantic parsing, translation, topic models, …
- Computer Vision
  - □ Object recognition, scene analysis, segmentation, tracking, …
- Computational Biology
  - □ Pedigree analysis, protein folding / binding / design, sequence matching, …
- Networks
  - □ Webpage link analysis, social networks, communications, citations, …
- Robotics
  - □ Planning and decision making, …
- …

Tree-solving is Easy

Belief updating (sum-prod)

CSP – consistency (projection-join)



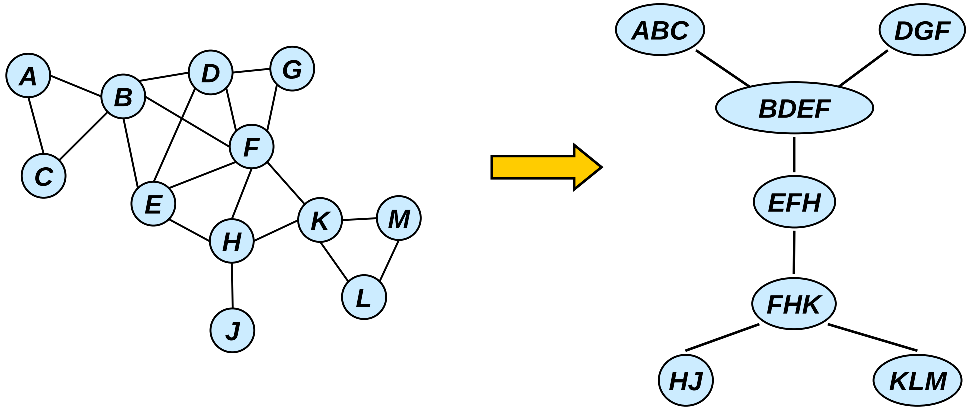
Dynamic Programming, Inference

MPE (max-prod)

#CSP (sum-prod)

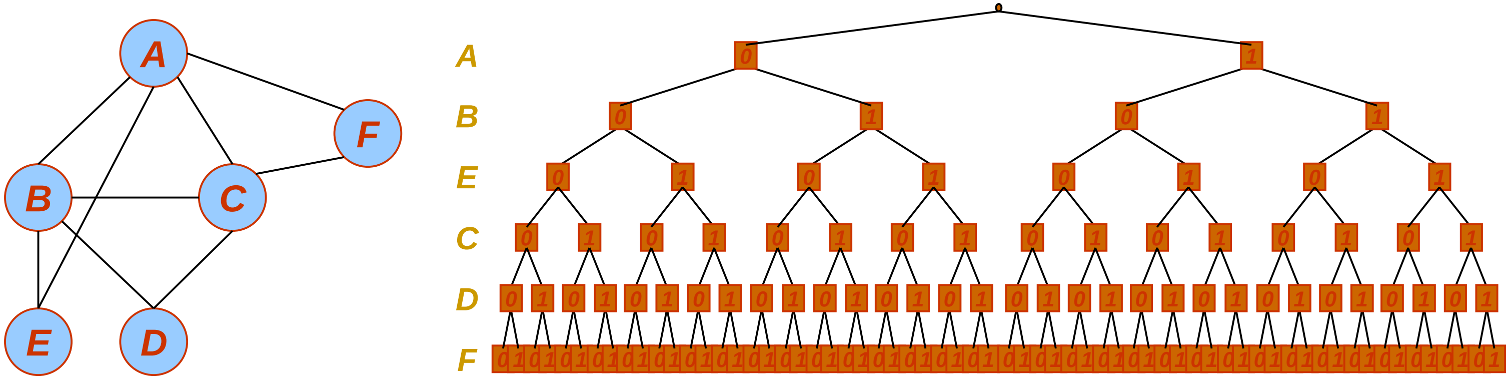
Trees are processed in linear time and memory
Message-passing

Inference vs Conditioning-Search



Inference

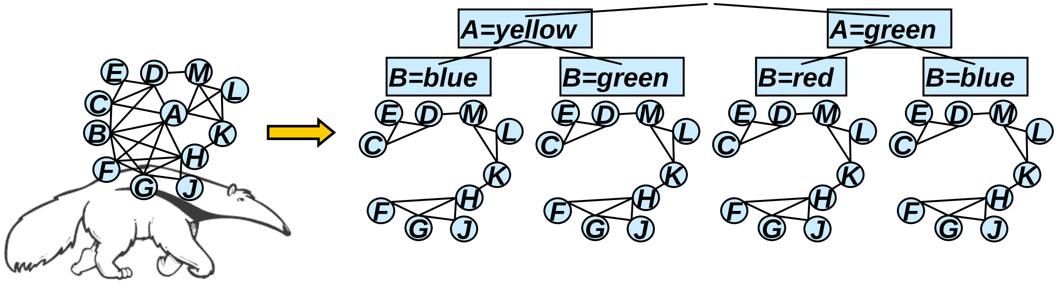
exp(w) time/space*



Search

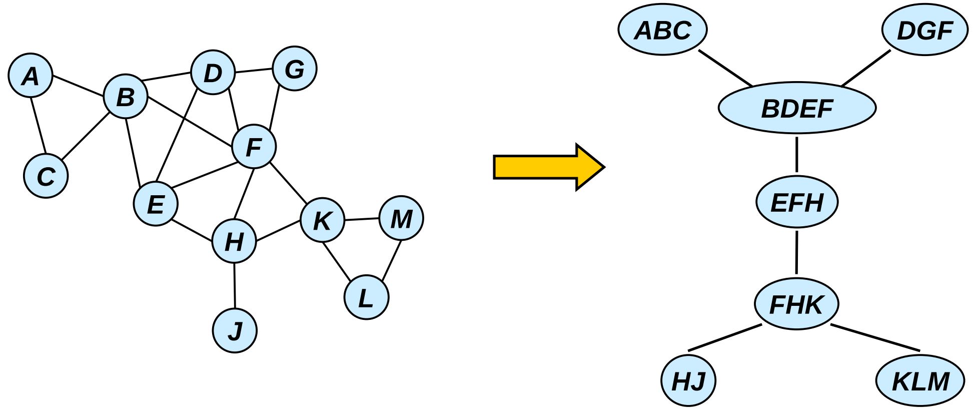
Exp(n) time

O(n) space



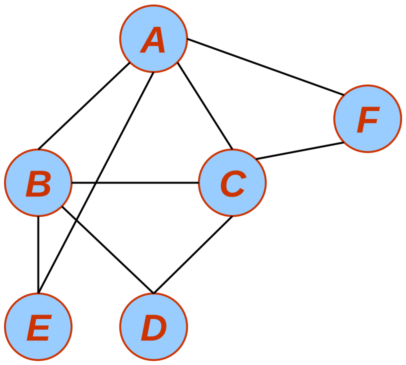
Search+inference:
Space: exp(w) *w: user*
Time: exp(w+c(w)) *controlled*

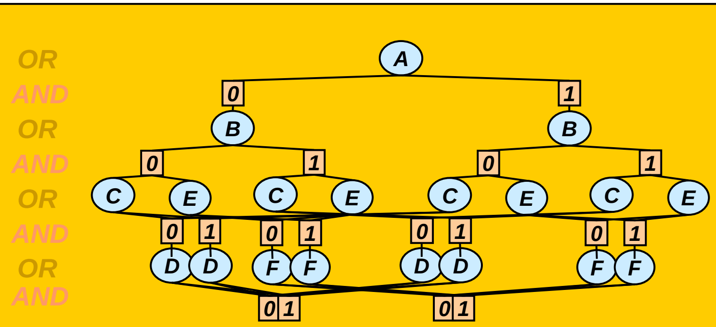
Inference vs conditioning-search



Inference

exp(w) time/space*



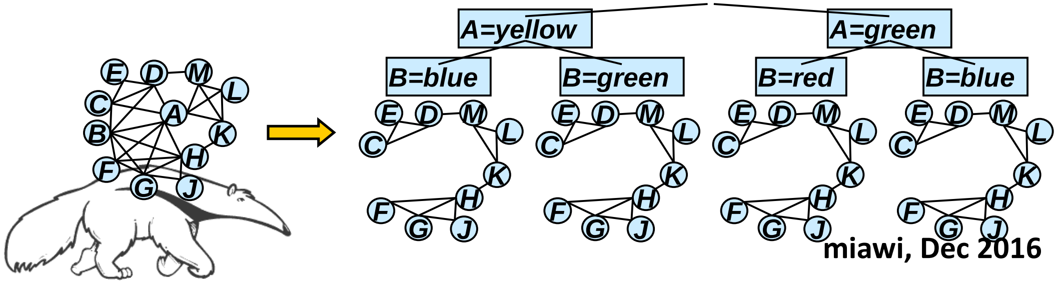


Context minimal AND/OR search graph

18 AND nodes

Search

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

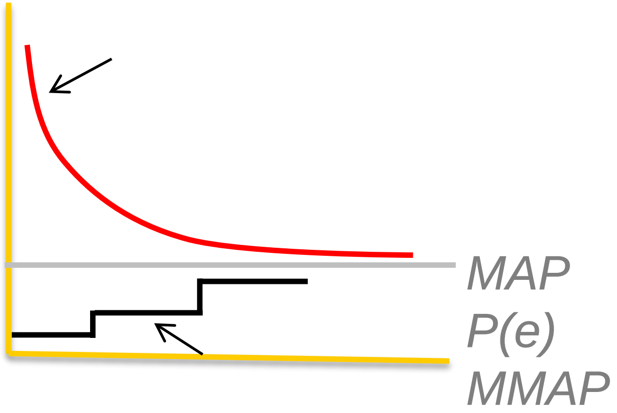


Search+inference:
Space: exp(q)
Time: exp(q+c(q))

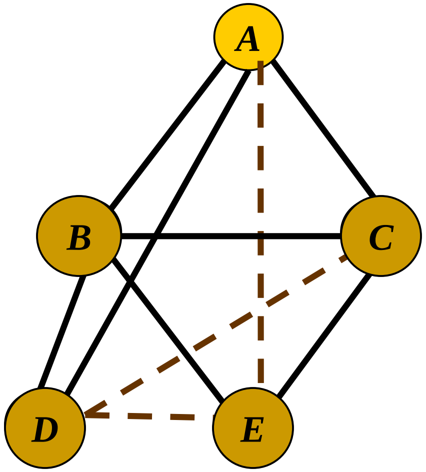
q: user controlled

Outline

- Graphical models, Queries, Inference vs search
- Inference Algorithms: bucket-elimination
- AND/OR search spaces
- Bounded Inference: a) mini-bucket, b) cost-shifting
- Generating heuristics using mini-bucket elimination
- AND/OR Heuristic Search for Map and Marginal Map
- Conclusion



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



"primal" graph

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

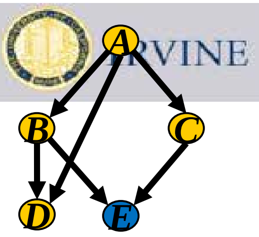
$$\sum_{e=0,d,c,b} P(a)\underbrace{P(b|a)}P(c|a)\underbrace{P(d|b,a)P(e|b,c)}$$

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

Variable Elimination

Finding Marginals by Bucket elimination

Algorithm *BE-bel* (Dechter 1996)



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

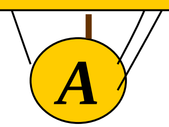
$$\sum_{b}\prod \longleftarrow$$ ***Elimination operator***

Time and space exponential in the induced-width / treewidth

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

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

induced width
(max clique size)



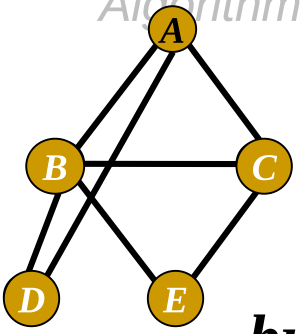
P(e=0)

P(a|e=0)

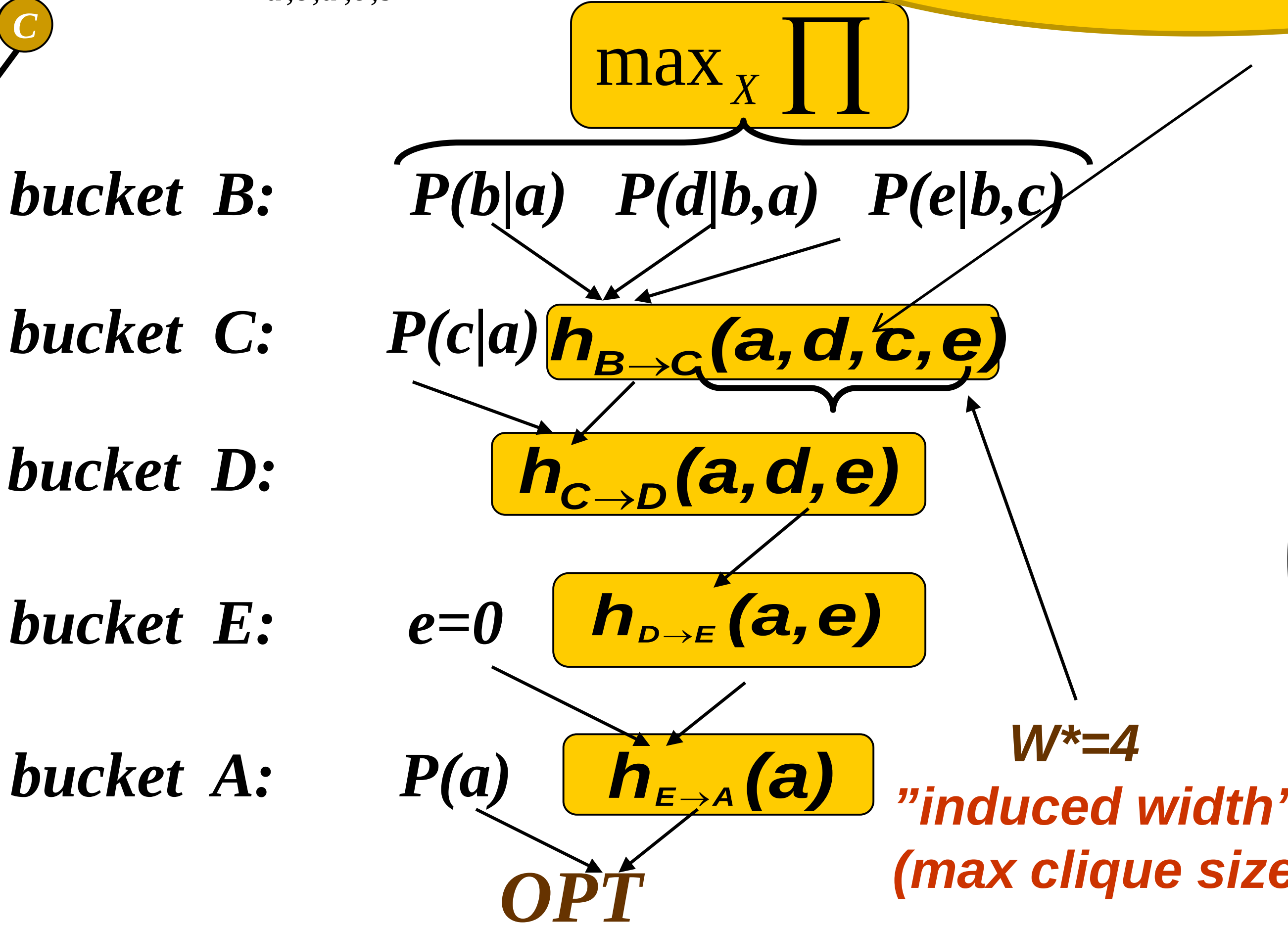
Query 2: Finding MAP by B

Algorithm BE-mpe (Dechter 1996, Berte

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



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



$\max_X \prod$

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

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

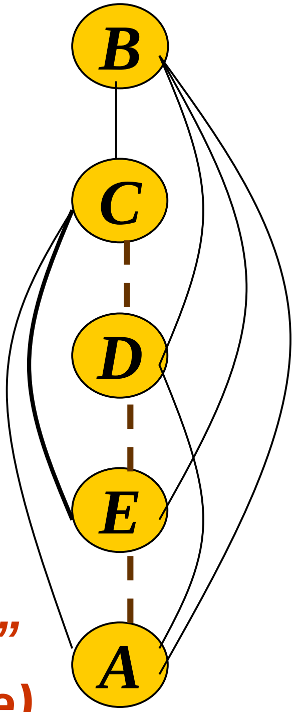
bucket D: $h_{C\to D}(a,d,e)$

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

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

OPT

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



Generating the MAP-tuple

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

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

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')

Complexity of Bucket Elimination

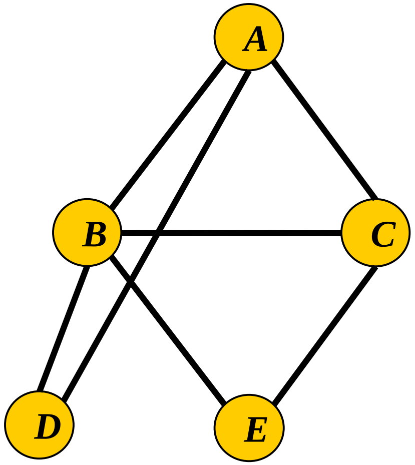
Bucket Elimination is time and space

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

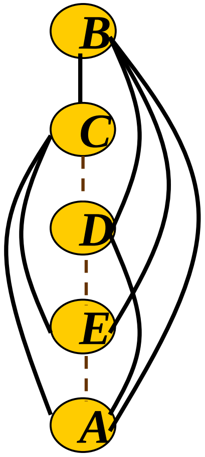
$w^*(d)$ – the induced width of graph along ordering d

r = number of functions

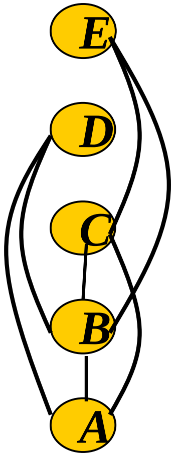
The effect of the ordering:



"Moral" graph



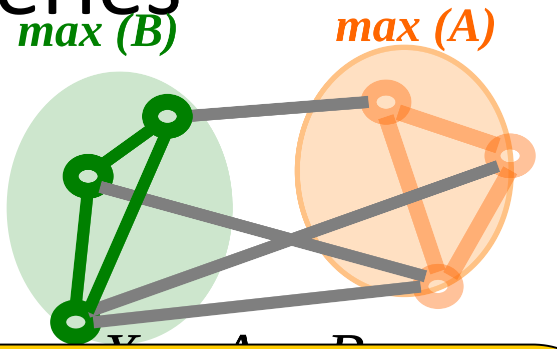
$w^*(d_1) = 4$



$w^*(d_2) = 2$

Finding the smallest induced width is hard!

Inference (by BE) Solves all Queries



- **MAP/MPE queries:**

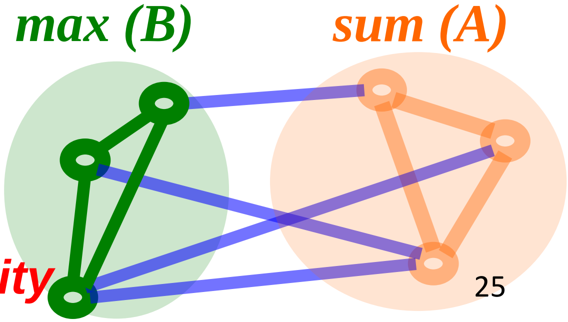
$$x^*_{AB} = \arg\max_{x_A, x_B} \prod_{\alpha} \varphi_\alpha$$

Inference requires memory!!!

