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

Modern Exact and Approximate MAP algorithms for Graphical Models

Rina Dechter

Donald Bren school of ICS, University of California, Irvine

Joint work with
Natasha Flerova, Alex Ihler and Lars Otten

Harvard, 10/12

Outline

- Graphical models and the MAP task
- Solving exactly: Variable elimination and search
- Upper bound approximations
  - The mini-bucket scheme
  - The cost-shifting or reparameterization scheme
- New Algorithms combinations
- Experiments

Sample Applications for Graphical Models

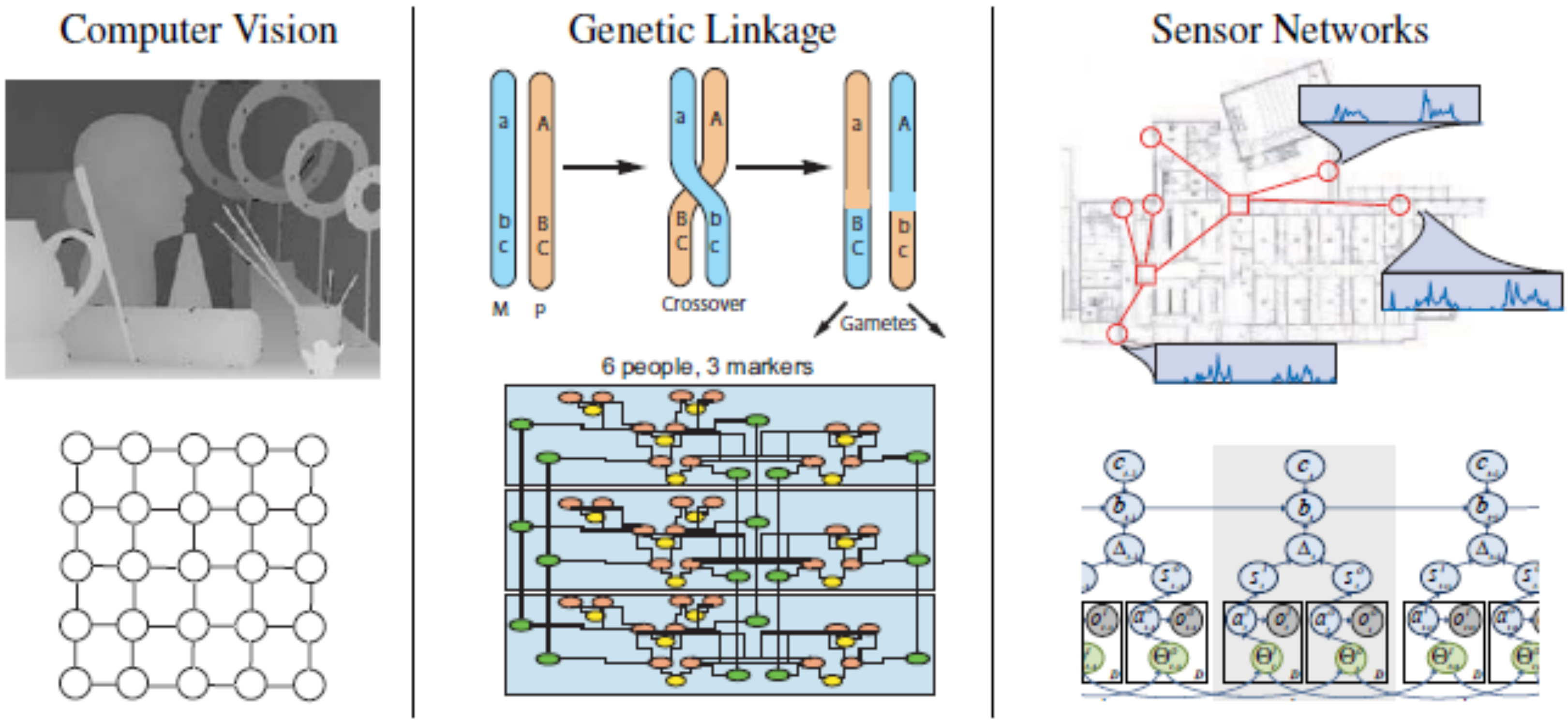
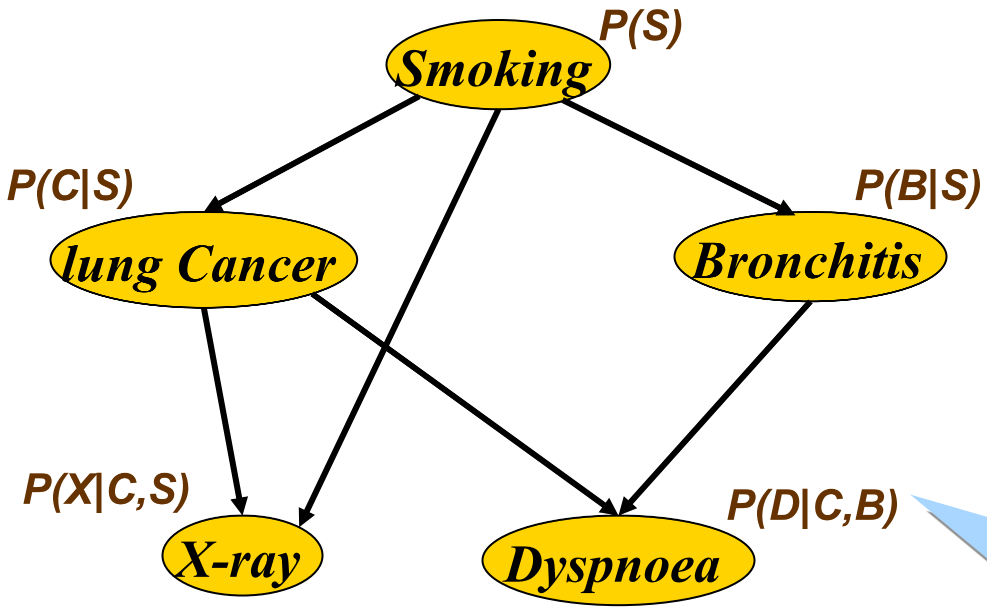


Figure 1: Application areas and graphical models used to represent their respective systems: (a) Finding correspondences between images, including depth estimation from stereo; (b) Genetic linkage analysis and pedigree data; (c) Understanding patterns of behavior in sensor measurements using spatio-temporal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

Bayesian Networks (Pearl 1988)



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

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

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

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

$$mpe = \max_x P(x)$$

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

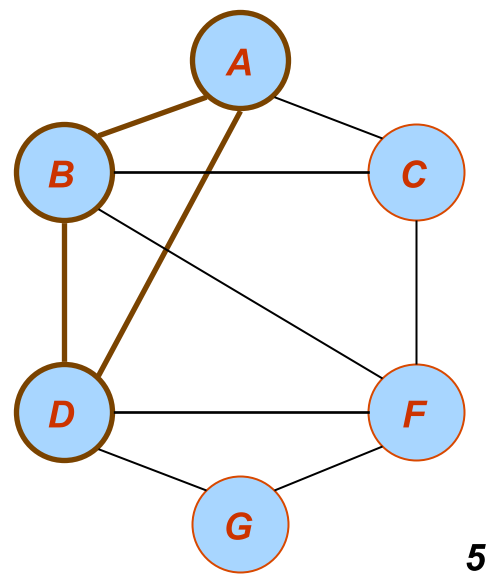
$D = \{D_1, ..., D_n\}$ - domains

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

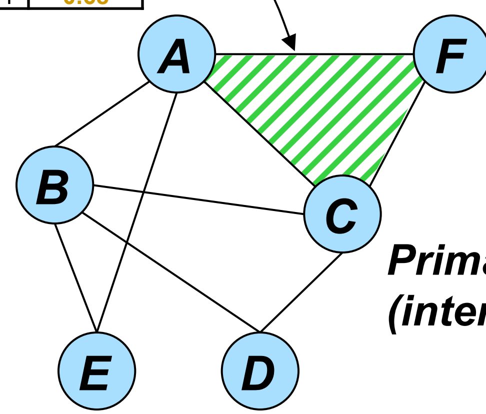
$$F(X) = \sum_{i=1}^{m} f_i(X)$$

Graphical Models

Conditional Probability or potential

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

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



Primal graph (interaction graph)