Solve the problem quickly only if treewidth smaller than 20

Or, not at all.

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



Also Satisfiability and Max- Expected utility

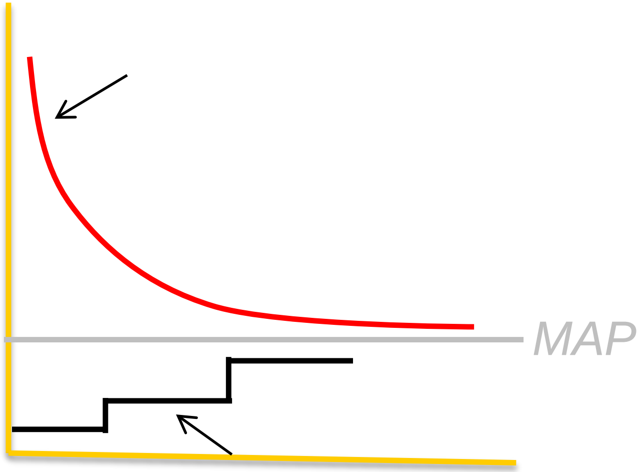
Outline

- Graphical models, Queries, Inference vs search
- Inference Algorithms: bucket-elimination
- AND/OR search spaces
- Bounded Inference: a) mini-bucket, b) cost-shifting
- Generating heuristics using mini-bucket elimination
- AND/OR Heuristic Search: Map and Marginal Map
- Conclusion

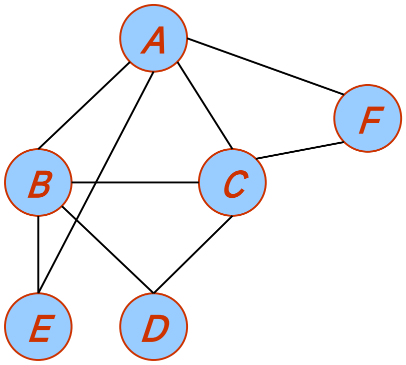
The AND/OR Search graph
Facilitates heuristic search

How to design a good Optimization solver (MAP)

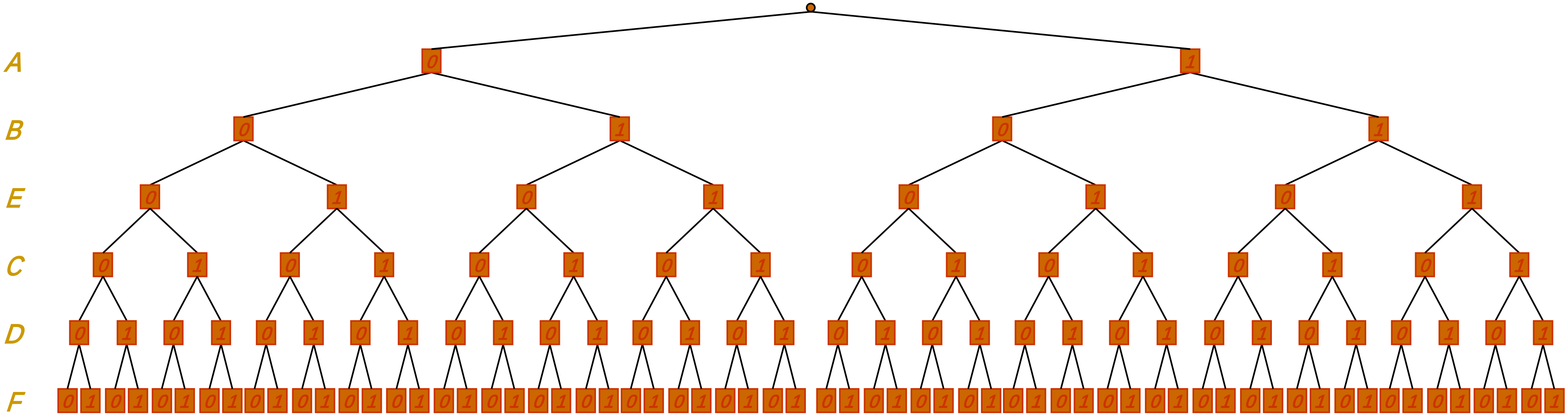
- Heuristic Search
- The core of a good search algorithm
  - □ A compact search space
  - □ A good heuristic evaluation function
  - □ A good traversal strategy
- Anytime search yields a good approximation.



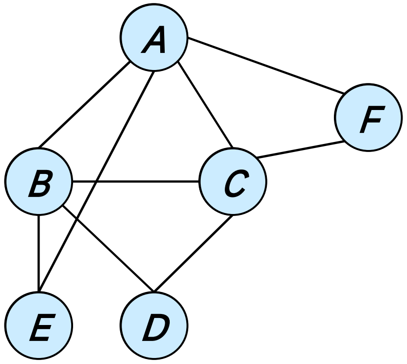
Classic OR Search Space



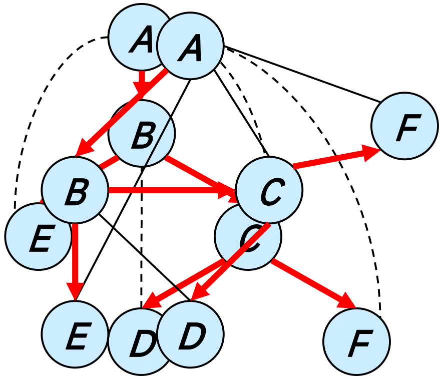
Ordering: A B E C D F



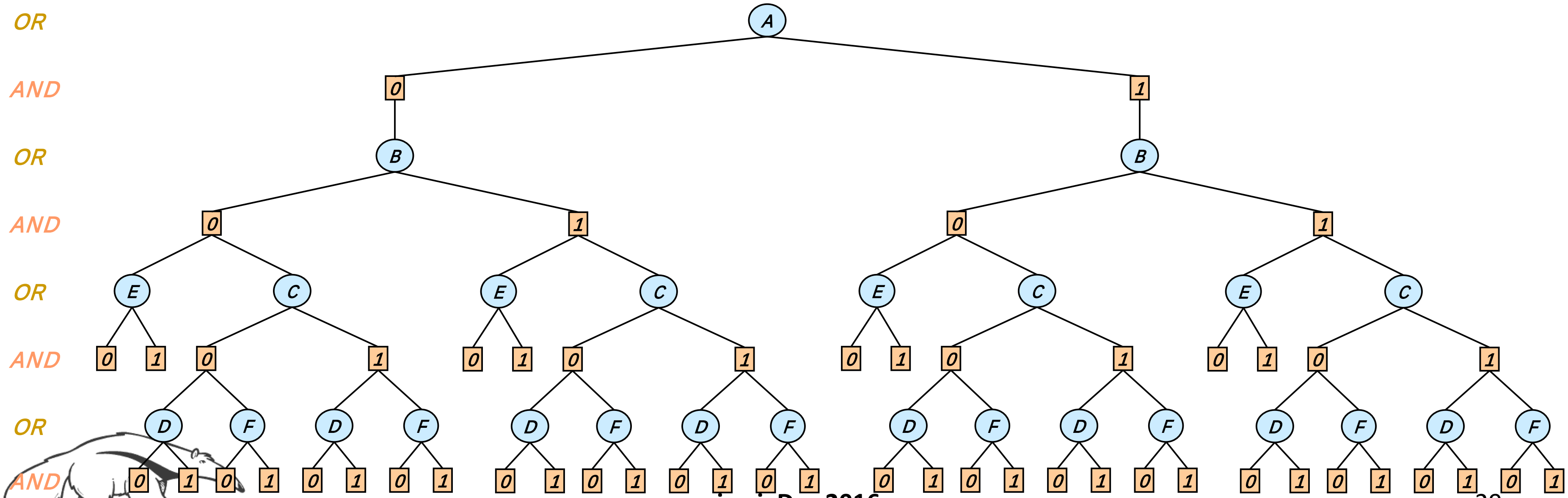
AND/OR Search Space



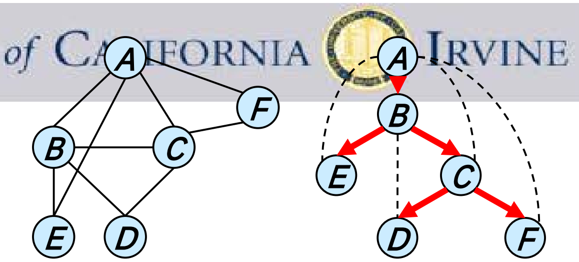
Primal graph

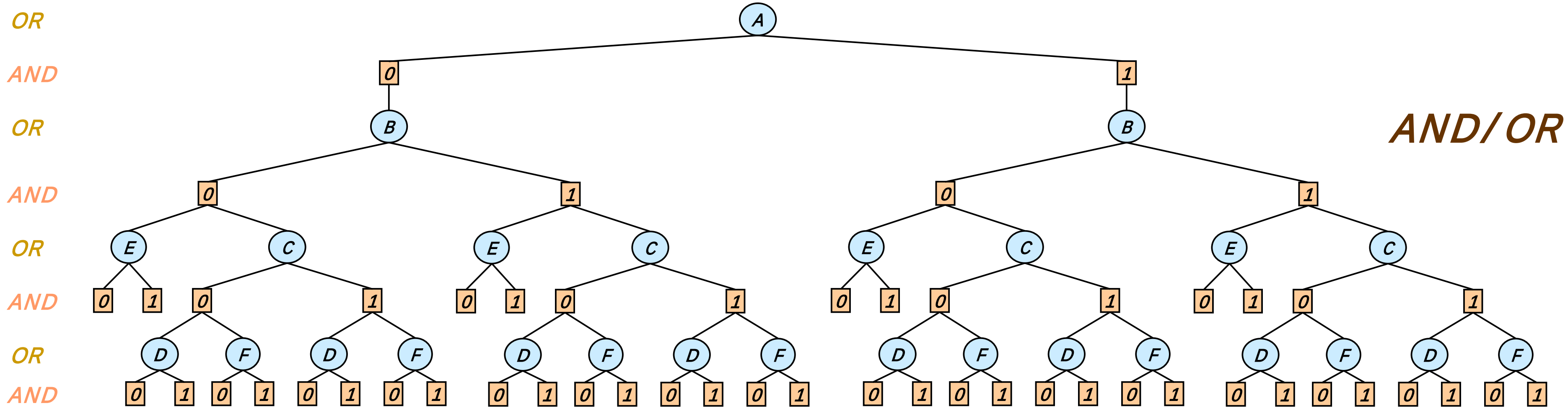


DFS tree

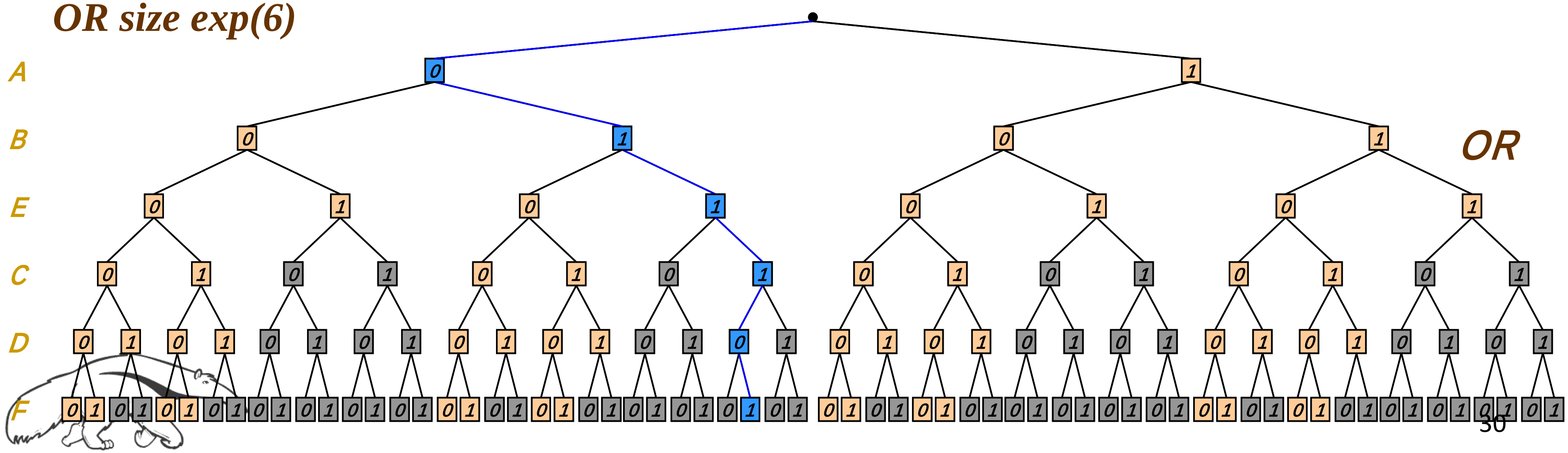


AND/OR vs. OR

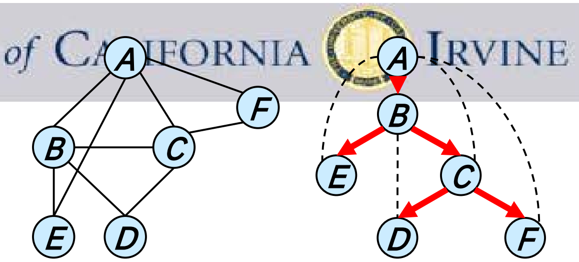


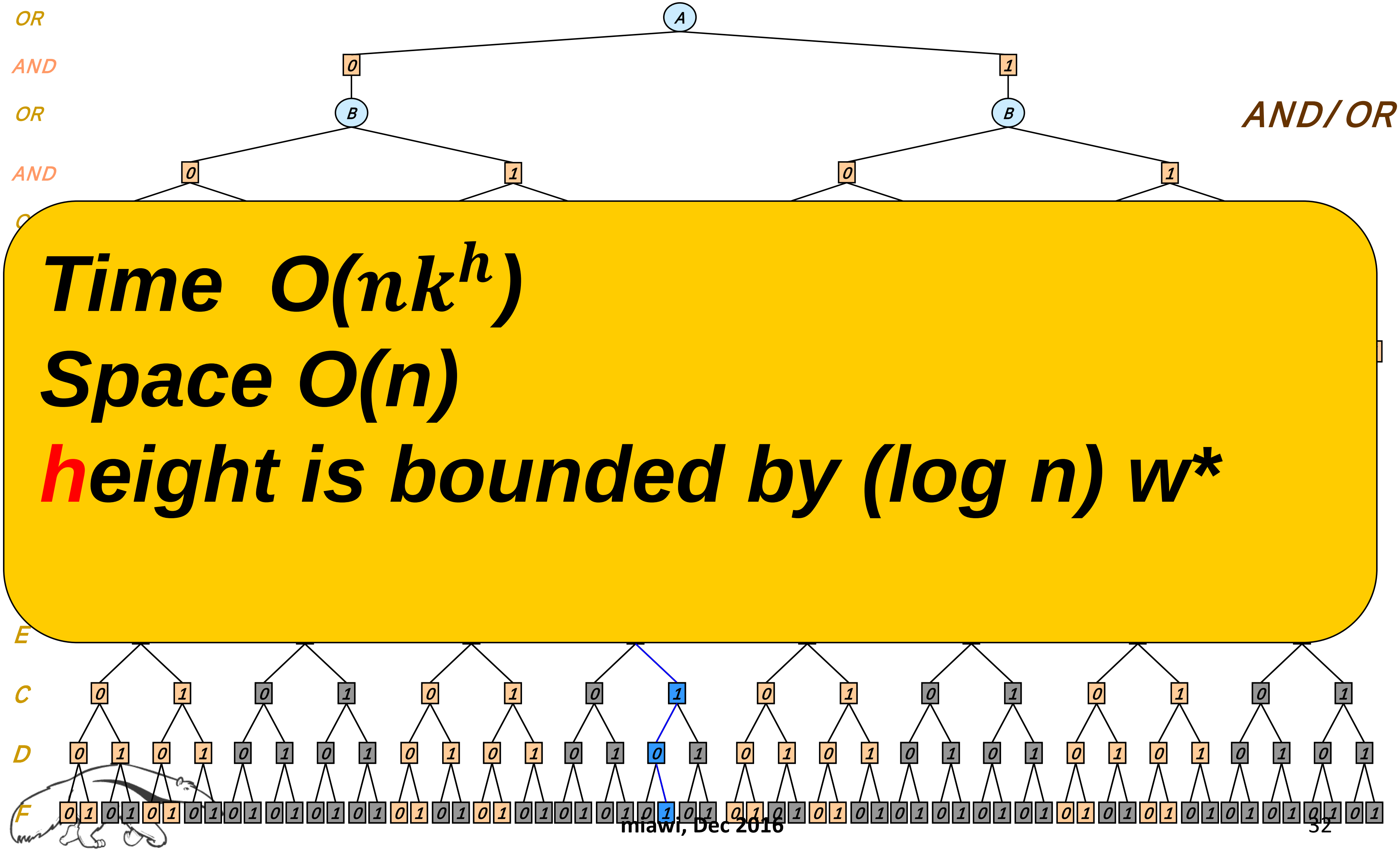


AND/OR size: exp(4),
OR size exp(6)



AND/OR vs. OR



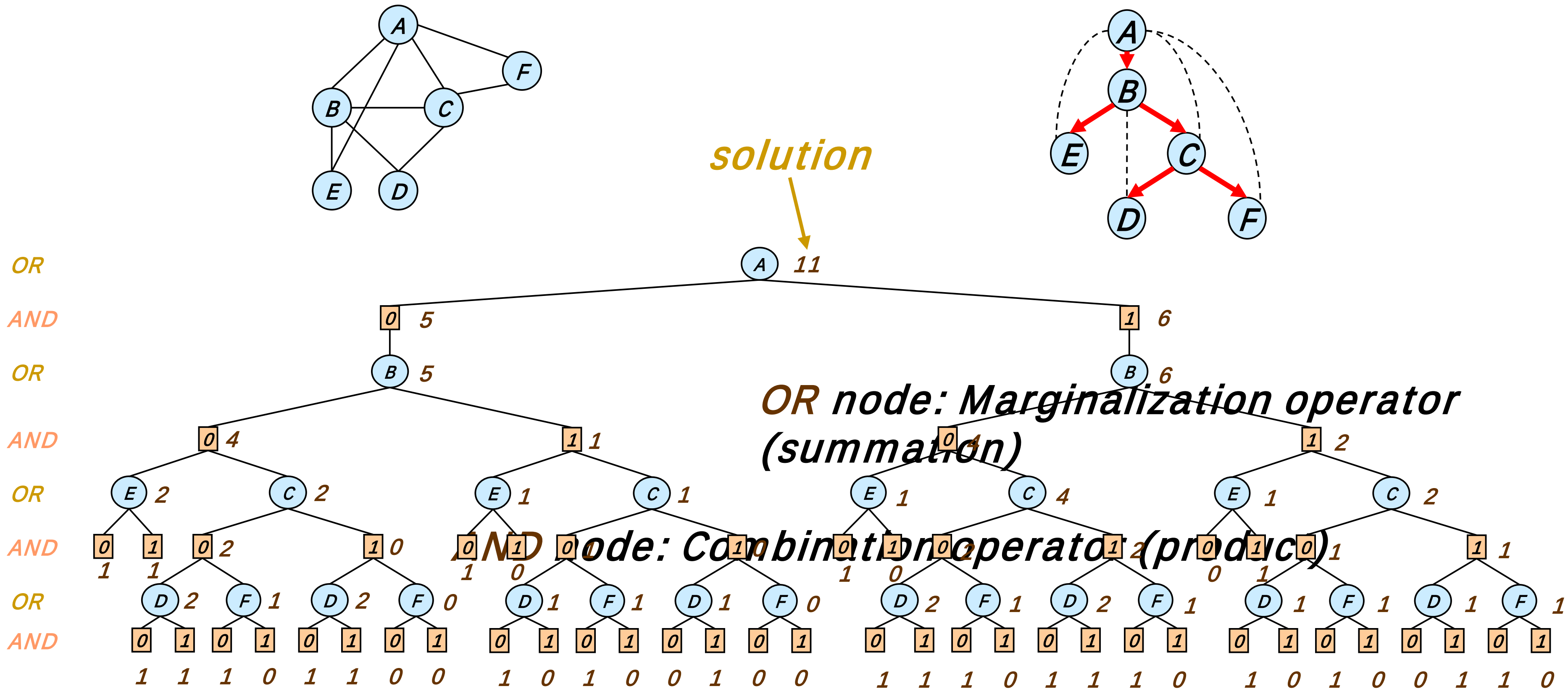


Time $O(nk^h)$

Space O(n)

height is bounded by (log n) w*

DFS algorithm (#CSP example)



OR node: Marginalization operator (summation)

AND node: Combination operator (product)

Value of node = number of solutions below it

AND/OR Tree DFS Algorithm (Belief Updating)

$P(E \mid A,B)$

A	B	E=0	E=1
0	0	.4	.6
0	1	.5	.5
1	0	.7	.3
1	1	.2	.8

Evidence: E=0

$P(B \mid A)$

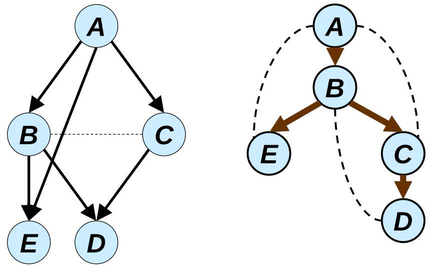
A	B=0	B=1
0	.4	.6
1	.1	.9

$P(C \mid A)$

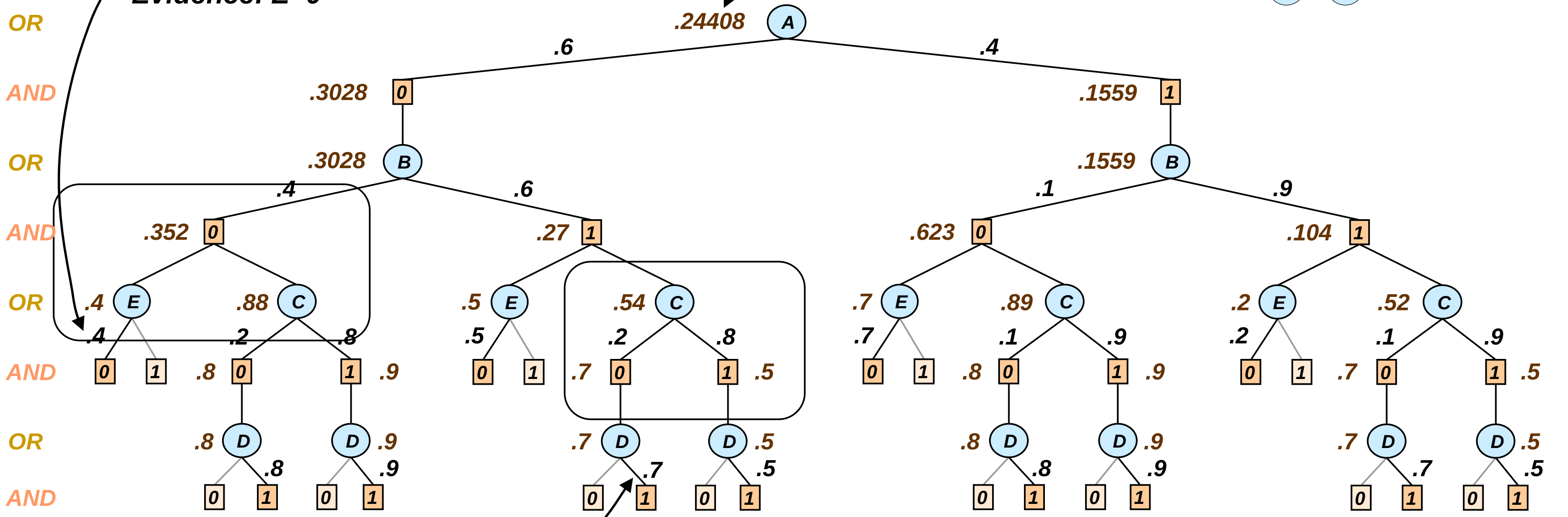
A	C=0	C=1
0	.2	.8
1	.7	.3

$P(A)$

A	P(A)
0	.6
1	.4



Result: P(D=1,E=0)



$P(D \mid B,C)$

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

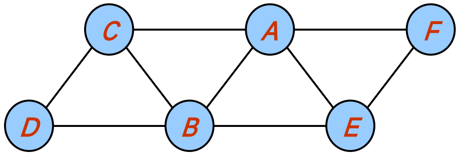
Evidence: D=1

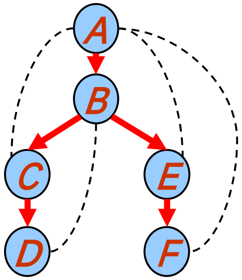
OR node: Marginalization by summation

AND node: product

Value of node = updated belief for sub-problem below

AND/OR Tree Search for Optimization





A	B	f_1
0	0	**2**
0	1	**0**
1	0	**1**
1	1	**4**

A	C	f_2
0	0	**3**
0	1	**0**
1	0	**0**
1	1	**1**

A	E	f_3
0	0	**0**
0	1	**3**
1	0	**2**
1	1	**0**

A	F	f_4
0	0	**2**
0	1	**0**
1	0	**0**
1	1	**2**

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

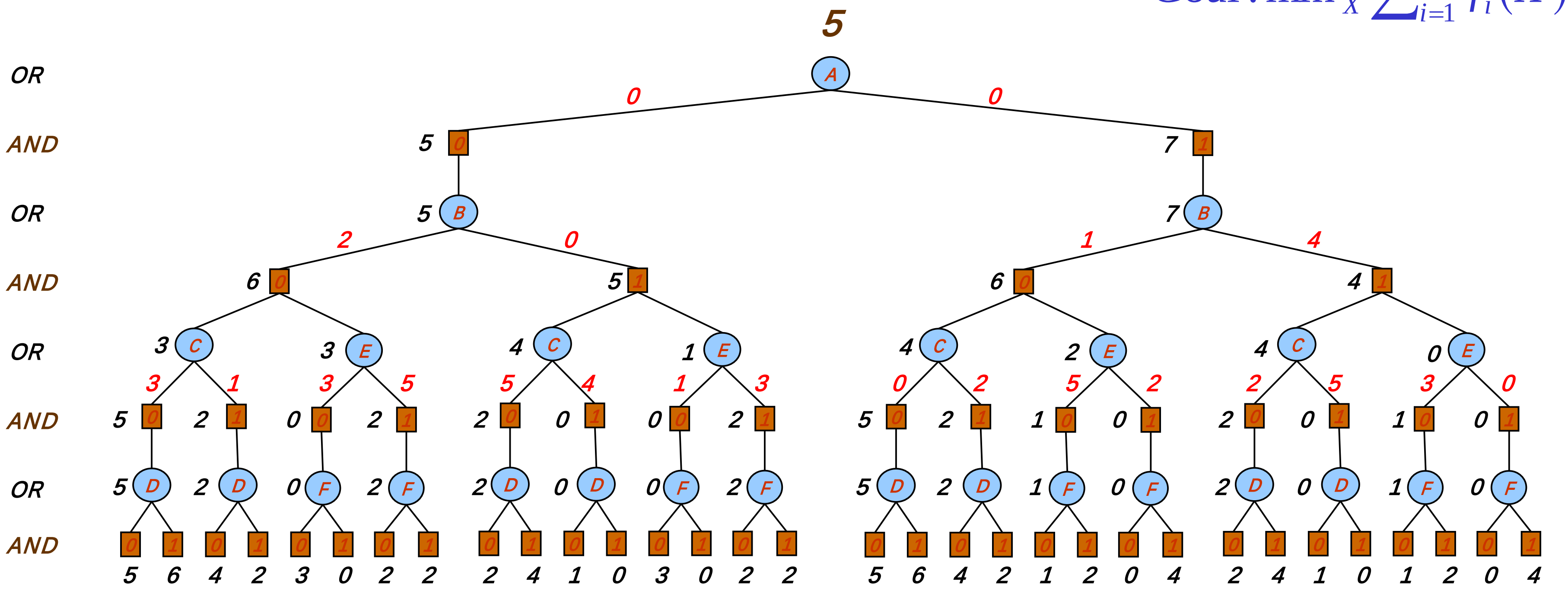
B	D	f_6
0	0	**4**
0	1	**2**
1	0	**1**
1	1	**0**

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

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

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

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

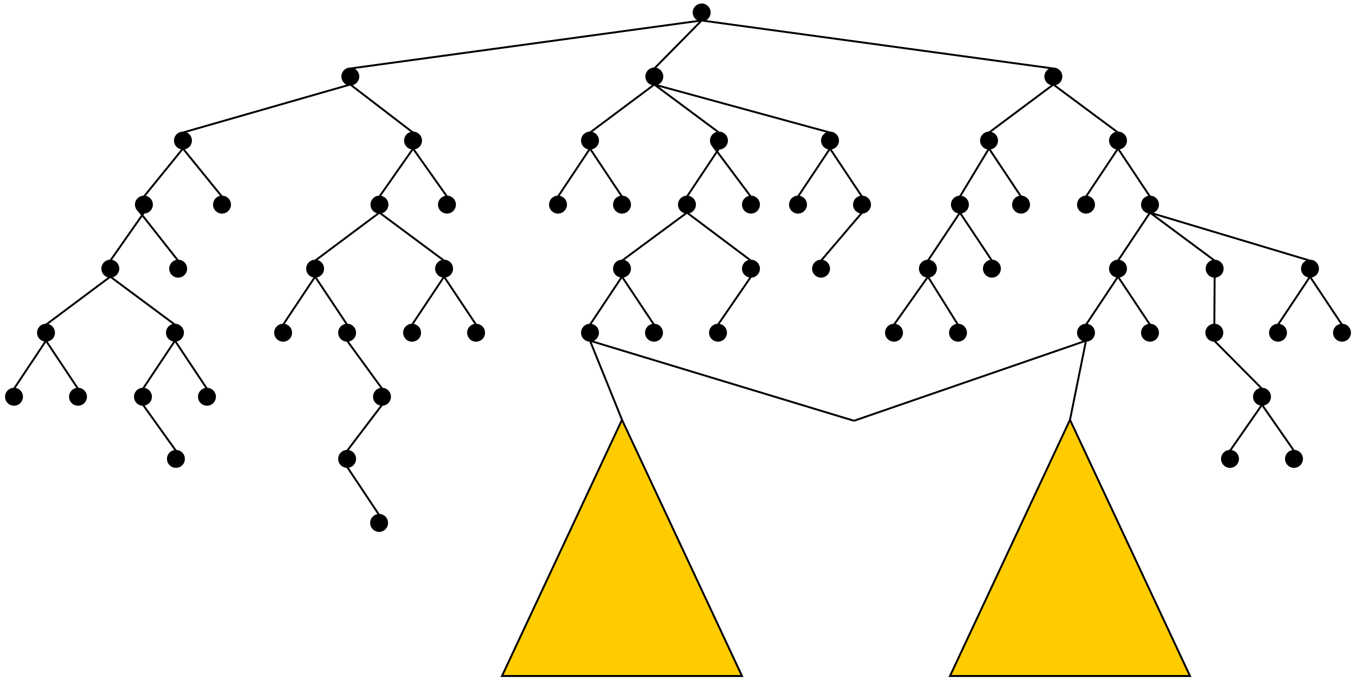


AND node = Combination operator (summation)

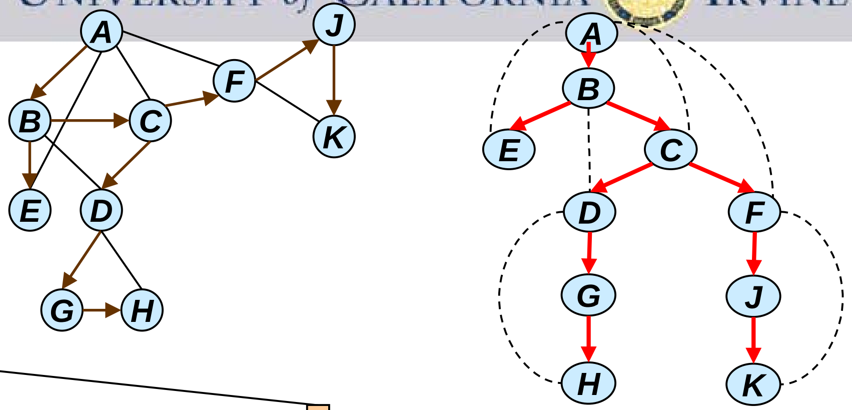
OR node = Marginalization operator (minimization)

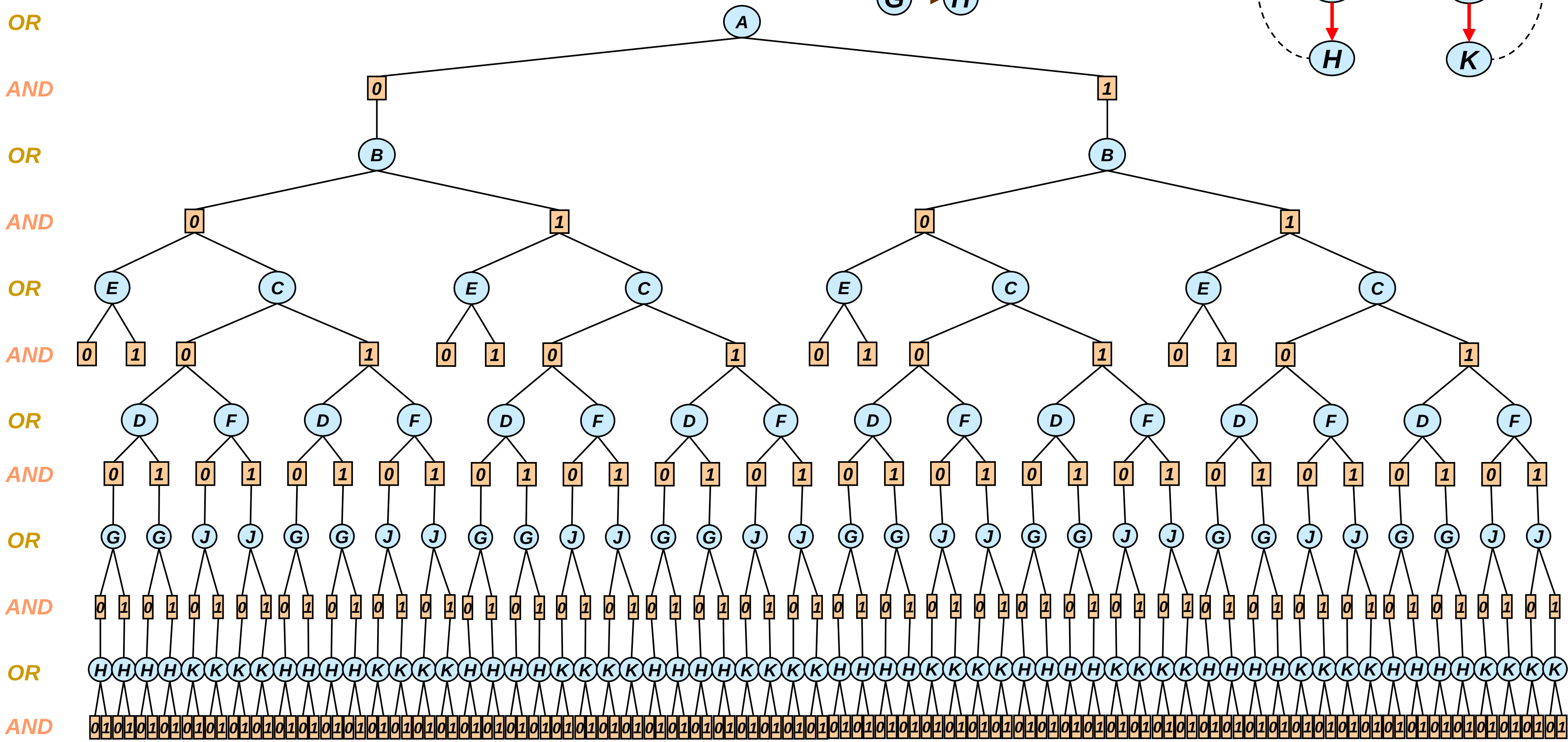
From Search Trees to Search Graphs

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

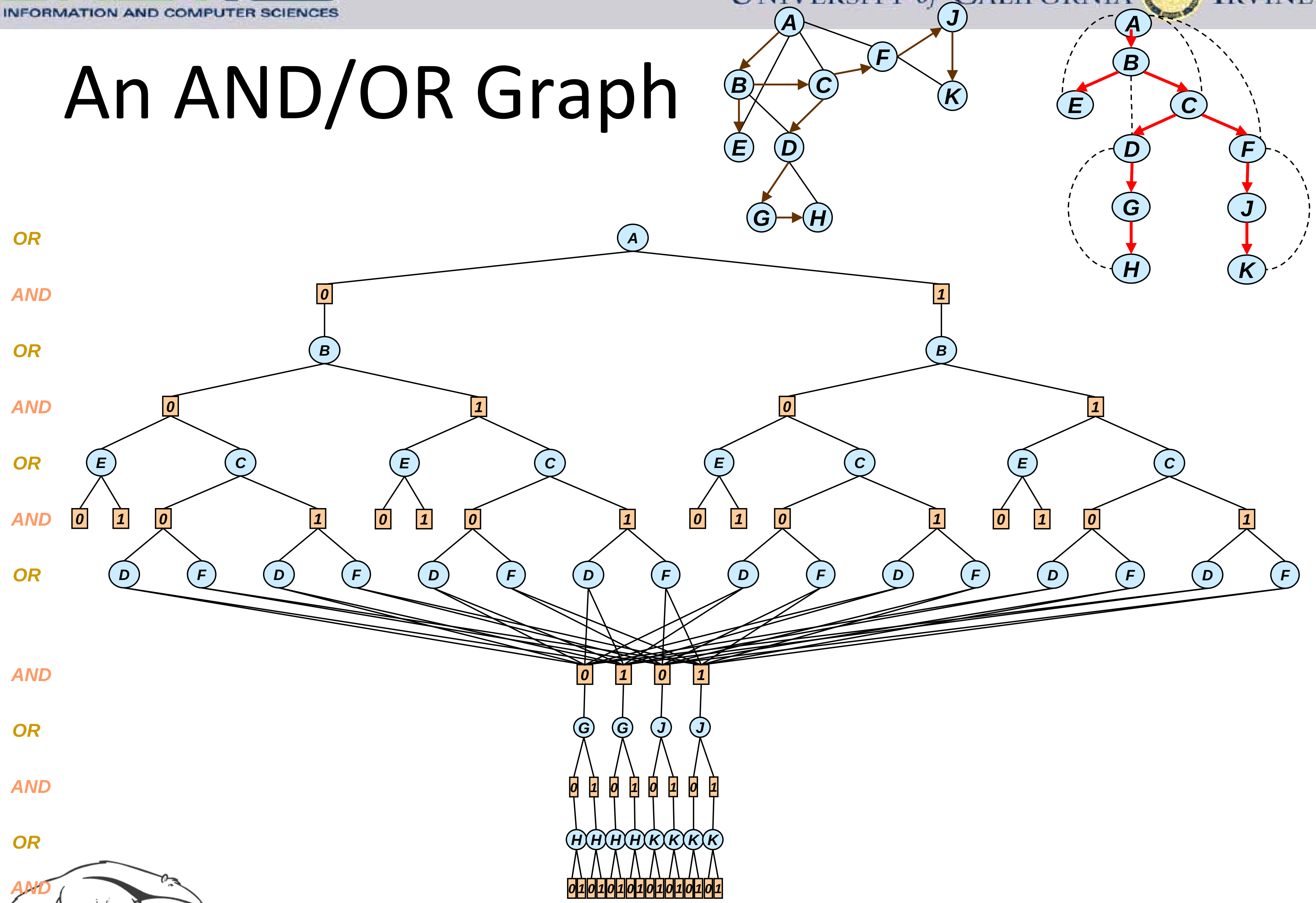


From AND/OR Tree





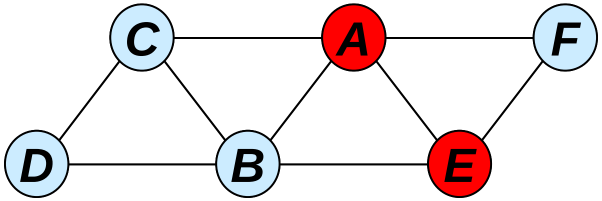
An AND/OR Graph

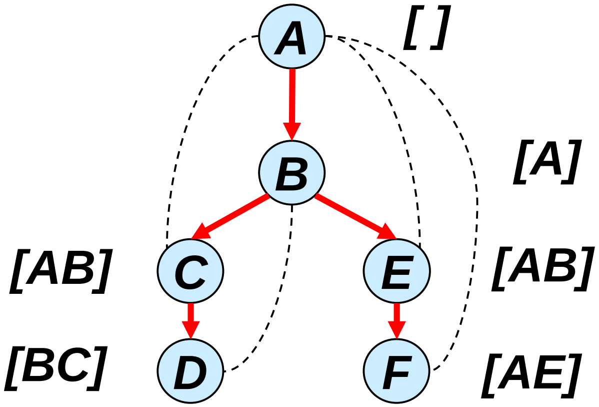


Merging Based on Context

- One way of recognizing nodes that can be merged (based on graph structure)