- Tasks:
  - **Belief updating**: $\Sigma_{X-y} \prod_j P_i$
  - **MPE**: $\max_X \prod_j P_j$

The MRF

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

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

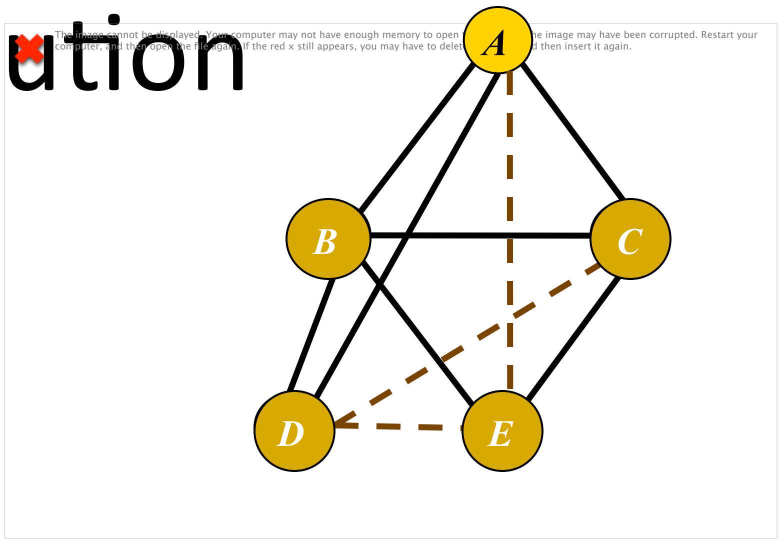
PASCAL 2011 Probabilistic Inference Challenge

- http://www.cs.huji.ac.il/project/PASCAL/
- Evaluates solvers in three categories:
  - PR: Probability of evidence / partition function
  - MAR: Posterior node marginals
  - MPE: Most probable explanation (our entry)
- Three tracks each: 20 sec, 20 min, 1 hour.
- Variety of benchmark domains:
  - CSPs, Deep Belief Nets, Image Alignment and Segmentation, Object Detection, Protein Folding, …

Outline

- Graphical models and the MAP task
- Solving exactly: Variable elimination and search
- Upper bound approximations
  - The mini-bucket scheme
  - The cost-shifting or reparameterization scheme
- New Algorithms combinations
- Experiments

Computing the Optimal Cost Solution



$$OPT = \min_{e=0,d,c,b} f(a,b)f(a,c)f(a,d)f(b,c)f(b,d)f(b,e)f(c,e) =$$

Combination

$$\min_{e=0}\min_{d} f(a,d) \cdot \min_{c} f(a,c)f(c,e) \min_{b} f(a,b)f(b,c)f(b,d)f(b,e)$$

$$h^{B}(a,d,c,e)$$

Variable Elimination

Inference by Variable 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: $P(c|a)$ $h^B(a,d,c,e)$

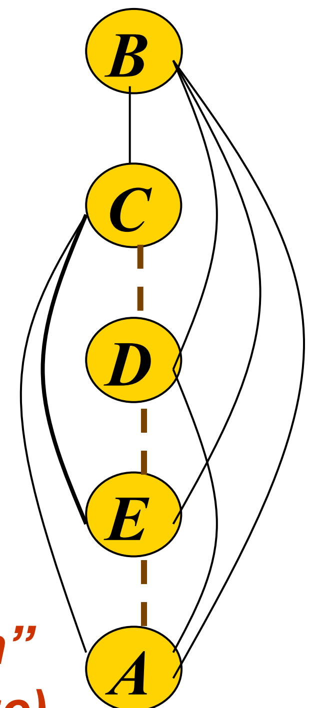
bucket D: $h^C(a,d,e)$

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

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

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

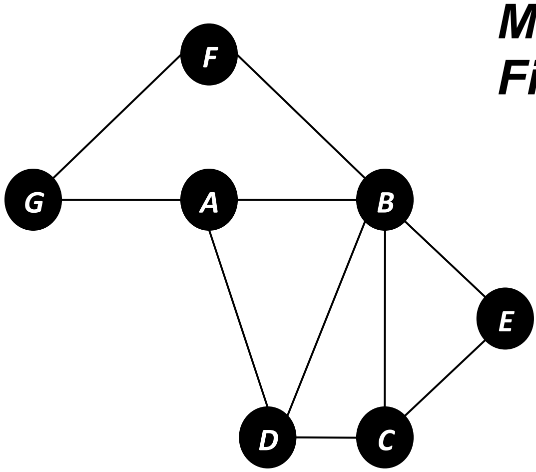
E: $e=0$ $h^D(a,e)$

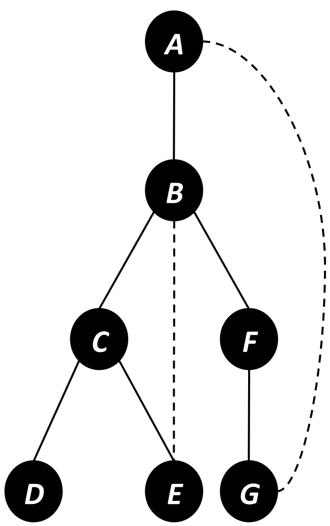
A: $P(a)$ $h^E(a)$

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

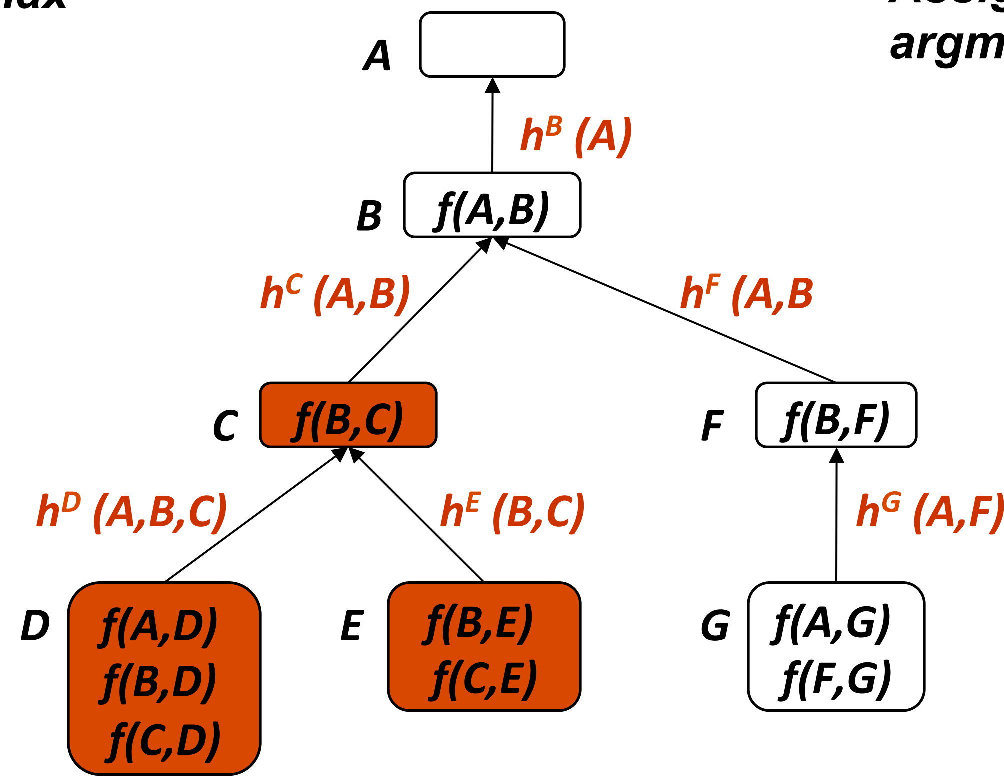
Bucket Elimination

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





Messages Finding max



Assignment argmax

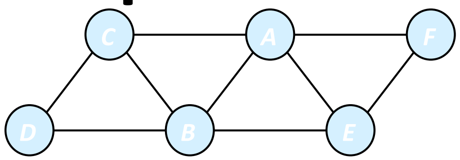
$$O(n\exp(w^*(d)))$$

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

Outline

- Introduction
- Inference
- **Search (OR)**
  - Branch-and-Bound and Best-First search
- Lower-bounds and relaxations
- Exploiting problem structure in search
- Software