    context(X) = ancestors of X in the pseudo tree that are connected to X, or to descendants of X





pseudo tree

Answering Queries: Sum-Product (Belief Updating)

$P(E\,|\,A,B)$

A	B	E=0	E=1
0	0	.4	.6
0	1	.5	.5
1	0	.7	.3
1	1	.2	.8

Evidence: E=0

$P(B\,|\,A)$

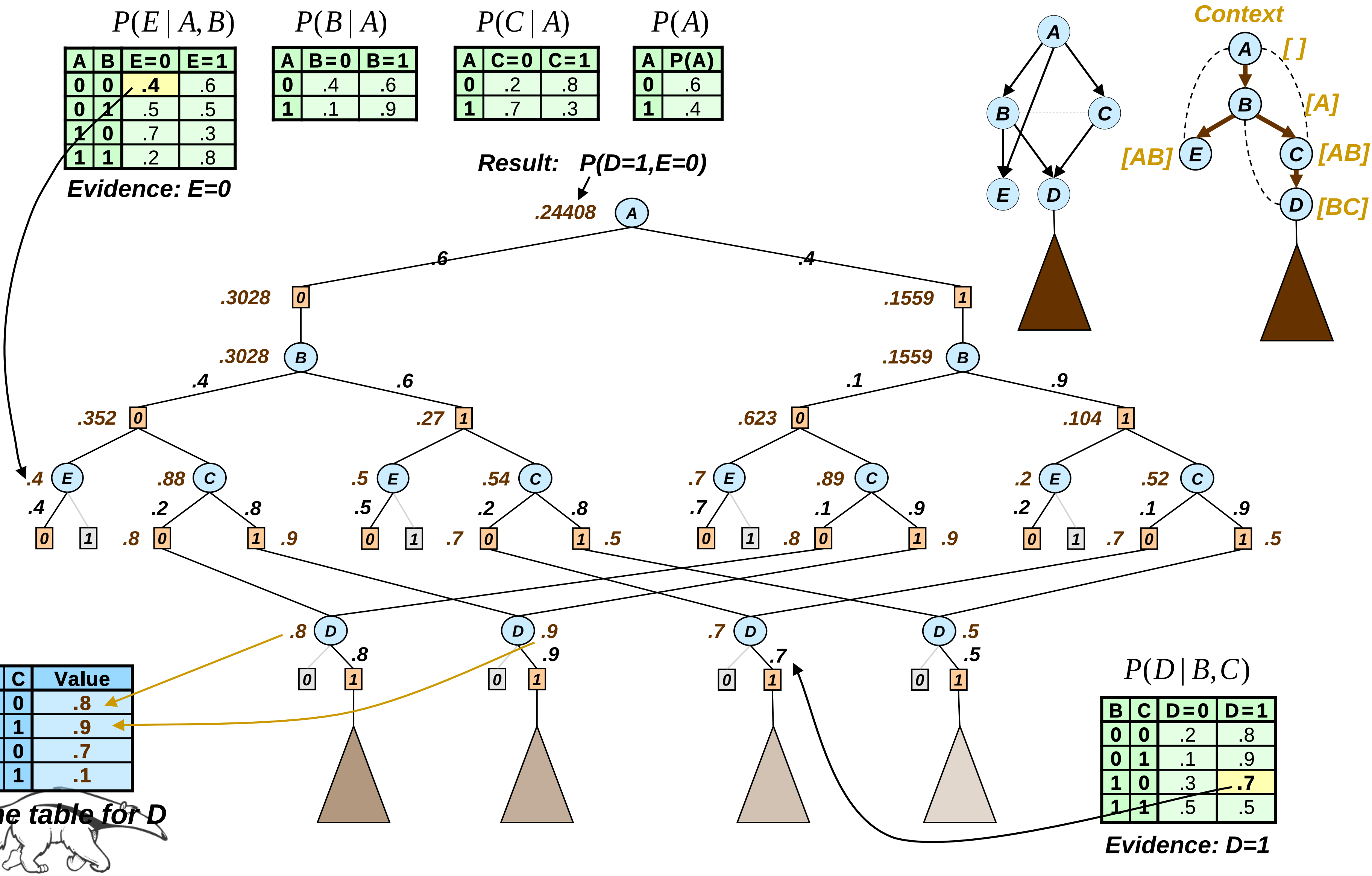
A	B=0	B=1
0	.4	.6
1	.1	.9

$P(C\,|\,A)$

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

$P(A)$

A	P(A)
0	.6
1	.4



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

Cache table for D

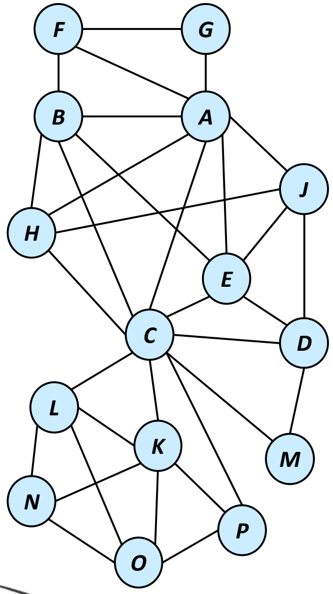
$P(D\,|\,B,C)$

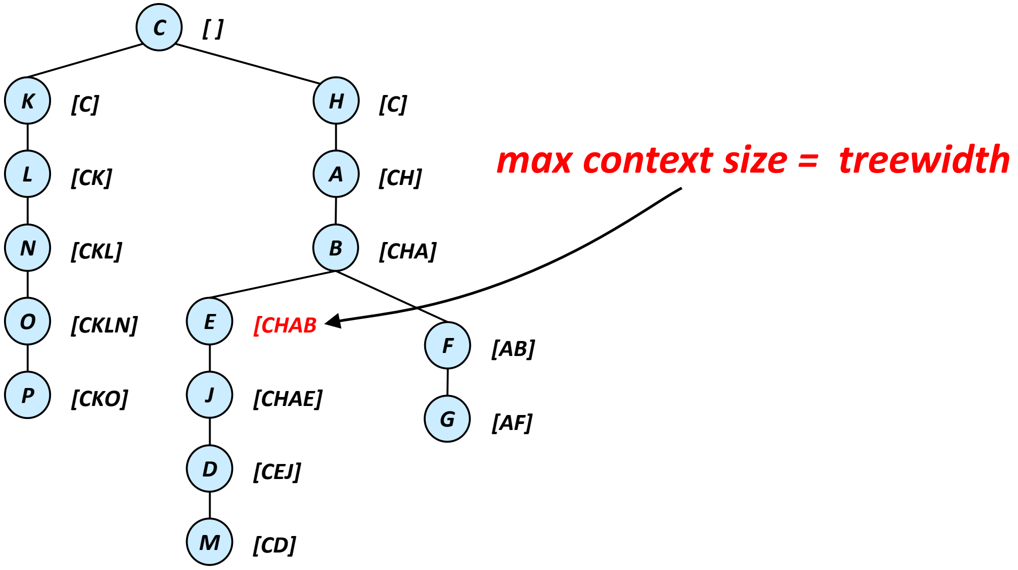
B	C	D=0	D=1
0	0	.2	.8
0	1	.1	.9
1	0	.3	.7
1	1	.5	.5

Evidence: D=1

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.

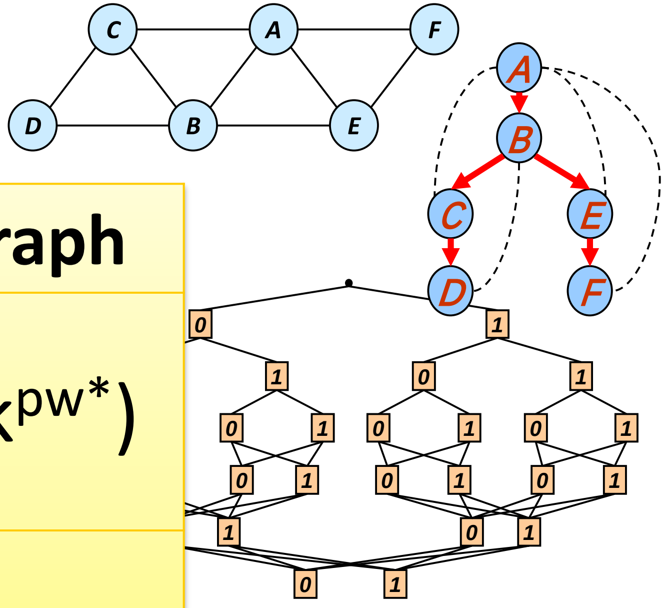




(C K H A B E J L N O D P M F G)

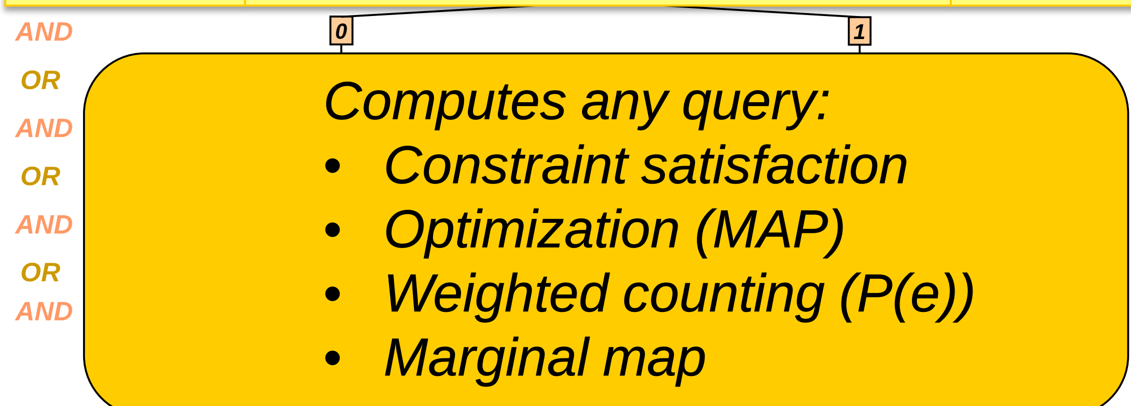
All Four Search Spaces

	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

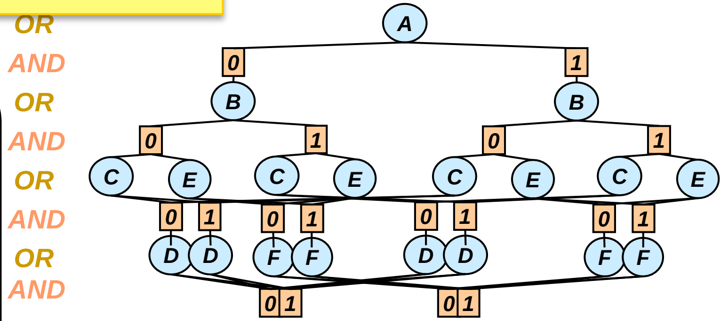
28 nodes



54 AND nodes

Computes any query:

- *Constraint satisfaction*
- *Optimization (MAP)*
- *Weighted counting (P(e))*
- *Marginal map*



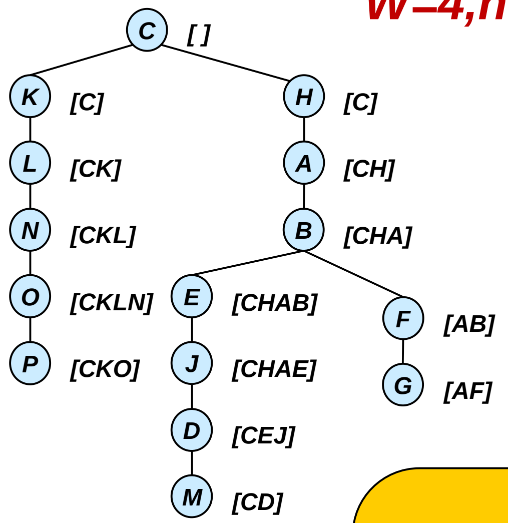
Context minimal AND/OR search graph

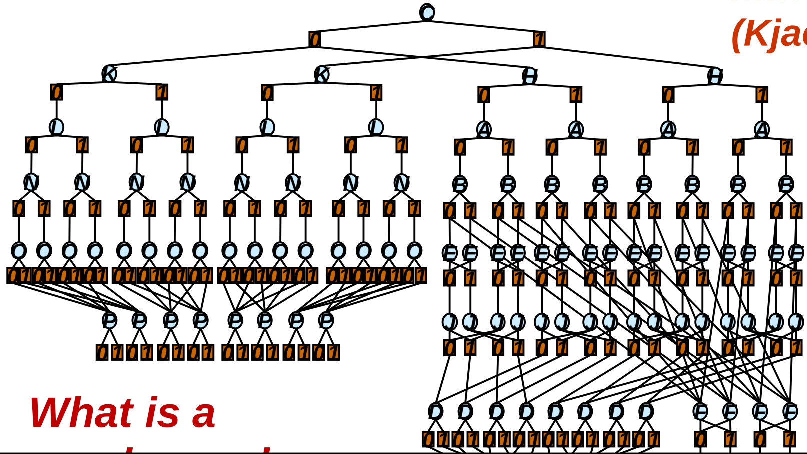
18 AND nodes

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

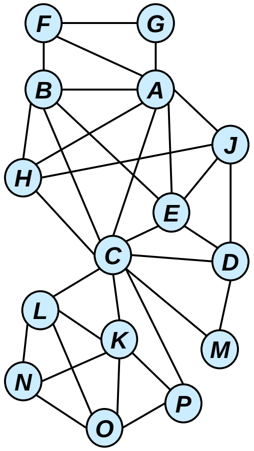
The impact of the pseudo-tree

W=4,h=8





Min-Fill (Kjaerulff90)



What is a

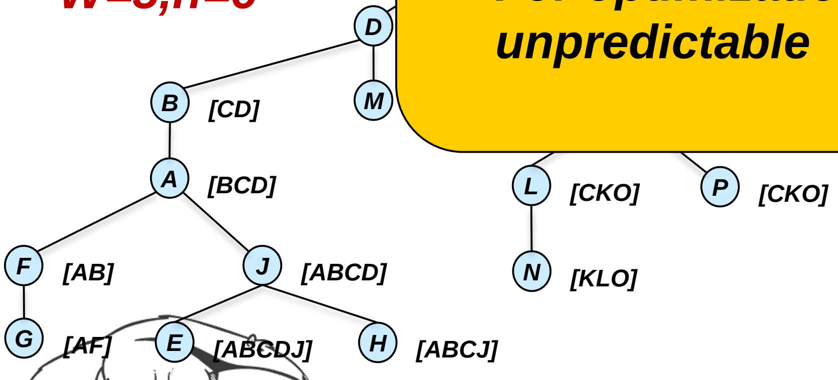
raph

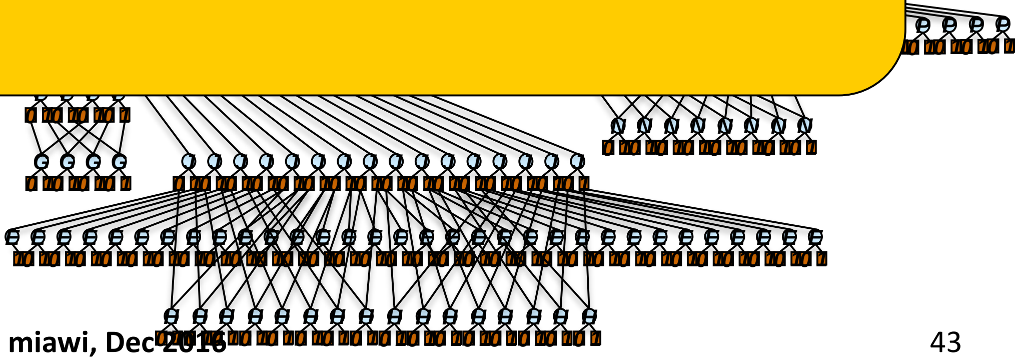
ing

(C K H A B E J L N O

- *Choose pseudo-tree with a minimal search graph*
- *But determinism is unpredictable*
- *For optimization , pruning by BnB is even more unpredictable*