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

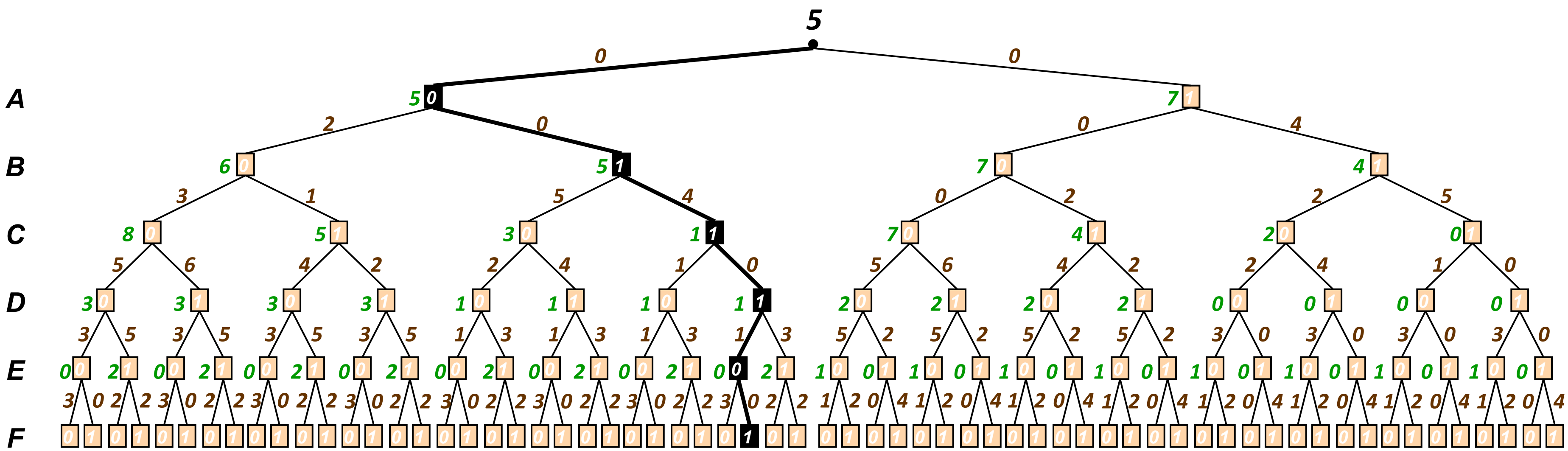
B	E	f_7
0	0	3
0	1	2
1	0	1
1	1	0

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

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

$$\min_{a,b,c,d,e,f} f_1(a,b) + f_2(a,c) + f_3(a,f) + f_4(b,c) + f_5(b,d) + f_6(b,e) + f_7(c,d) + f_8(e,f)$$

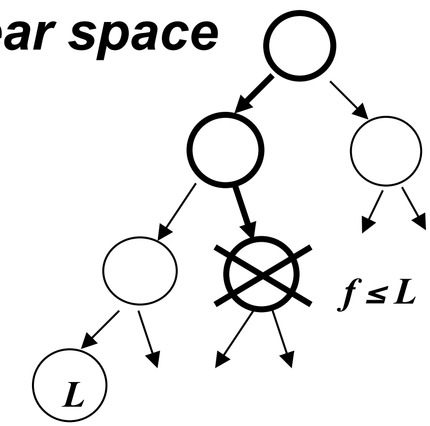


An optimal assignment is A=0, B=1, C=1, D=1, E=0, F=1 with cost 5

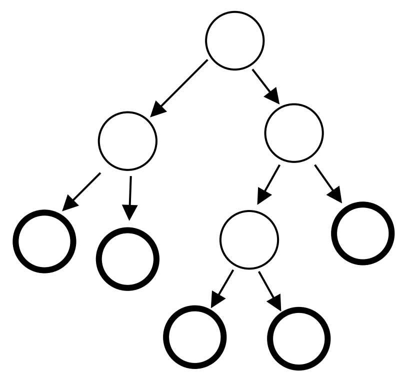
Basic Heuristic Search Schemes

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

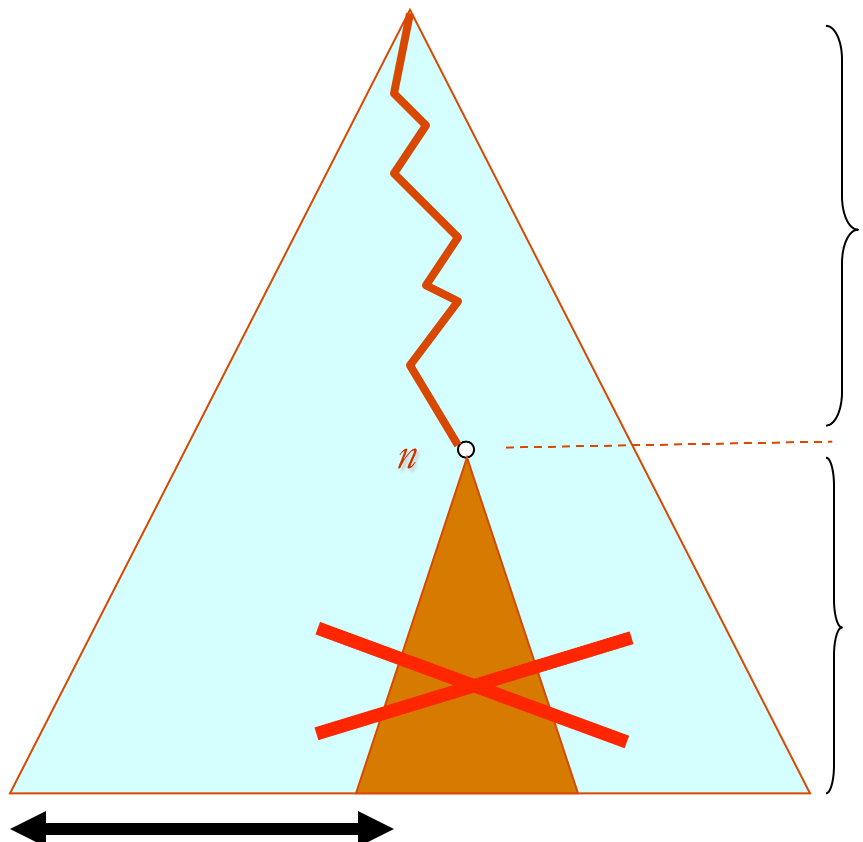
1. Branch-and-Bound
Use heuristic function $f(x^p)$ to prune the depth-first search tree
Linear space



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



Classic Branch-and-Bound



g(n)

f(n) = g(n) + h(n)
f(n) = lower bound

Prune if f(n) ≥ UB

h(n) - under-estimates
Optimal cost below n

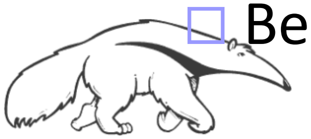
(UB) Upper Bound = best solution so far

Outline

- Graphical models and the MAP task
- Solving exactly: Variable elimination and search
- Bound (lower or upper) approximations
  - The mini-bucket scheme
  - The cost-shifting or re-parameterization scheme
- New Algorithms combinations
- Experiments

Primary Bounding Schemes

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



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

Split a bucket into mini-buckets =>bound complexity

$$\textbf{bucket}(X) = \{ h_1, \ldots, h_r, h_{r+1}, \ldots, h_n \}$$

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

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

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

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

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

Mini-Bucket Elimination

$max_B \prod$ $max_B \prod$

Node duplication, renaming

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

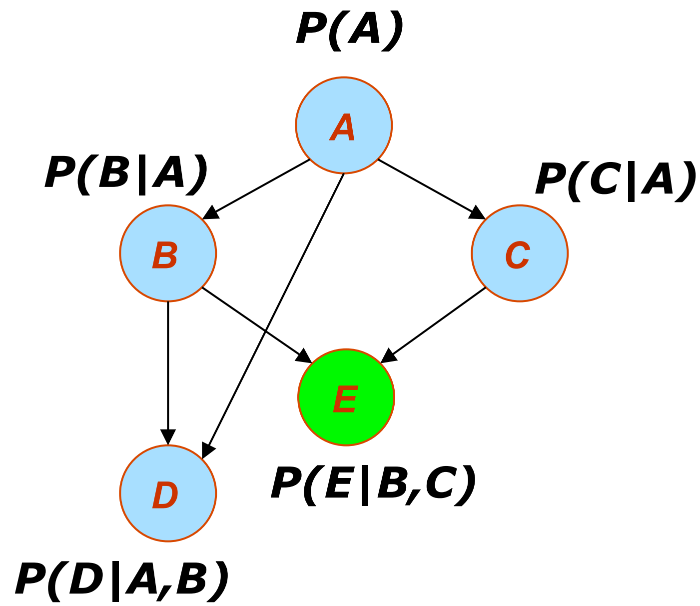
Bucket C **P(C|A) h^B (C,E)**

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

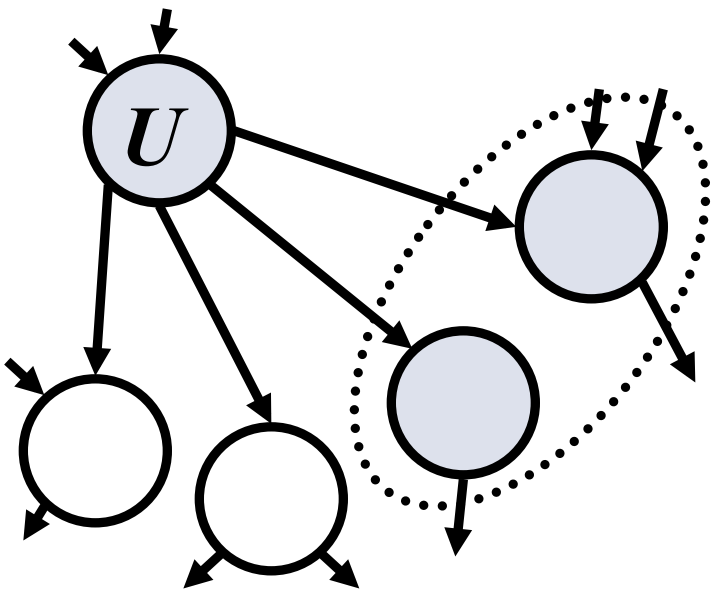


MPE* is an upper bound on MPE --U
Generating a solution yields a lower bound--L

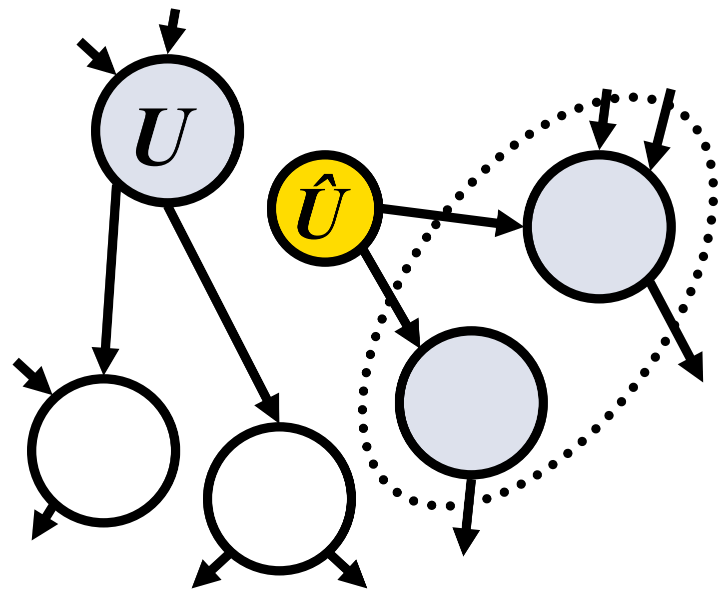
Semantics of Mini-Bucket: Splitting a Node

Variables in different buckets are renamed and duplicated
(Kask and Dechter, 2001), (Geffner et. al., 2007), (Choi, Chavira, Darwiche , 2007)

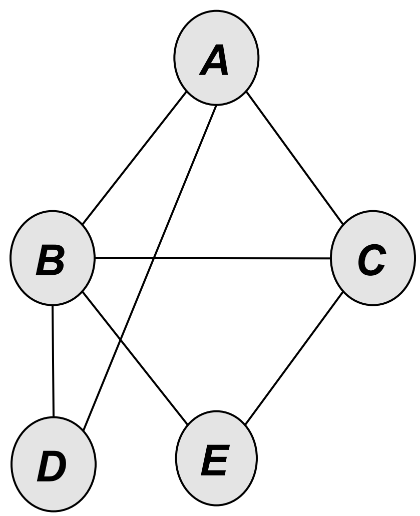
Before Splitting:
Network N



After Splitting:
Network N'