W=5,h=6





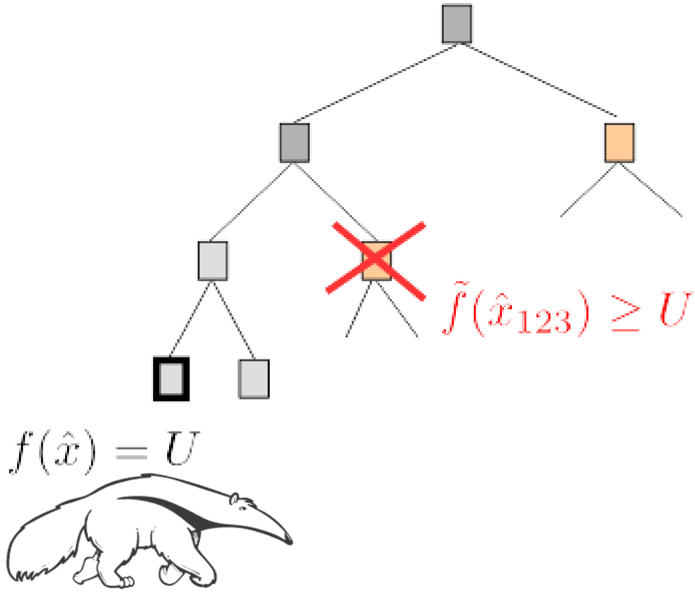
(C D K B A O M L N P J H E F G)

Basic Heuristic Search Schemes

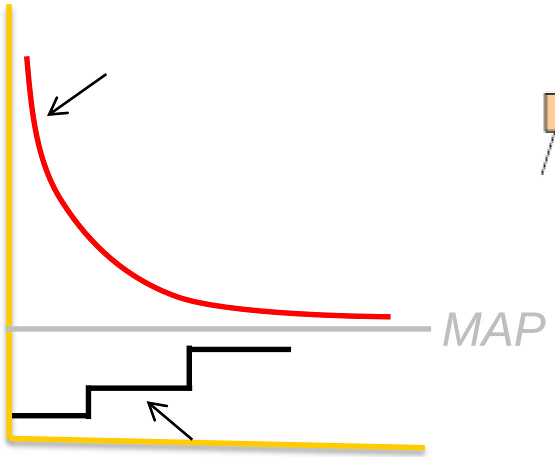
Heuristic function $\tilde{f}(\hat{x}_p)$ computes a lower bound on the best extension o partial configuration $\hat{x}_p$ and can be used to guide heuristic search.
We focus on:

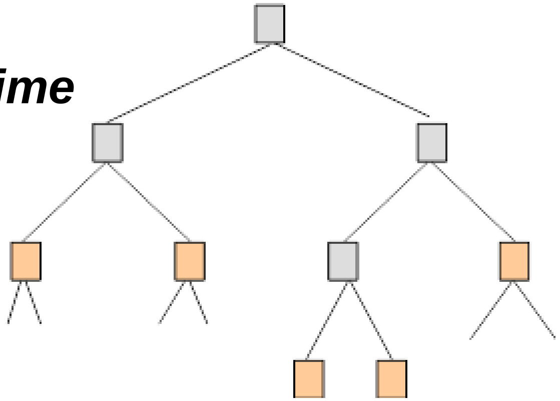
1. Branch-and-Bound
Use heuristic function $\tilde{f}(\hat{x}_p)$ to prune the depth-first search tree
Linear space

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



BnB is upper-bound anytime





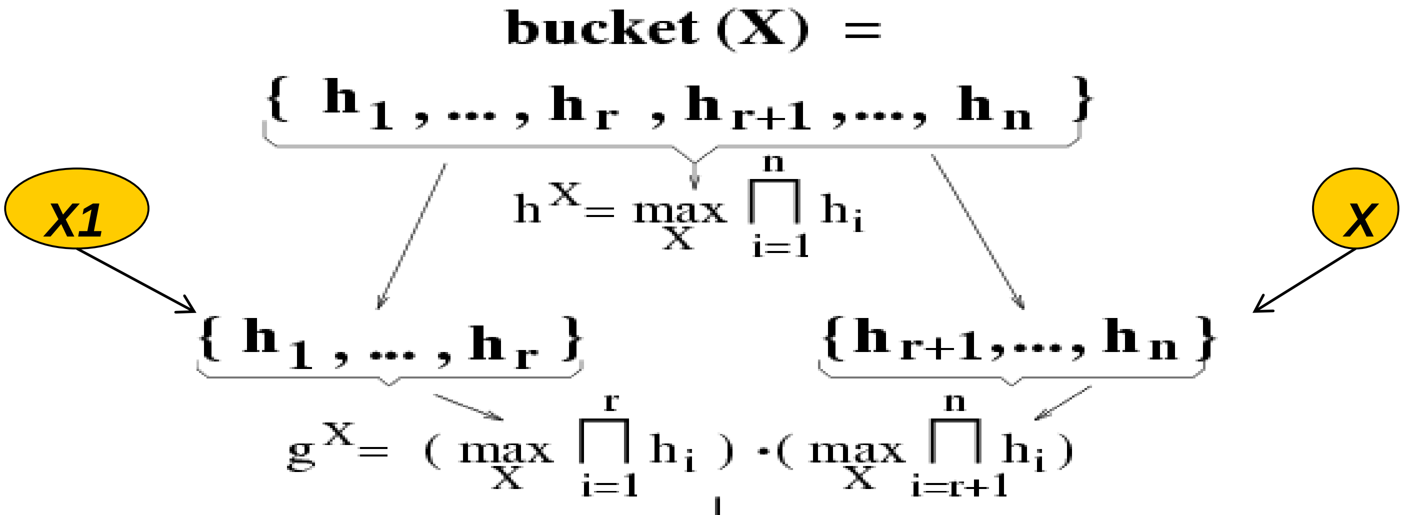
Outline

- Graphical models, Queries, Inference vs search
- Inference Algorithms: bucket-elimination
- AND/OR search spaces and AND/OR BnB
- Bounded Inference: a) mini-bucket, b) cost-shifting
- Evaluation, Software, Map and Marginal Map
- Conclusion

Mini-bucket Approximation

(Dechter and Rish, 1997, 2003)

Split a bucket into mini-buckets =>bound complexity



$$\mathbf{bucket}\ (\mathbf{X})\ =$$

$$\underbrace{\{\ \mathbf{h}_1, \ldots, \mathbf{h}_r, \mathbf{h}_{r+1}, \ldots, \mathbf{h}_n\ \}}$$

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

$$\underbrace{\{\ \mathbf{h}_1, \ldots, \mathbf{h}_r\ \}} \qquad \underbrace{\{\mathbf{h}_{r+1}, \ldots, \mathbf{h}_n\}}$$

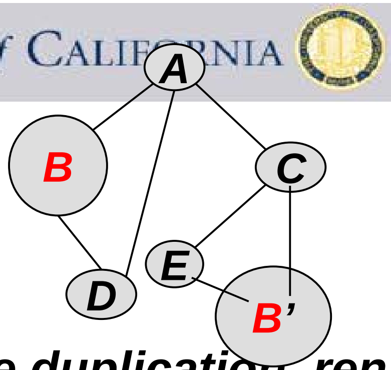
$$g^X = \left(\max_X \prod_{i=1}^{r} h_i\right) \cdot \left(\max_X \prod_{i=r+1}^{n} h_i\right)$$

$$\downarrow$$

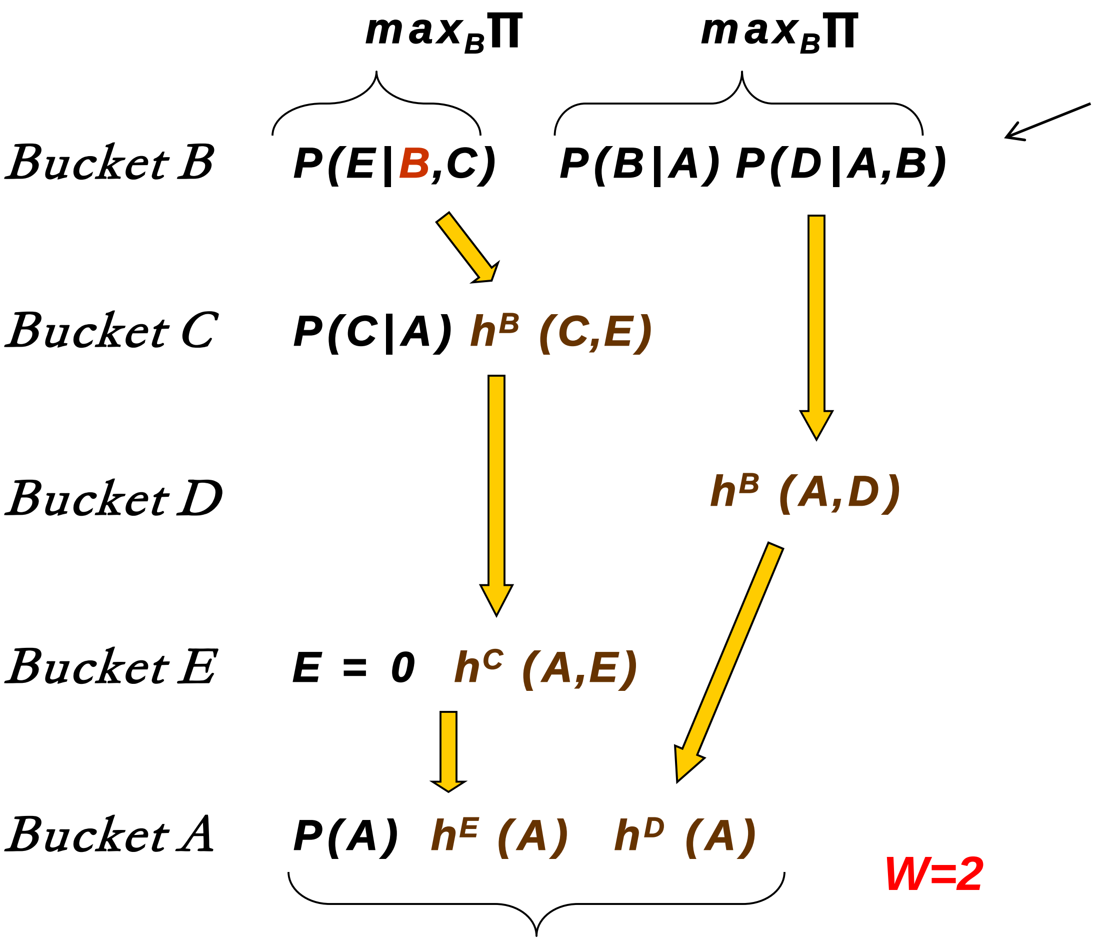
$$\mathbf{h}^{\mathbf{x}} \leq \mathbf{g}^{\mathbf{x}}$$

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

Mini-Bucket Elimination



Node duplication, renaming



$\max_B \prod$ $\max_B \prod$

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

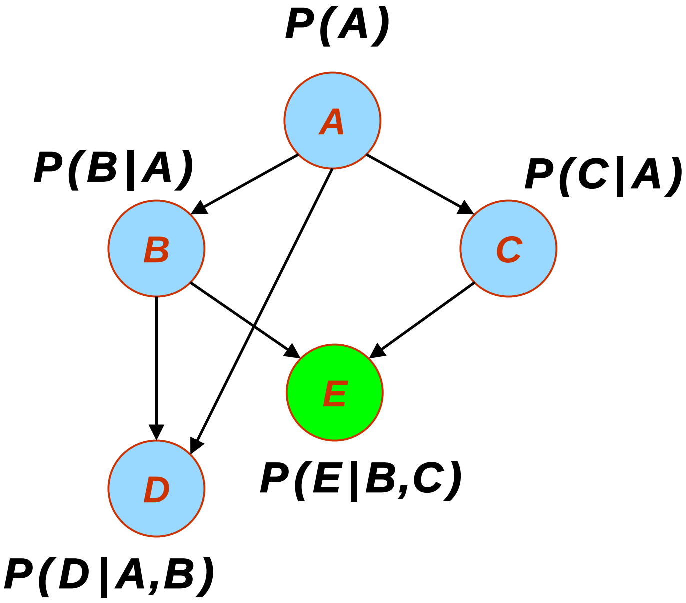
Bucket C $P(C|A)\ h^B(C,E)$

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

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

Bucket A $P(A)\ \ h^E(A)\ \ h^D(A)$

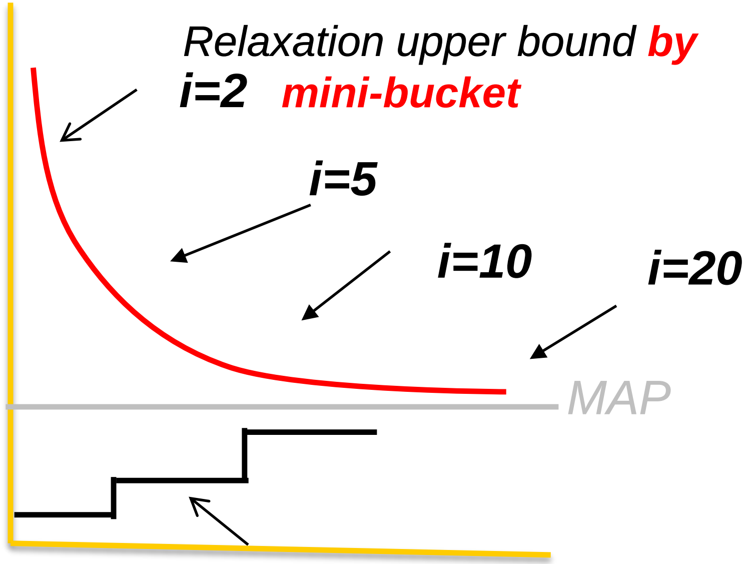
W=2



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

Properties of Mini-Bucket Eliminaton

- Bounding from above and below



Consistent solutions (greedy search)

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

Outline

- Graphical models, Queries, Algorithms
- Inference Algorithms: bucket-elimination
- Bounded Inference: a) mini-bucket, b) cost-shifting
- AND/OR search spaces and AND/OR BnB
- Evaluation, Software
- Conclusions

Cost-Shifting

(Reparameterization)

$+\lambda(B)$

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

+

$-\lambda(B)$

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

B	λ(B)
b	3
g	-1

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

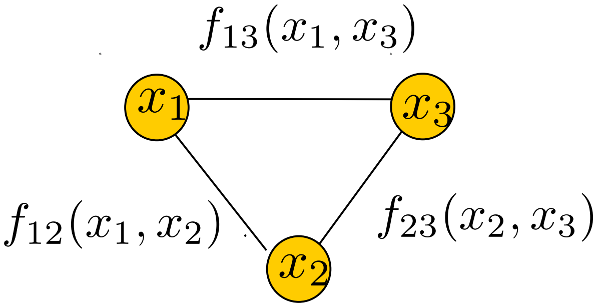
= 0 + 6

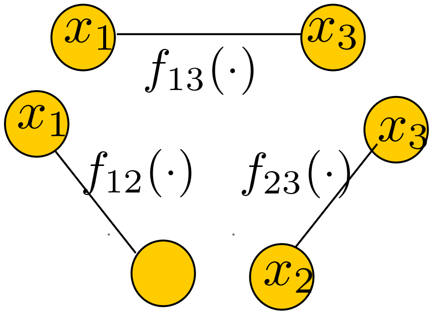
Modify the individual functions

- but -

keep the sum or product of functions unchanged

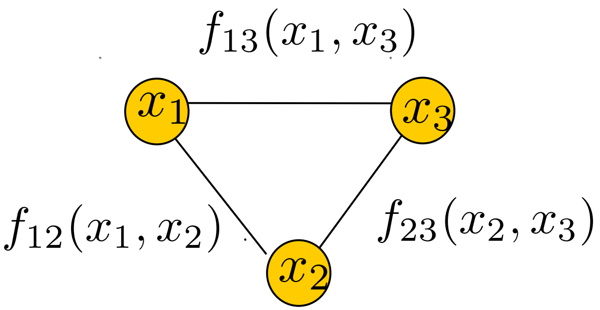
Dual Decomposition

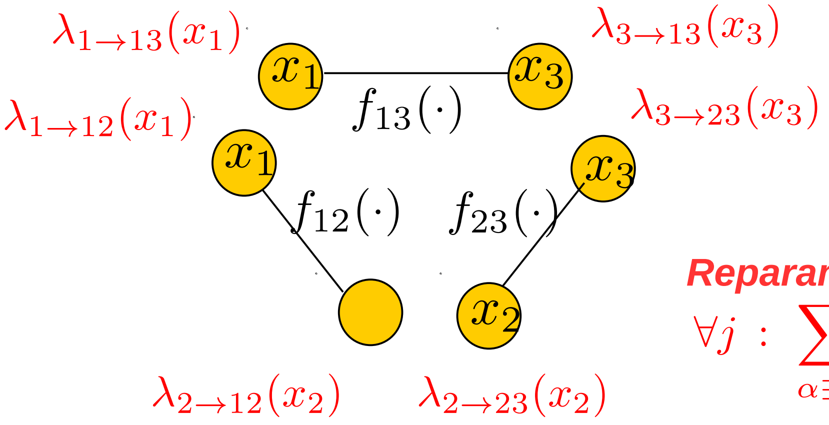




$$F^* = \min_x \sum_\alpha f_\alpha(x) \qquad \geq \qquad \sum_\alpha \min_x \; f_\alpha(x)$$

Dual Decomposition





Reparameterization:

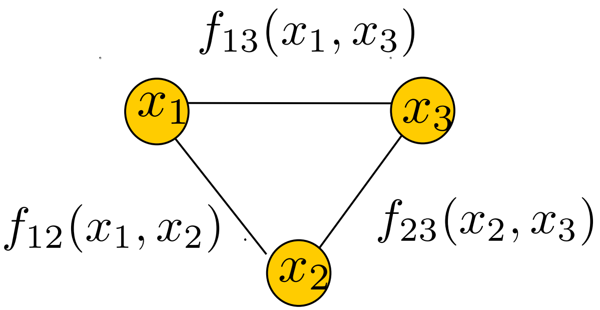
$$\forall j \; : \; \sum_{\alpha \ni j} \lambda_{j\to\alpha}(x_j) = 0$$

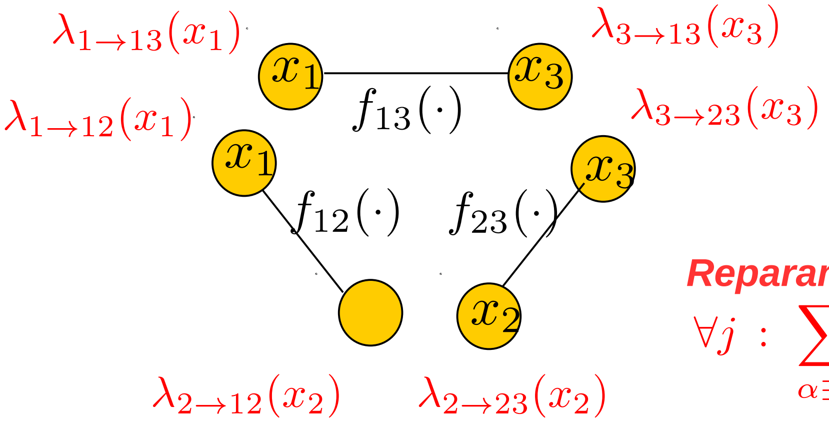
$$F^* = \min_x \sum_\alpha f_\alpha(x) \qquad \geq \max_{\lambda_{i\to\alpha}} \sum_\alpha \min_x \left[f_\alpha(x) + \sum_{i\in\alpha} \lambda_{i\to\alpha}(x_i) \right]$$

- *Bound solution using decomposed optimization*
- *Solve independently: optimistic bound*
- *Tighten the bound by reparameterization*
  - *Enforce lost equality constraints via Lagrange multipliers*

(Convex dual: linear programming relaxation)

Dual Decomposition





Reparameterization:

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

$$F^* = \min_x \sum_\alpha f_\alpha(x) \qquad \geq \max_{\lambda_{i\to\alpha}} \sum_\alpha \min_x \Big[f_\alpha(x) + \sum_{i\in\alpha} \lambda_{i\to\alpha}(x_i) \Big]$$

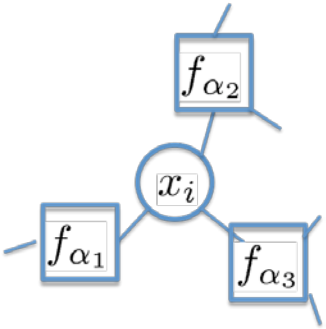
Many names for the same class of bounds:

- Dual decomposition [Komodakis et al. 2007]
- TRW, MPLP [Wainwright et al. 2005, Globerson & Jaakkola 2007]
- Soft arc consistency [Cooper & Schiex 2004]
- Max-sum diffusion [Warner 2007]

(Convex dual: linear programming relaxation)

Various Update Schemes

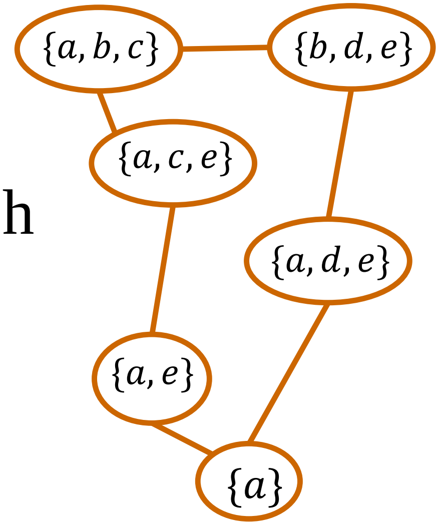
- Can use any decomposition updates
  - (message passing, subgradient, augmented, etc.)