Mini-Bucket Elimination Semantic



Mini-buckets

$\min_B \Sigma$ $\min_B \Sigma$

bucket B: *F(a,b') F(b',c)*   *F(b,d) F(b,e)*

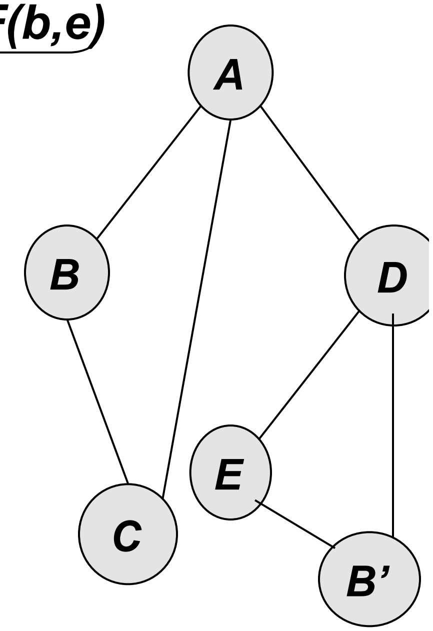
bucket C: *h^B(a,c) F(c,e) F(a,c)*

bucket D: *F(a,d) h^B(d,e)*

bucket E: *h^C(e,a) h^D(e,a)*

bucket A: *h^E(a)*

L = lower bound

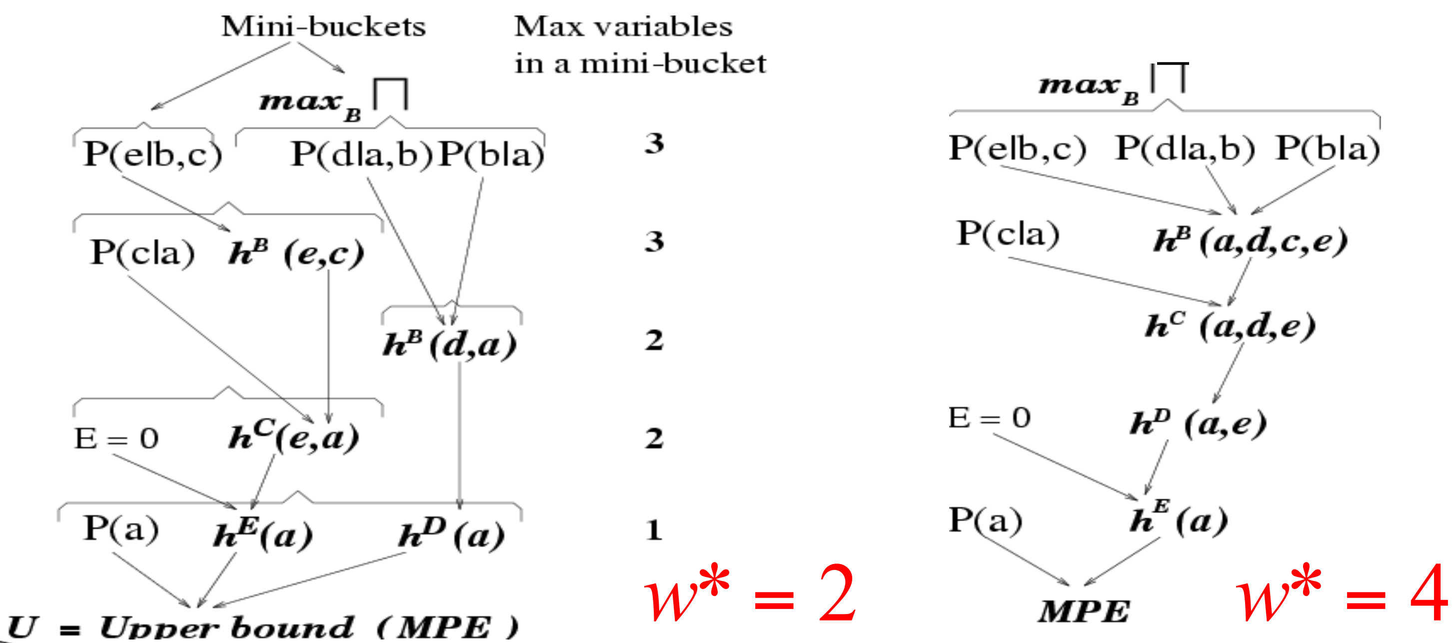


MBE-MPE(i)

Algorithm **Approx-MPE** (Dechter & Rish, 1997)

- **Input**: i – max number of variables allowed in a mini-bucket
- **Output**: [lower bound (P of a sub-optimal solution), upper bound]

Example: approx-mpe(3) versus elim-mpe

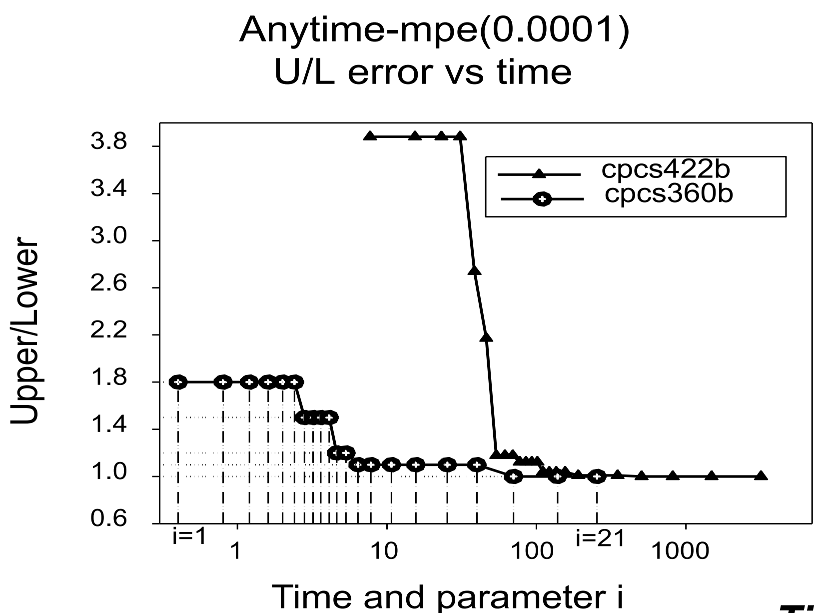


Properties of MBE(i)

- **Complexity:** *O(r exp(i))* time and *O(exp(i)) space*.
- Yields an upper-bound and a lower-bound.
- **Accuracy:** determined by upper/lower (U/L) bound.
- As ***i*** increases, both accuracy and complexity increase.
- Possible use of mini-bucket approximations:
  - As anytime algorithms
  - As heuristics in search
- Other tasks: similar mini-bucket approximations for: belief updating, MAP and MEU (Dechter and Rish, 1997)

CPCS Networks – Medical Diagnosis
(noisy-OR CPD's)

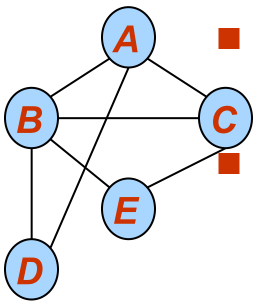
Test case: no evidence



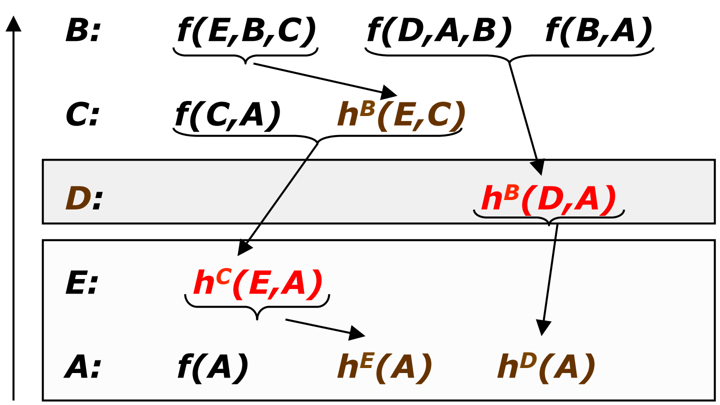
Algorithm	***Time (sec)*** ***cpcs360***	***cpcs422***
elim-mpe	***115.8***	***1697.6***
anytime-mpe(ε***),*** $\varepsilon = 10^{-4}$	***70.3***	***505.2***
anytime-mpe(ε***),*** $\varepsilon = 10^{-1}$	***70.3***	***110.5***

Static MBE Heuristics

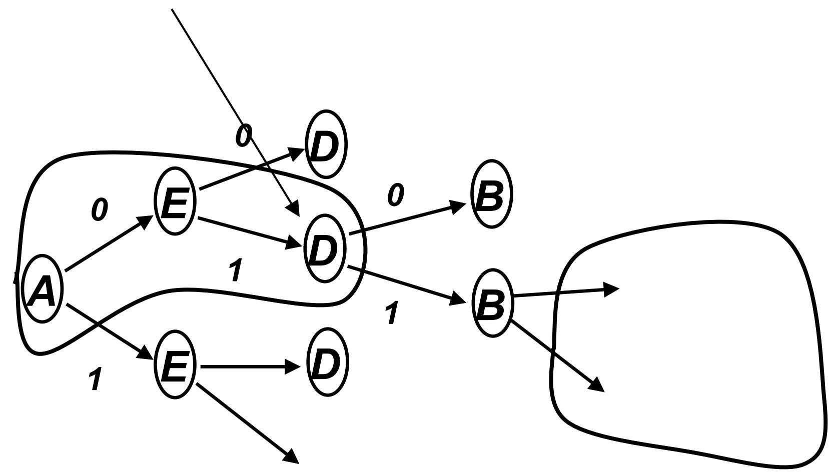
- Given a partial assignment x^p, estimate the cost of the best extension to a full solution
- The evaluation function $f(x^p)$ can be computed using function recorded by the Mini-Bucket scheme
- Heuristic is consistent and admissible.