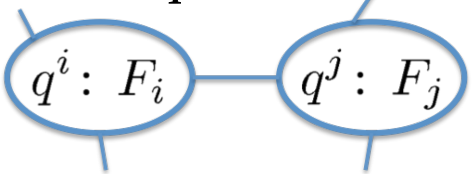
- FGLP: Update the original factors

$$\forall \alpha,\ \gamma_\alpha(x_i) = \max_{x_\alpha \setminus x_i} f_\alpha$$

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



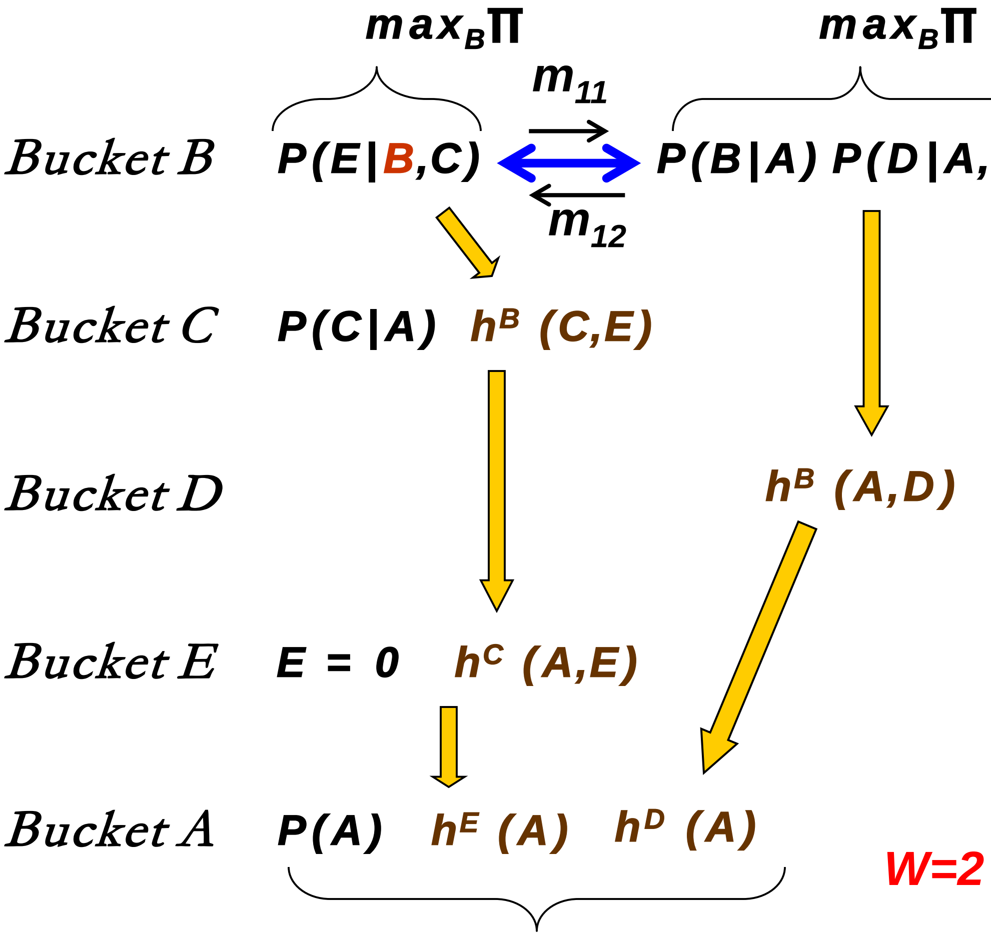
- JGLP: Update clique function of the join graph



- MBE-MM Update within each bucket only

MBE-MM: MBE with moment matching

m_{11},m_{12}- moment-matching messages



$\max_B \prod$ $\max_B \prod$

m_{11} m_{12}

Bucket B $P(E|B,C)$ $P(B|A)$ $P(D|A,B)$

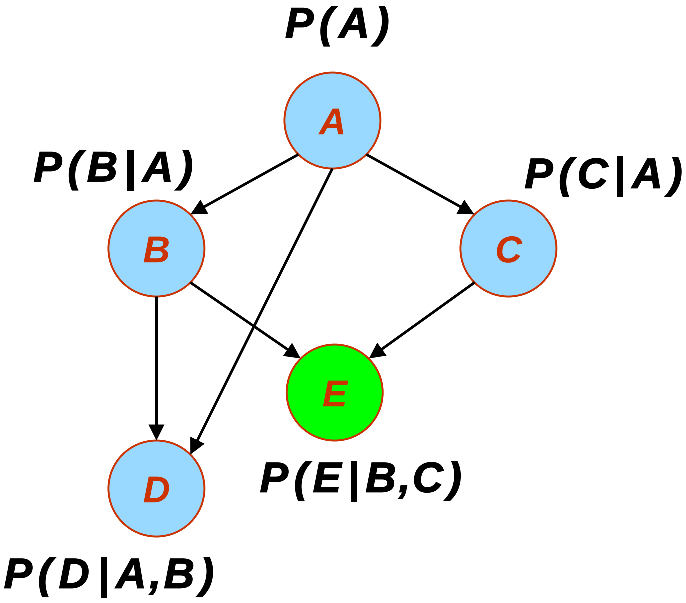
Bucket C $P(C|A)$ $h^B(C,E)$

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

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

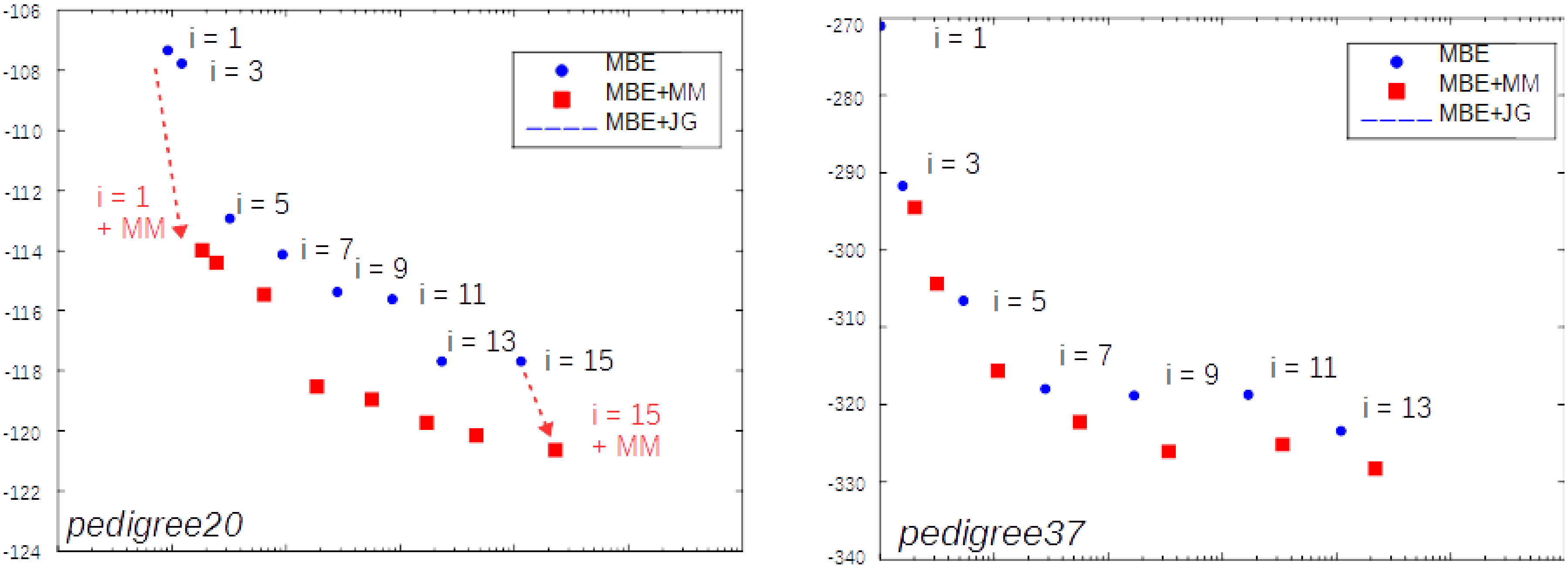
Bucket A $P(A)$ $h^E(A)$ $h^D(A)$

W=2



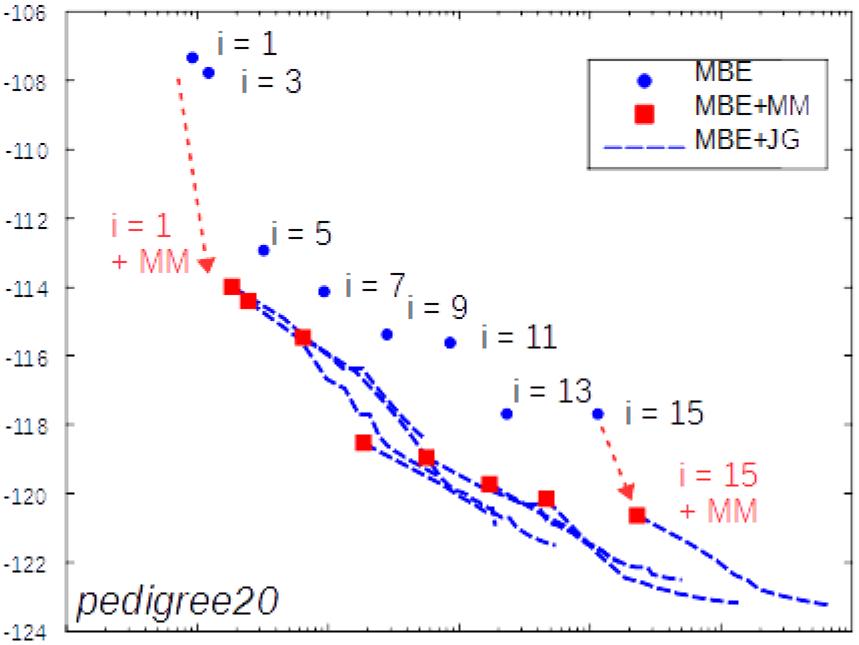
MPE* is an upper bound on MPE --U
Generating a solution yields a lower bound--L

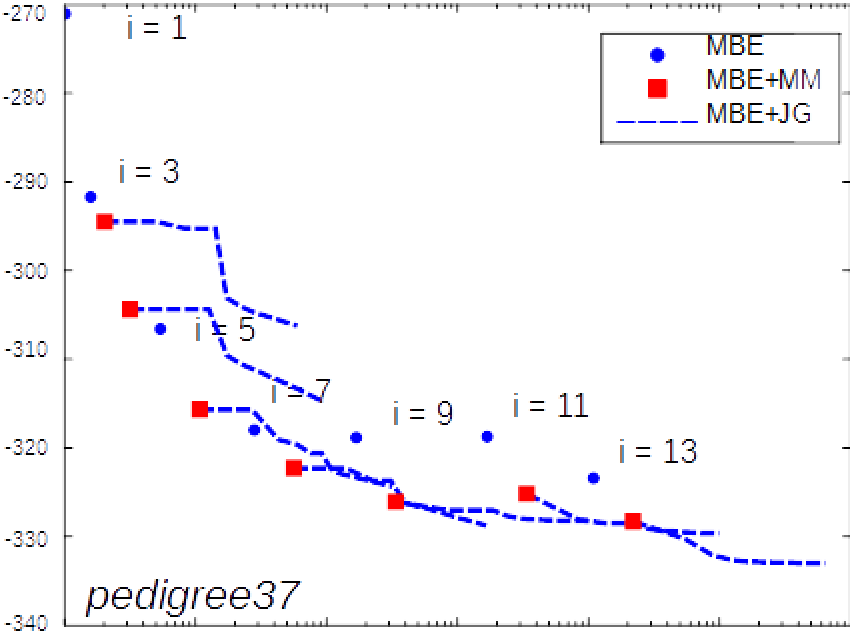
Anytime Approximation



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

Anytime Approximation





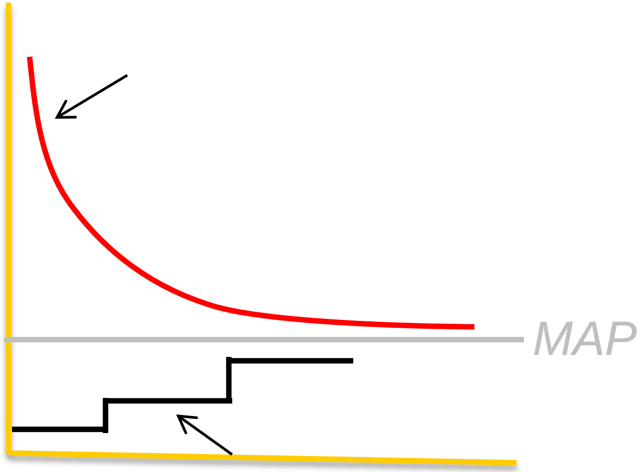
- Can tighten the bound in various ways
  - Cost-shifting (improve consistency between cliques)
  - Increase i-bound (higher order consistency)
- Simple moment-matching step improves bound significantly

Outline

- Graphical models, Queries, Inference vs search
- Inference Algorithms: bucket-elimination
- AND/OR search spaces and AND/OR BnB
- Bounded Inference: a) mini-bucket, b) cost-shifting
- Generating heuristics using mini-bucket elimination
- AND/OR Heuristic Search: Map and Marginal Map
- Conclusion

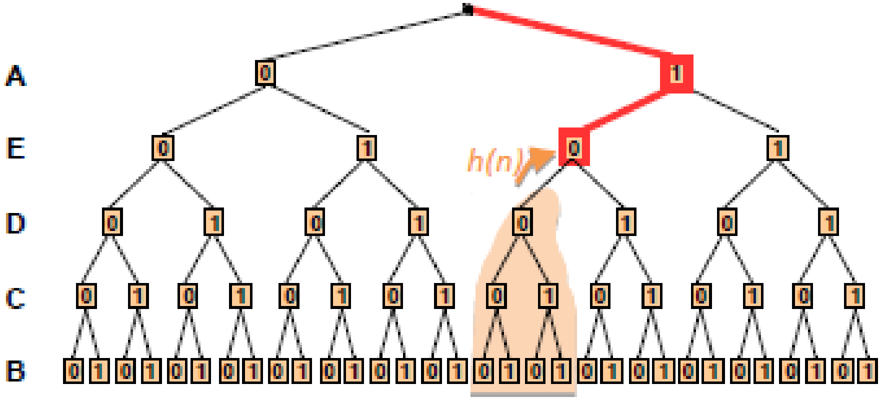
How to design a good Optimization solver (MAP)

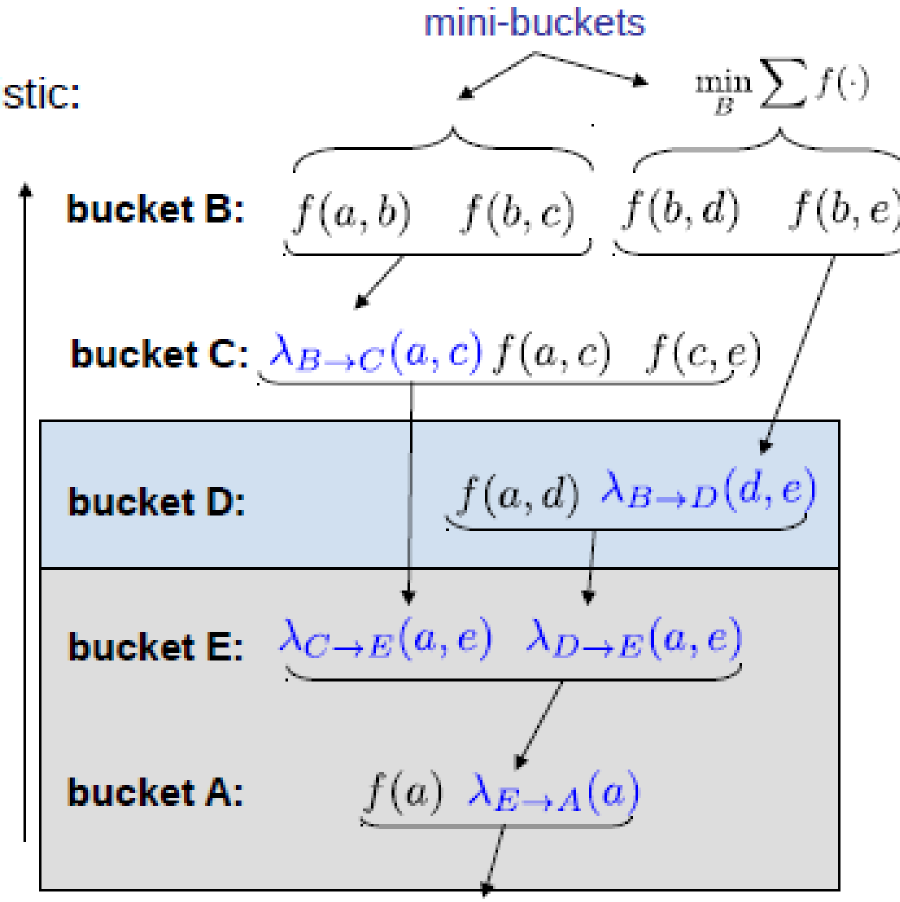
- Heuristic Search
- The core of a good search algorithm
  - □ A compact search space
  - □ A good heuristic evaluation function
  - □ A good traversal strategy
- Anytime search yields a good approximation.