Cost Network



$f(a,e,D))=g(a,e) + H(a,e,D)$

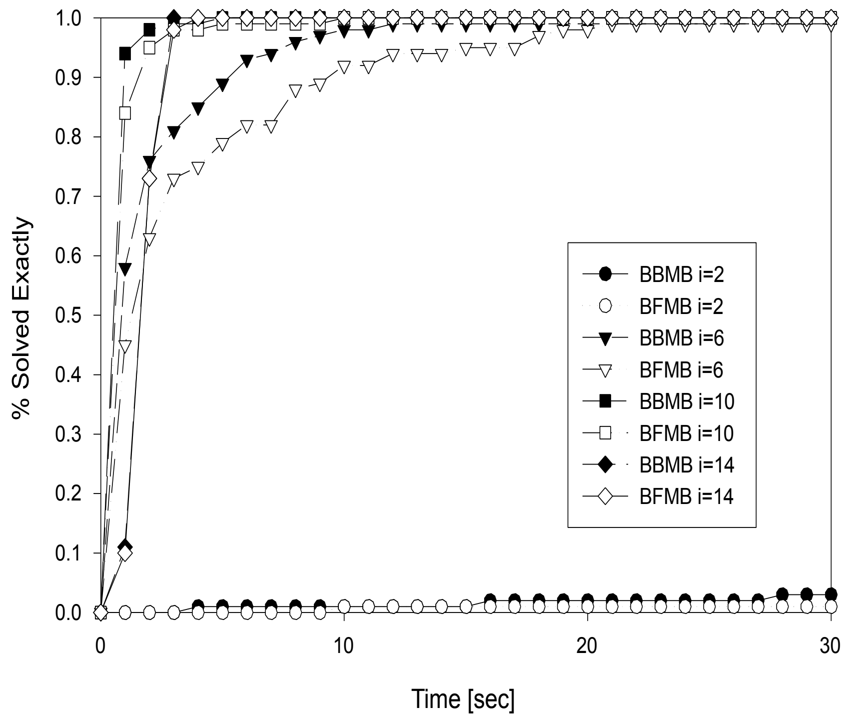


$$f(a,e,D) = \underbrace{f(a)}_{g} + \underbrace{h^B(D,a) + h^C(e,a)}_{h \text{ – is admissible}}$$

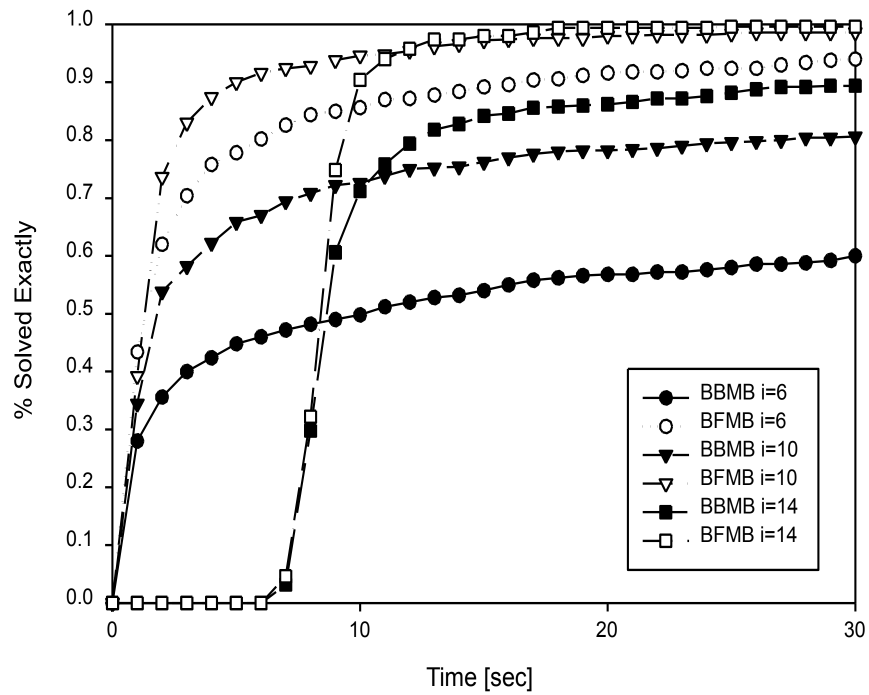
Empirical Evaluation of Mini-Bucket heuristics:

Random coding networks (Kask & Dechter, UAI'99, Aij 2000)

Random Coding, K=100, noise=0.28



Random Coding, K=100, noise=0.32



Each data point represents an average over 100 random instances

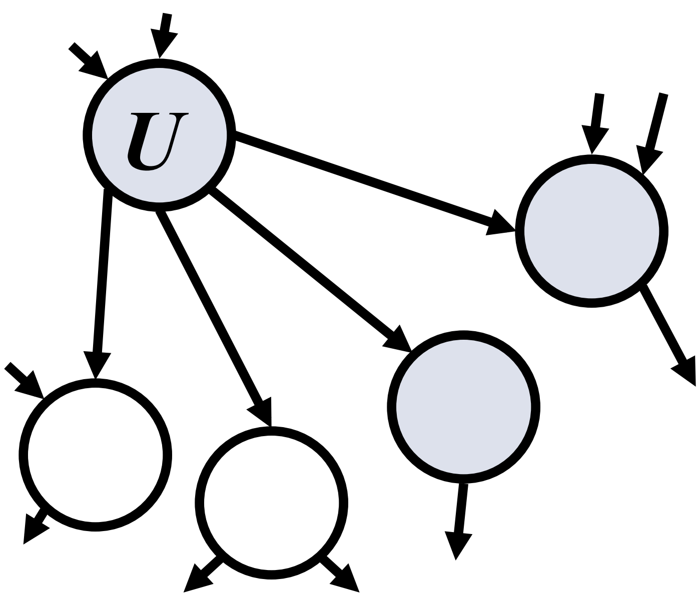
Outline

- Graphical models and the MAP task
- Solving exactly: Variable elimination and search
- Upper bound approximations
  - The mini-bucket scheme
  - The cost-shifting or reparameterization scheme
- New Algorithms combinations
- Experiments

Semantics of Dual Decomposition: Each Functions is a Mini-Bucket + Reparameterization

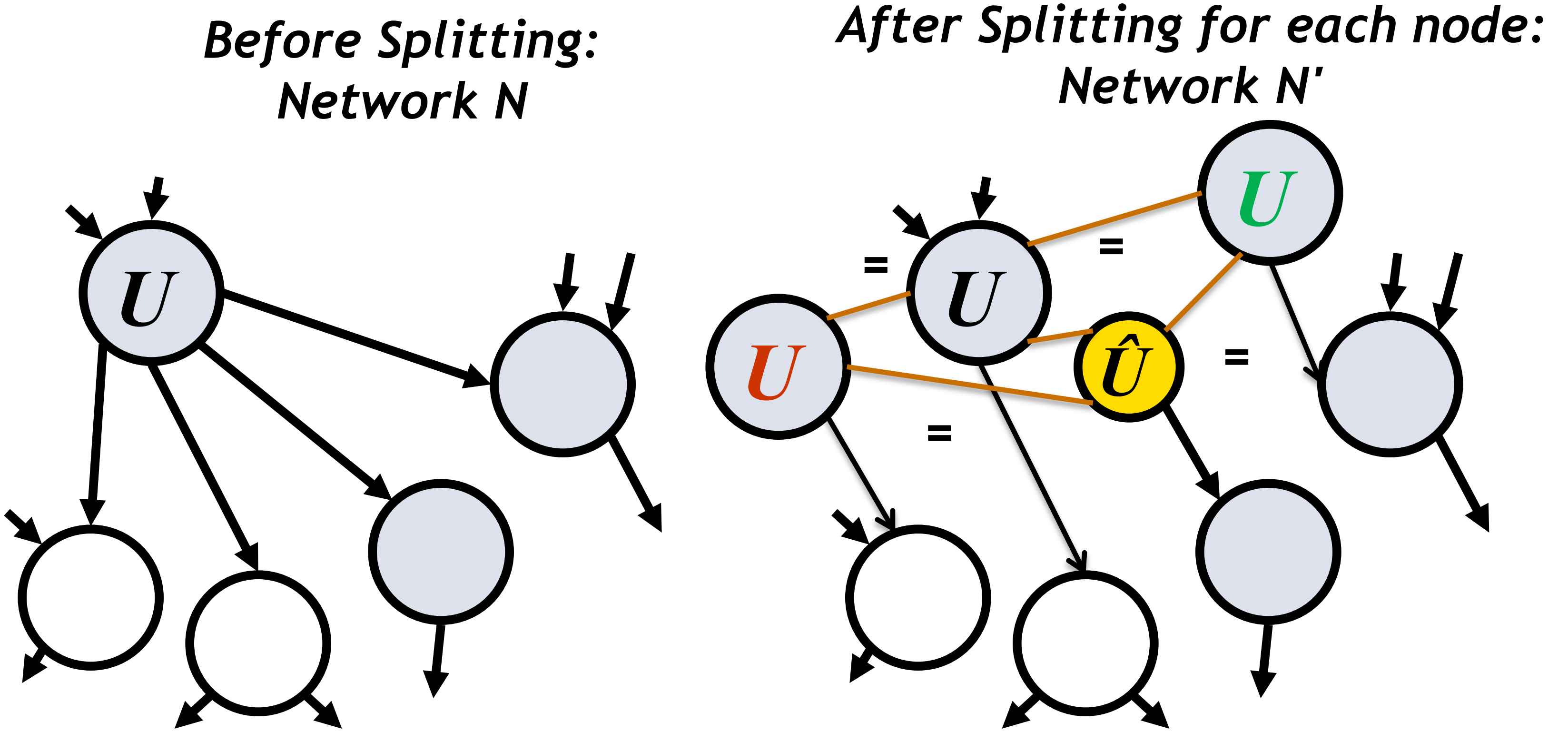
Variables in different buckets are renamed and duplicated (Globerson and Jakkola, 2008),