Static Mini-Bucket Heuristics

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





bucket B: $\underbrace{f(a,b) \quad f(b,c)} \quad \underbrace{f(b,d) \quad f(b,e)}$

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

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

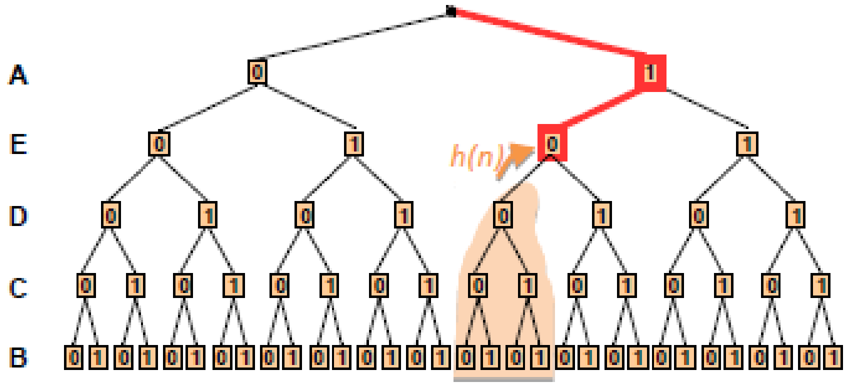
bucket E: $\lambda_{C\to E}(a,e) \quad \lambda_{D\to E}(a,e)$

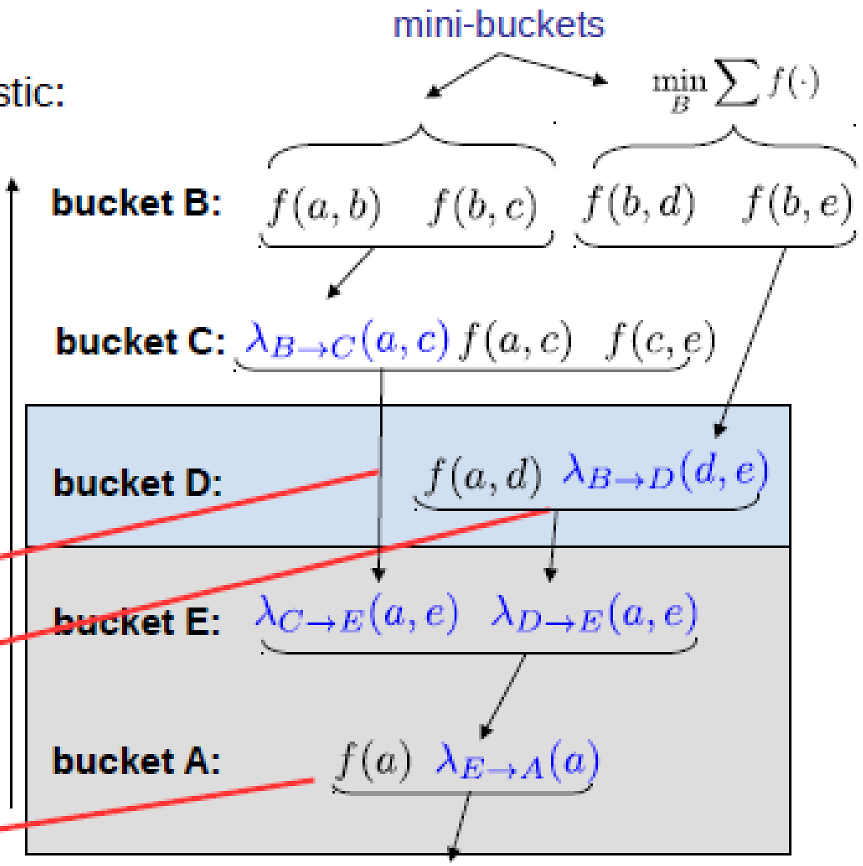
bucket A: $f(a) \quad \lambda_{E\to A}(a)$

L = lower bound

Static Mini-Bucket Heuristics

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




mini-buckets

$\min_B \sum f(\cdot)$

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

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

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

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

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

L = lower bound

cost to go:

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

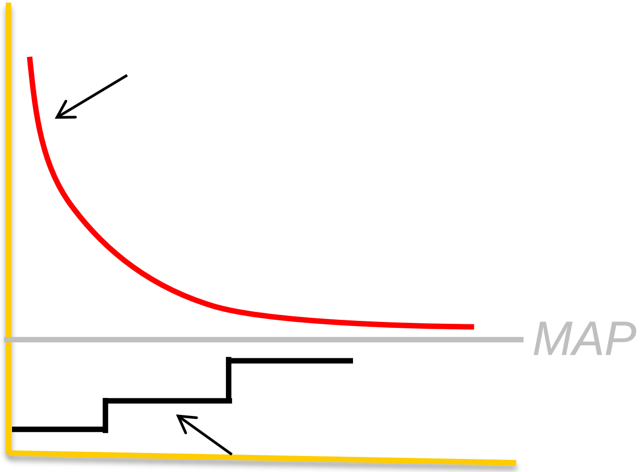
(admissible: $\tilde{h}(\hat{a},\hat{e},D) \le h^*(\hat{a},\hat{e},D)$)

cost so far:

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

How to design a good Optimization solver (MAP)

- 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 vs search
- Inference Algorithms: bucket-elimination
- AND/OR search spaces and AND/OR BnB
- Bounded Inference: a) mini-bucket, b) cost-shifting
- Generating heuristics using mini-bucket elimination
- AND/OR Heuristic Search: Map and Marginal Map
- Conclusion

Searching AND/OR Space Solves all Queries

MAP: AND/OR search

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

MMAP: AND/OR search

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

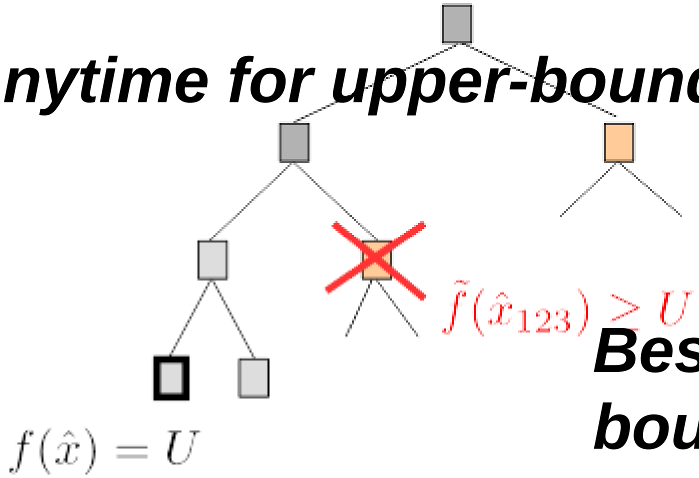
Basic Heuristic Search Schemes

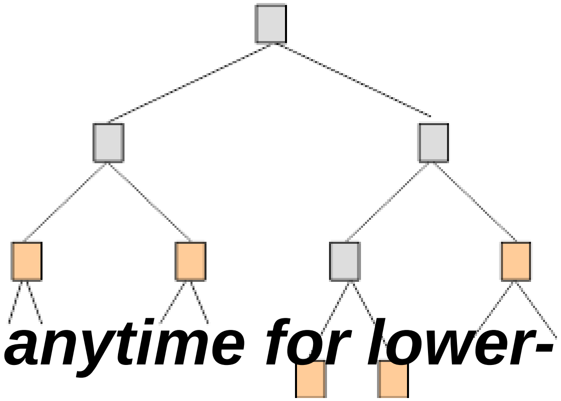
Heuristic function $\tilde{f}(\hat{x}_p)$ computes a lower bound on the best extension of partial configuration $\hat{x}_p$ and can be used to guide heuristic search.
We focus on:

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

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

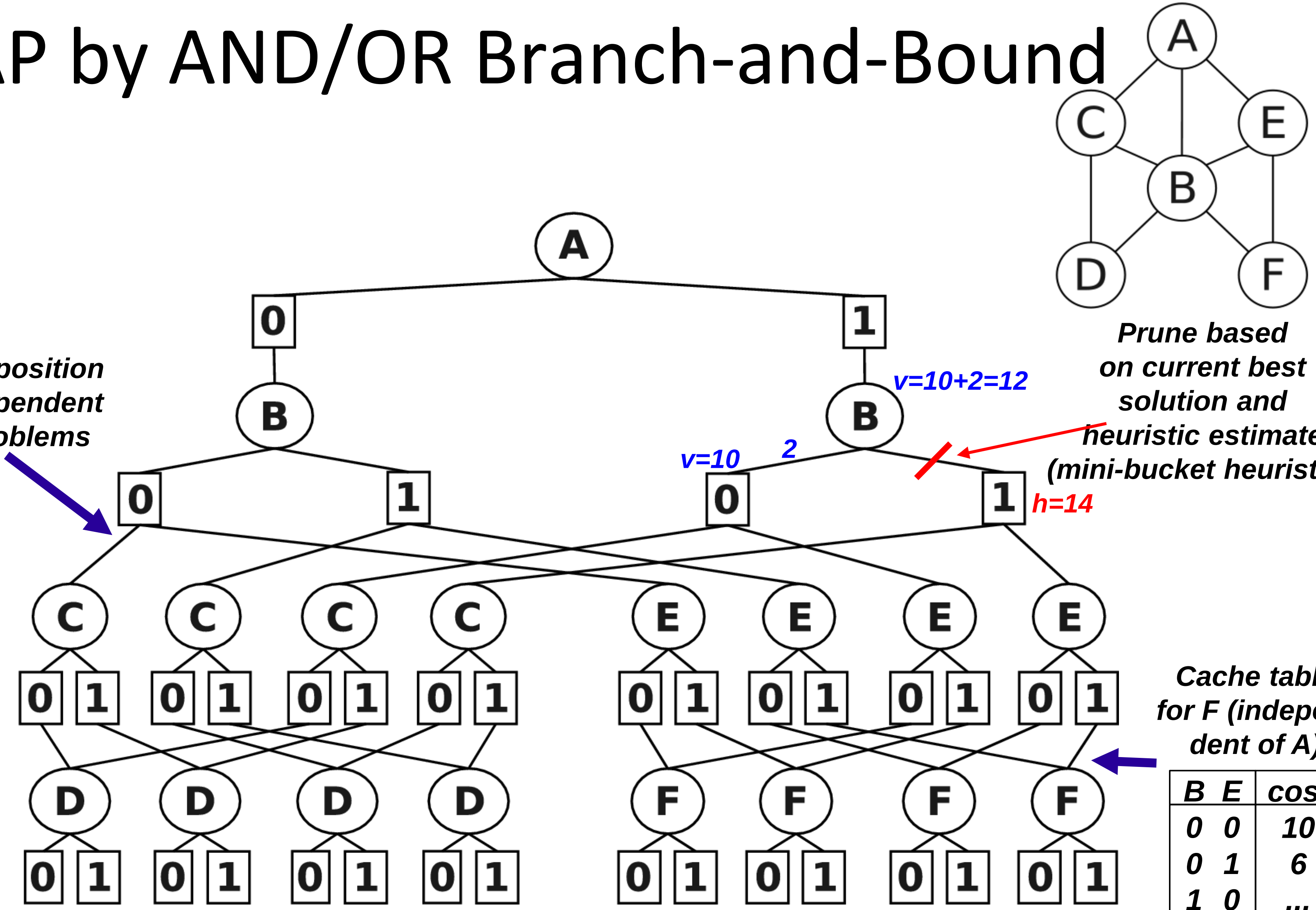
BnB is Anytime for upper-bound





Best-first: anytime for lower-bounds

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)

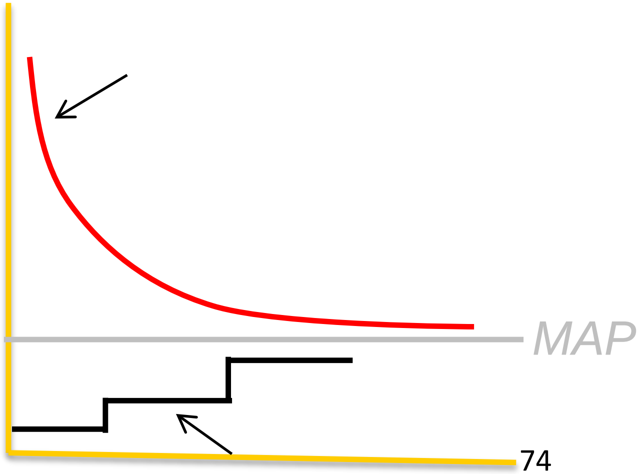
B	*E*	*cost*
0	*0*	*10*
0	*1*	*6*
1	*0*	*...*
1	*1*	*...*

[Marinescu & Dechter, AIJ'09]

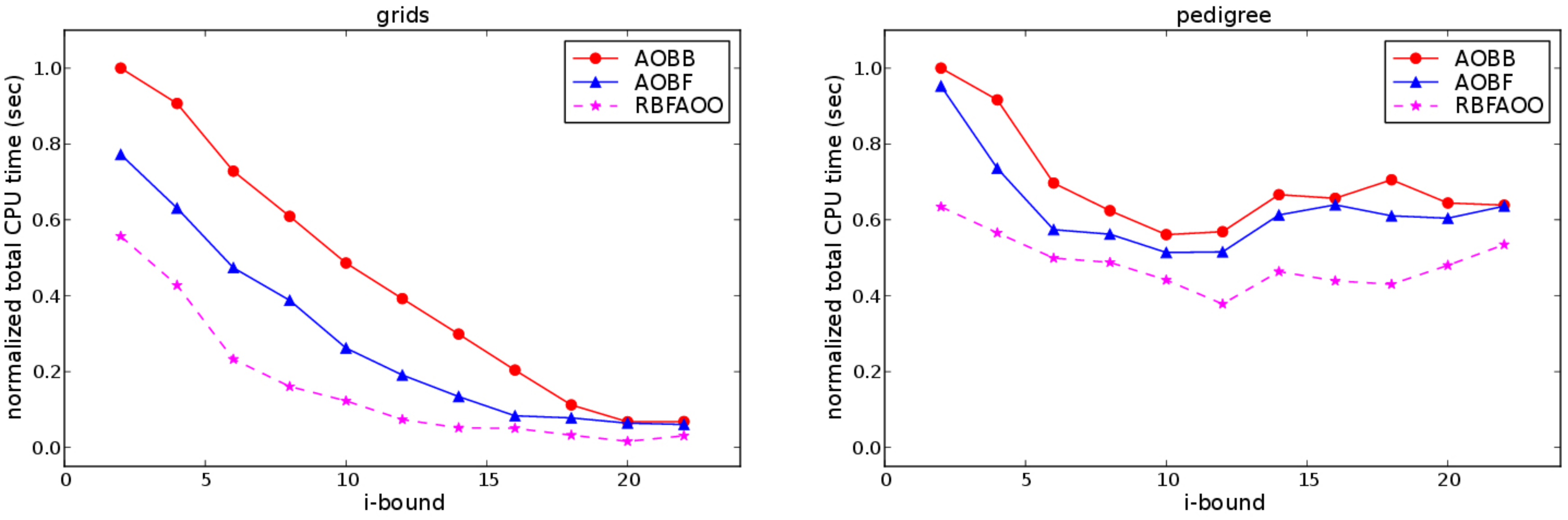
MAP: Anytime

- AOBF, AOBB (Best-First, BnB, Recursive)
- Breadth-Rotate AND/OR BnB (Otten & Dechter, 2011)
- Weighted heuristic AND/OR search (Flerova, et. Al, 2014)
- Parallel AOBB (Otten et. Al., 2012)
- Look-ahead AOBB and AOBF (Lam, at. Al, 2016)
- Finding m-best solutions (Flerova et. Al, 2015)

- ***Memory, Time, Accuracy***
- ***Extensive empirical evaluation of upper-bound***
- ***Won 2 UAI competitions***

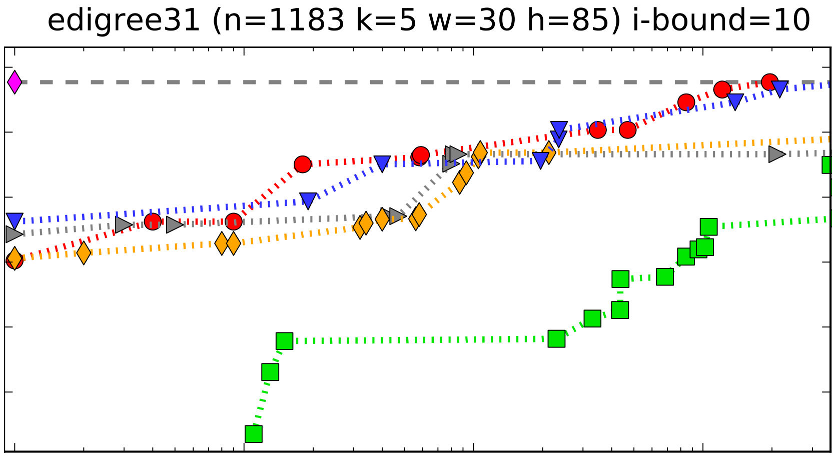


Empirical Evaluation; Exact

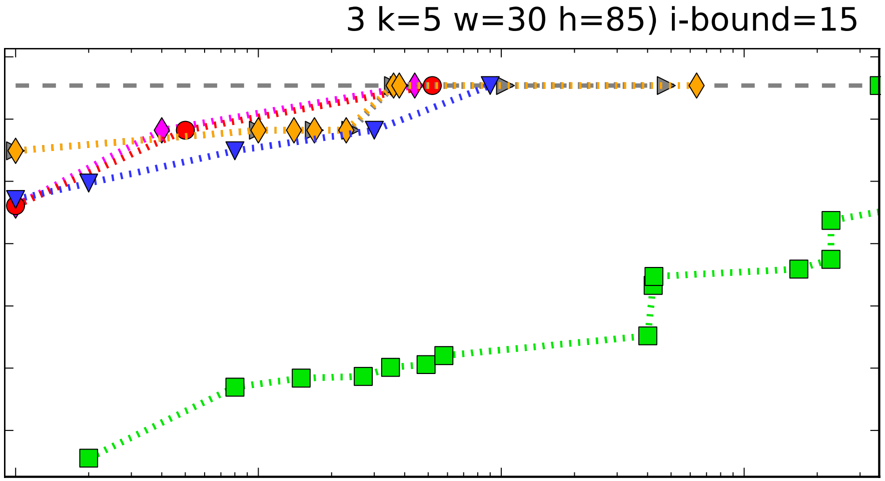


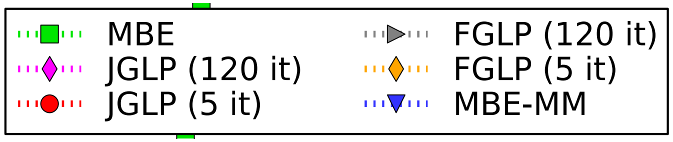
Exact MAP inference. Grid and Pedigree benchmarks. Time limit 1 hour.

Empirical Evaluation; Anytime: Haplotype problems



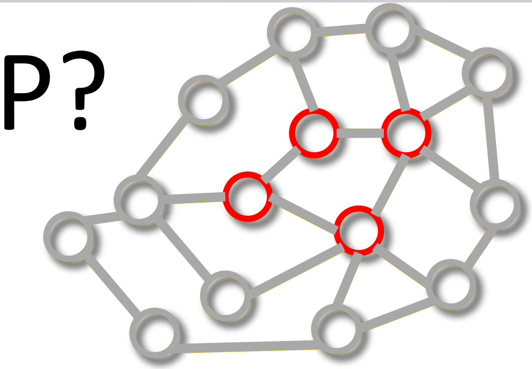
Time bound – 24 h
3 GB memory
i-bound(5,10,20)





AOBB-MBE:
AOBB-MBE+MM
AOBB-FGLP+MBE
AOBB-JGLP

MMAP: Why Marginal MAP?