Before Splitting:
Network N



Re-parameterization: Duplicating a Node for Each Arc/Function

Variables in different buckets are renamed and duplicated (Globerson and Jakkola, 2008),

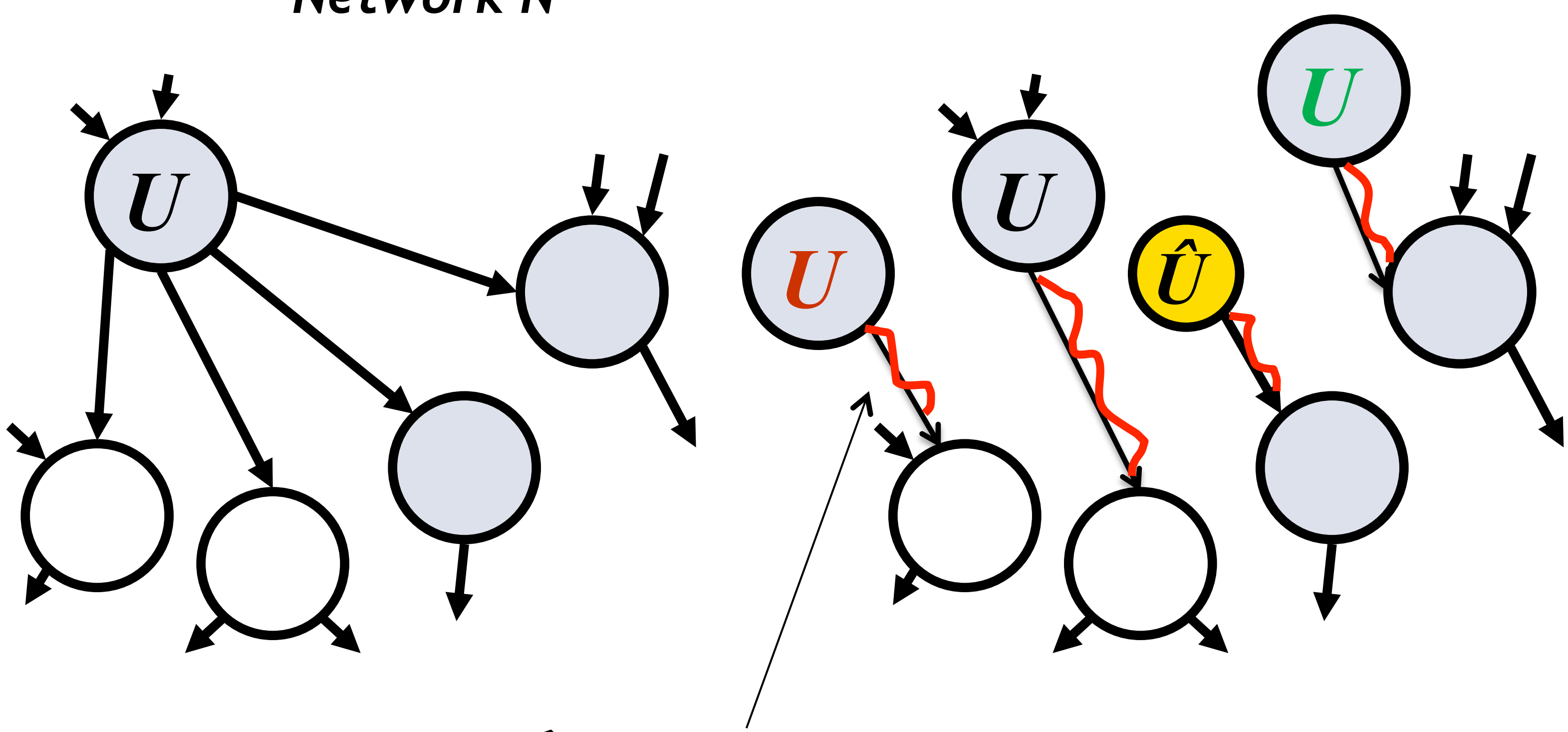


Re-parameterization: Duplicating a Node for Each Arc/Function

Variables in different buckets are renamed and duplicated
(Globerson and Jakkola, 2008)

Before Splitting:
Network N

After Splitting:
Network N'



Reparameterize by cost shifting,
optimally by linear programming

The Principle of Cost-Shifting

$$C^* = \max_{\mathbf{X}} \sum_{(ij)\in F} f_{ij}(X_i, X_j) \leq \sum_{(ij)\in F} \max_{\mathbf{X}} f_{ij}(X_i, X_j),$$

Introduce a collection of functions:

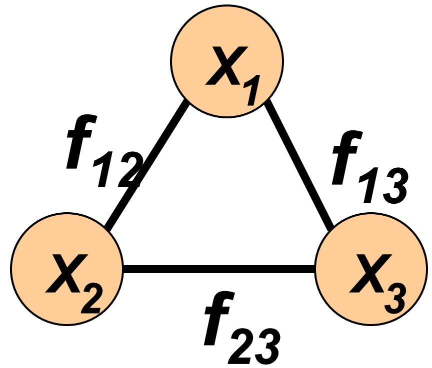
$$\lambda \in \Lambda \quad \Leftrightarrow \quad \forall i, \quad \sum_j \lambda_{ij}(X_i) = 0$$

Then, we have

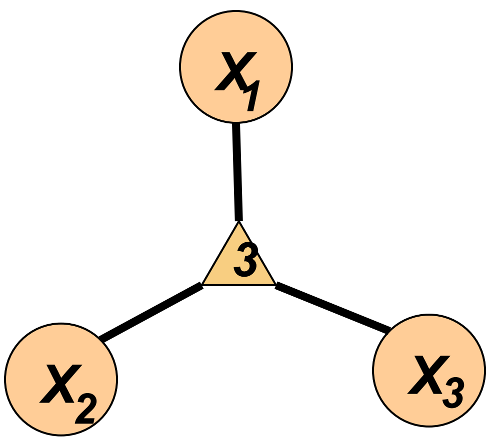
$$\begin{aligned} C^* &= \max_{\mathbf{X}} \sum_{(ij)\in F} f_{ij}(X_i, X_j) \\ &= \max_{\mathbf{X}} \sum_{(ij)\in F} f_{ij}(X_i, X_j) + \sum_i \sum_j \lambda_{ij}(X_i) \qquad (3) \\ &\leq \min_{\lambda\in\Lambda} \sum_{(ij)\in F} \max_{\mathbf{X}} \left(f_{ij}(X_i, X_j) + \lambda_{ij}(X_i) + \lambda_{ji}(X_j) \right) \end{aligned}$$