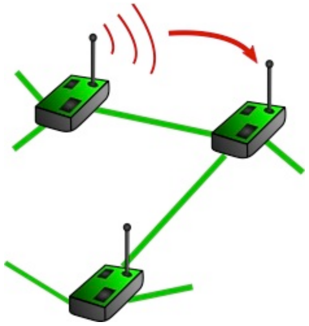


- Often, Marginal MAP is the “right” task:
  - □ We have a model describing a large system
  - □ We care about predicting the state of some part

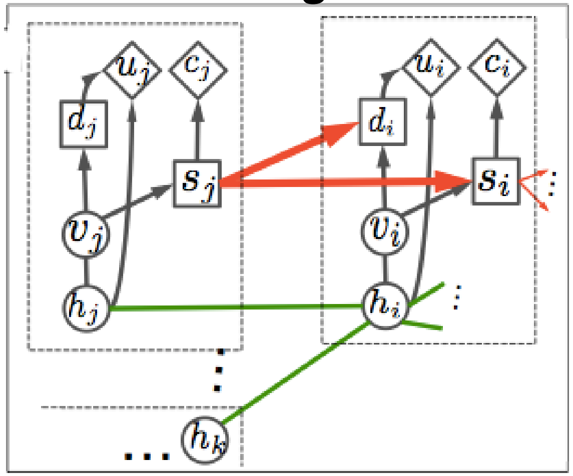
$$\boldsymbol{x}_B^* = \arg\max_{\boldsymbol{x}_B} \sum_{\boldsymbol{x}_A} \prod_{\alpha} \psi(\boldsymbol{x}_\alpha)$$

- Example: decision making
  - □ Sum over random variables (random effects, etc.)
  - □ Max over decision variables (specify action policies)

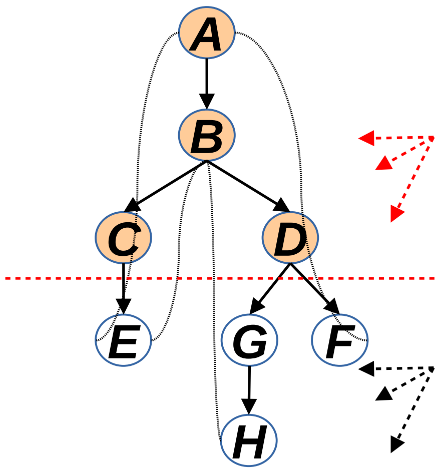
Sensor network



Influence diagram:



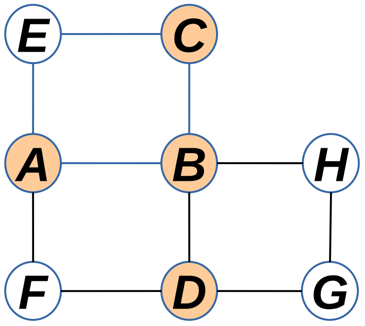
MMAP: AND/OR Search Spaces for MMAP



***MAP* variables**

***SUM* variables**

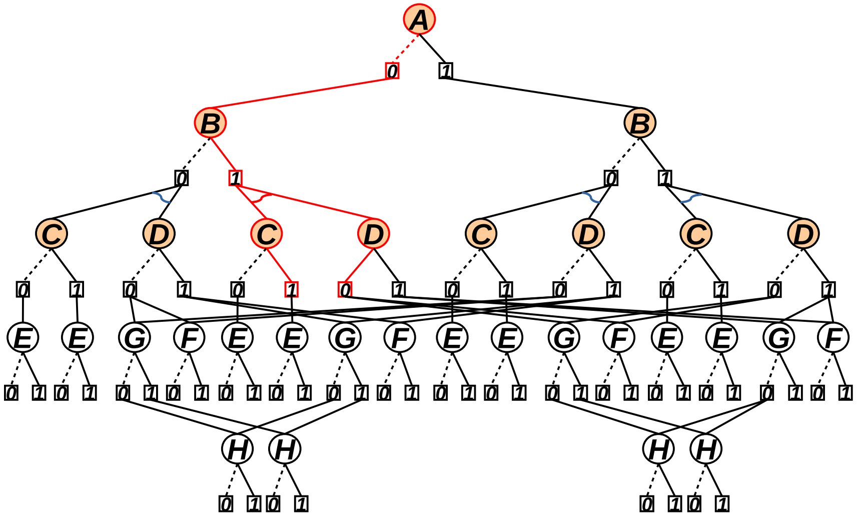
constrained pseudo tree



primal

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

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



[Marinescu, Dechter and Ihler, 2014]

Marginal Map results

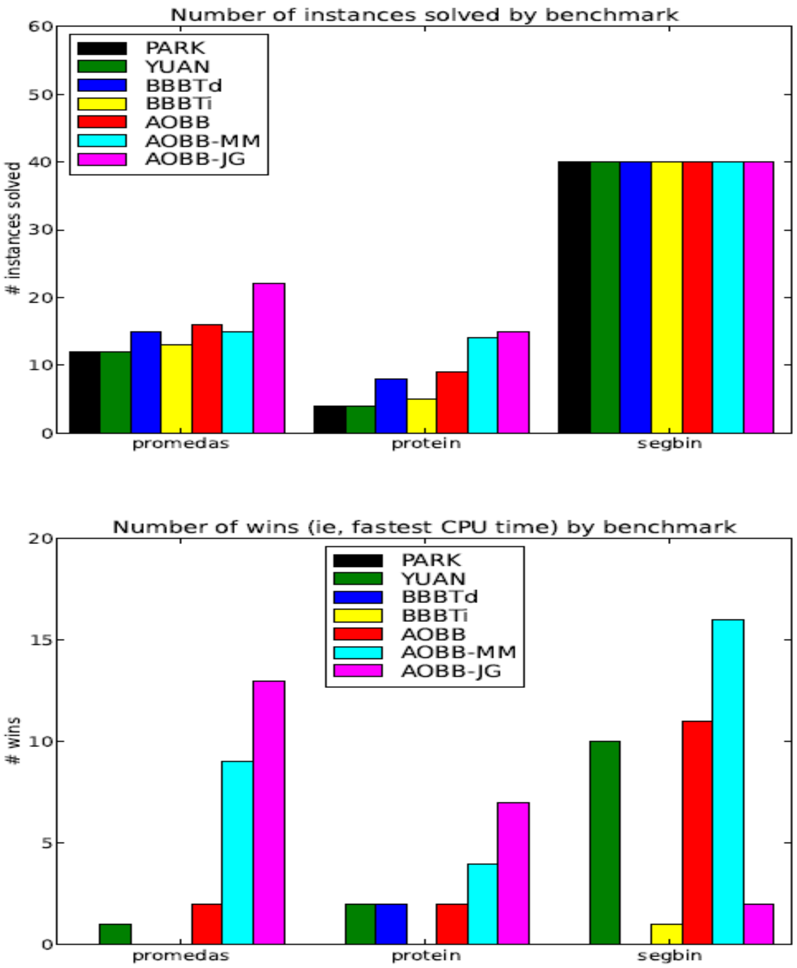


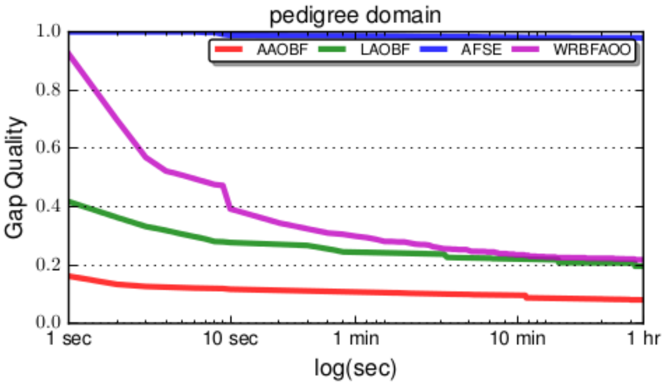
Figure 5: Number of instances solved (top) and number of wins (bottom) by benchmark.

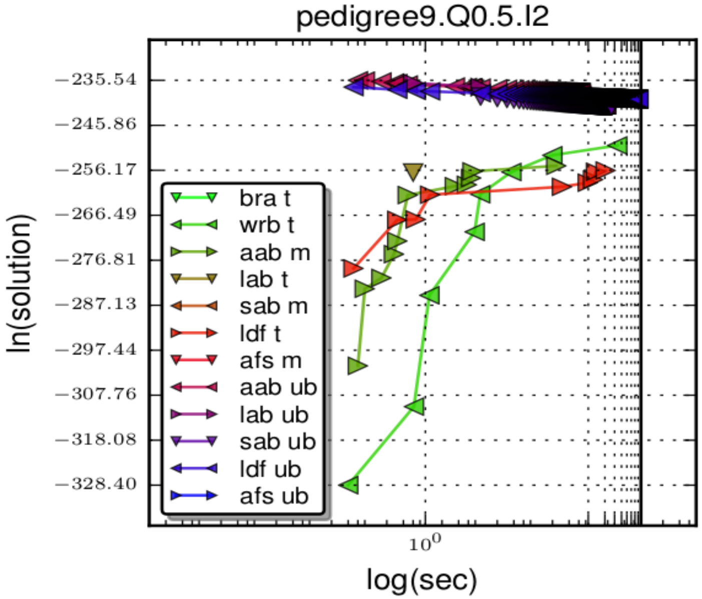
Marginal MAP

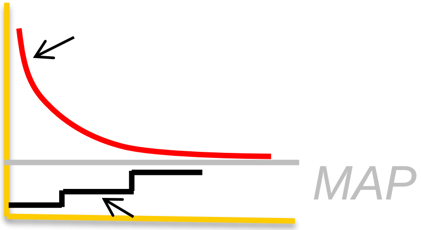
(UAI'14, IJCAI'15, AAAI'16, AAAI'17, (Marinescu, Lee, Ihler, Dechter)

Algorithms: AO best or depth with WMB+MM heuristic

- Balance best-first behavior, Quickly tighten upper & lower bounds
- vs depth-first behavior, Quickly find a (suboptimal) solution Series of improvements in performance







UAI Probabilistic Inference Competitions

- **2006** 🥇 ***(aolib)***
- **2008** 🥈 ***(aolib)***
- **2011** 🥇 ***(daoopt)***
- **2014** 🥈 ***(daoopt)*** 🥇 ***(daoopt)*** 🥈 ***(merlin)***

MPE/MAP

MMAP

Partition function

(AAAI-2017, Liu, Dechter and Ihler)

- Anytime performance

- Message-passing bounds

Incremental construction (UAI’15a)

Optimization algorithm (NIPS’15b)

Use as heuristic for true anytime algorithms

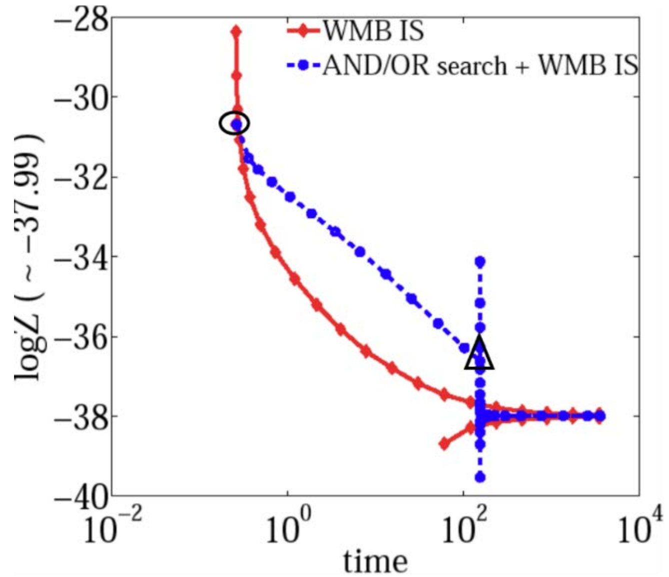
Search methods

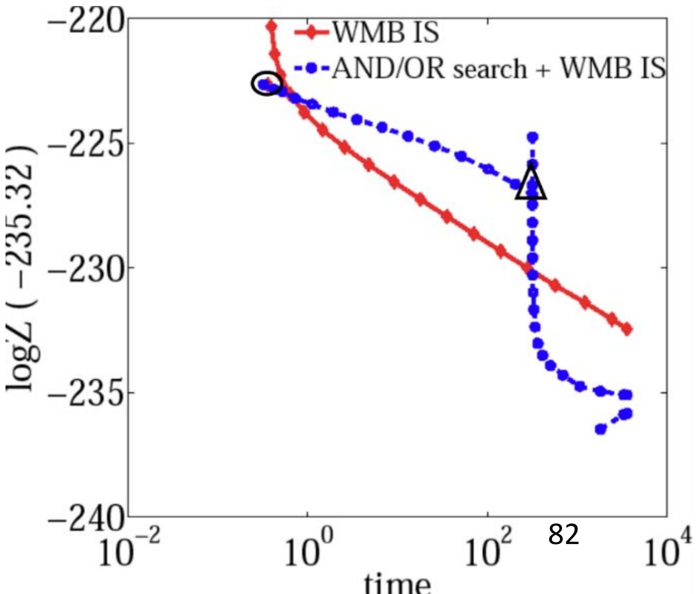
Memory-limited best-first (AAAI’17)

Sampling methods

Probabilistic bounds via IS (NIPS’15a)

Also: discriminance sampling (UAI’15b)

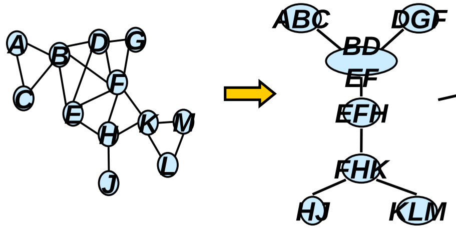


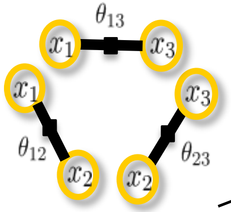


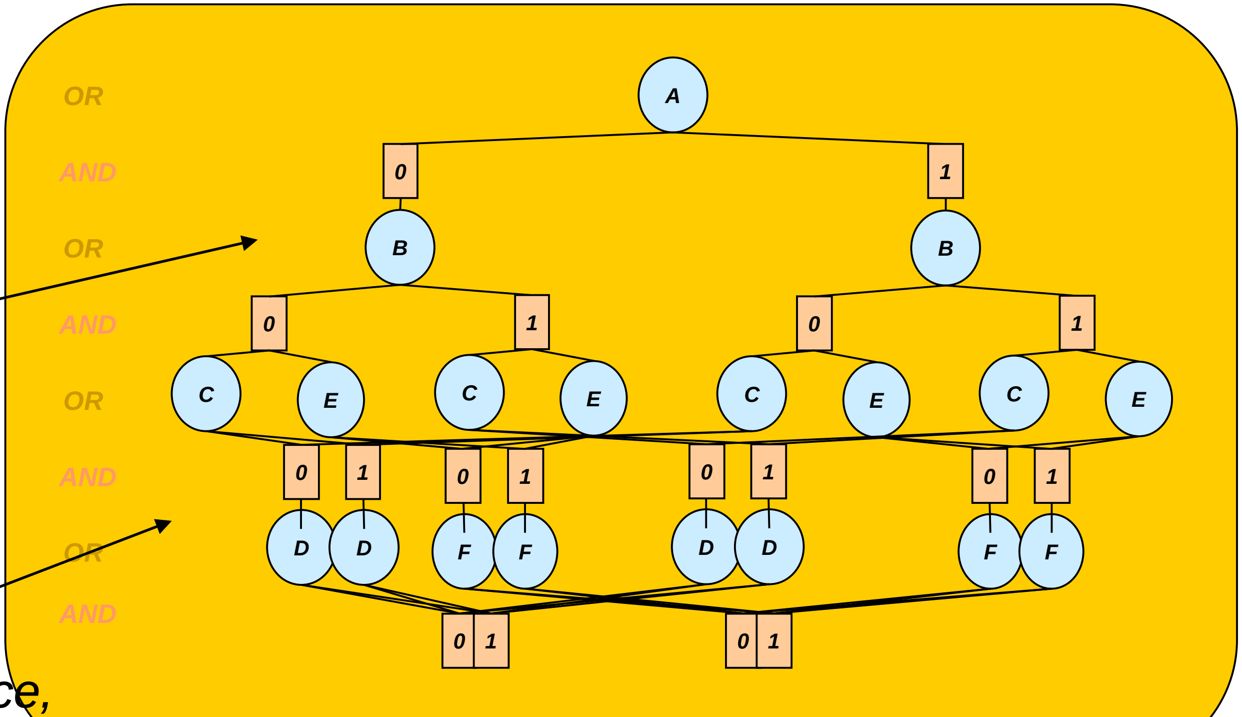
Conclusion: Search Swallows Inference

Inference

exp(w*) time/space



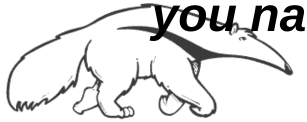


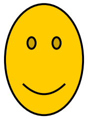


Context minimal AND/OR search graph

h(n): bounded inference,
mini-buckets+ moment matching
(exp, i-bound)

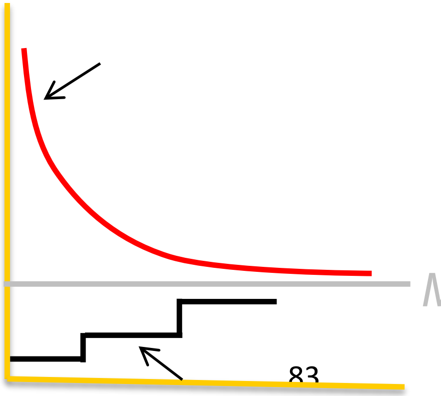
For:
MAP, MMAP, Marginals
Max-expected utility,
you name it





Anytime behavior

Best-first → upper-bounds
Depth-first → lower-bounds
Interaction prunes the space



Software

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

  (standalone AOBB, AOBF solvers)
- **daoopt**
  - https://github.com/lotten/daoopt

  (distributed and standalone AOBB solver)

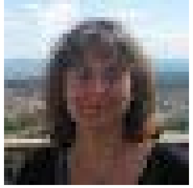
Our solvers are being used at:

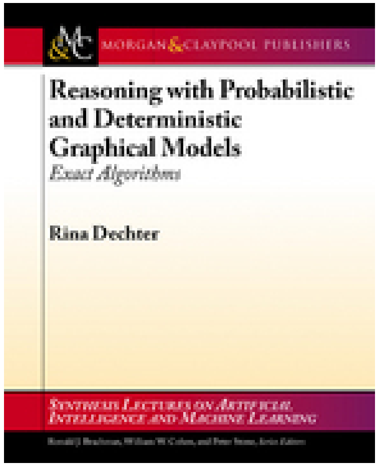
- *Super link online, software for linkage analysis (Geiger et. Al)*
- *Figaro, probabilistic language (Avi Pfeffer)*

Thank You !

For publication see:

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





MORGAN & CLAYPOOL PUBLISHERS

Reasoning with Probabilistic and Deterministic Graphical Models

Exact Algorithms

Rina Dechter

SYNTHESIS LECTURES ON ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

Ronald J. Brachman, William W. Cohen, and Peter Stone, *Series Editors*

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