Bounding by Full Decomposition

Original problem:

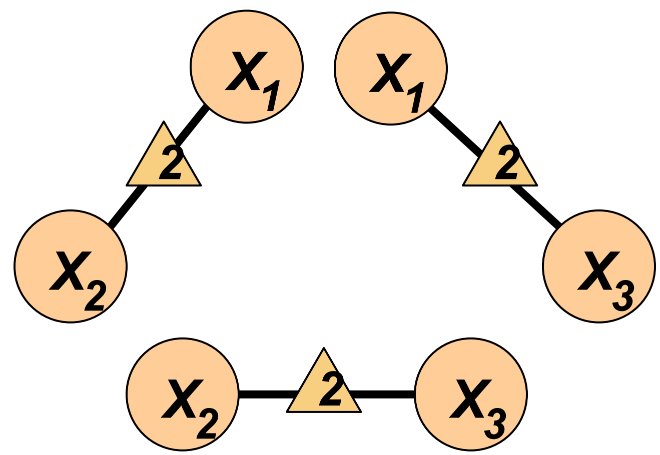


Exact solution:



$\leq$

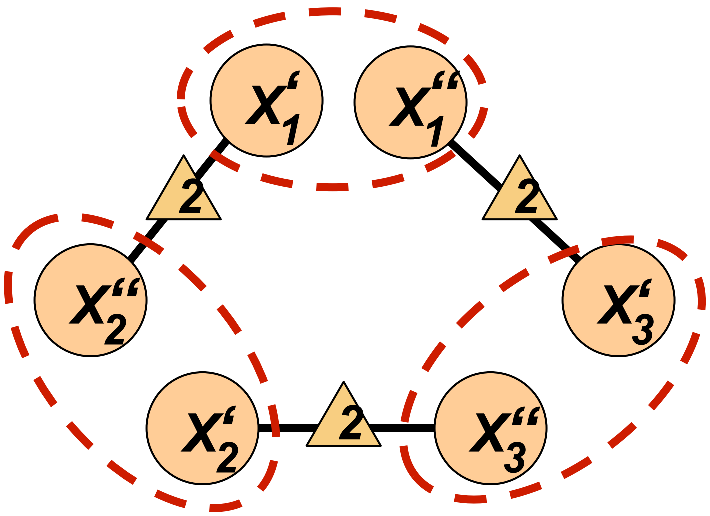
Upper bound:



$\max(f_{12}+f_{13}+f_{23})$

$\max f_{12}+\max f_{13}+\max f_{23}$

Tightening the upper bound



maximize each factor independently

$$\max f_{12} + \max f_{13} + \max f_{23}$$

subject to $X'_1 = X''_1,\ X'_2 = X''_2,\ X'_3 = X''_3$

Introduce functions
$\lambda_{ij}(X_i), \lambda_{ji}(X_j)$ *for each edge (ij)*
whose sum is 0

minimize upper bound:

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

re-parameterization

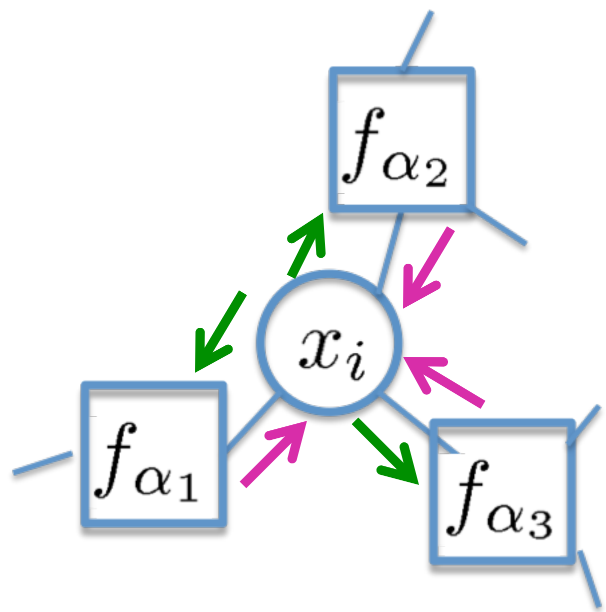
LP-tightening: use message passing, based on coordinate descent or gradient, sub-gradient approaches.

λ's   *Lagrangian multipliers*

Factor graph Linear Programming

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

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

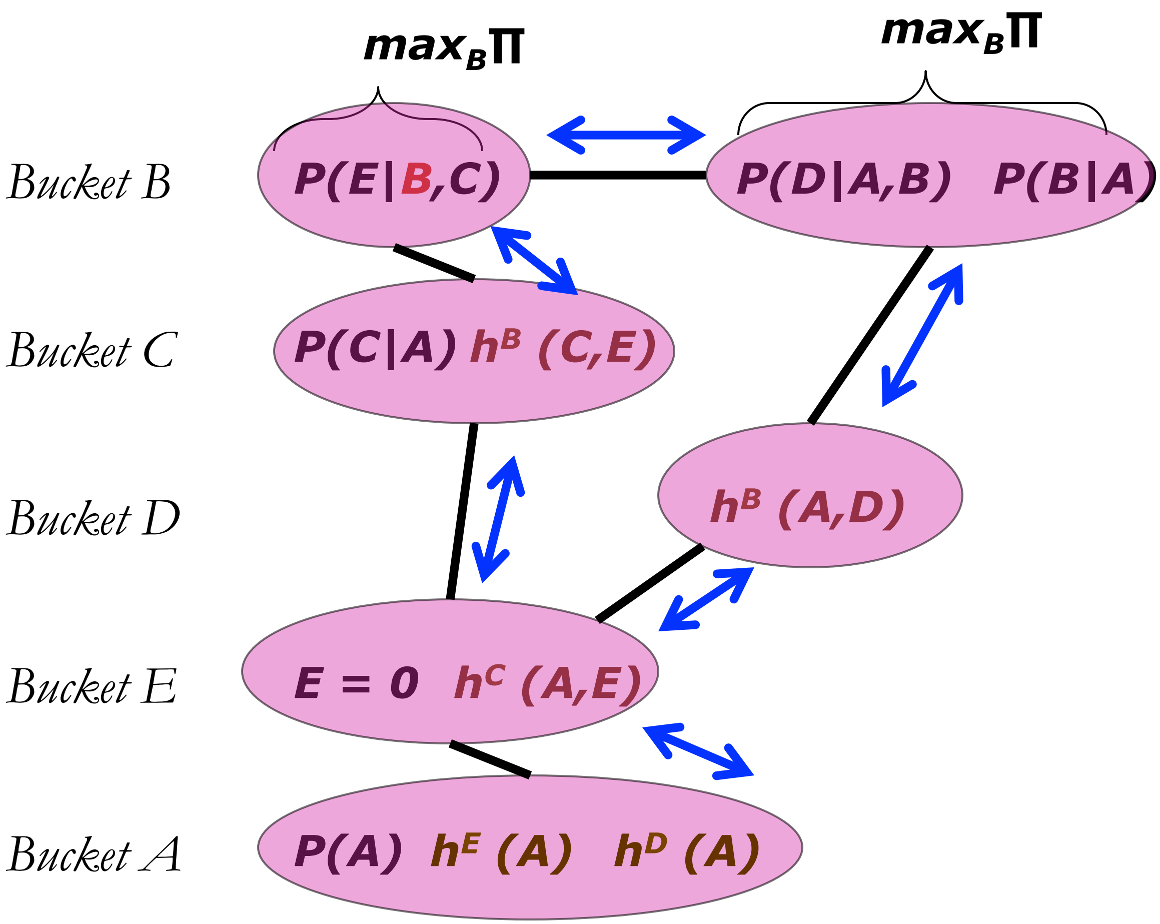


Combining Mini-buckets with cost-shifting

- Use mini-bucket as structuring a join-graph with a given i-bound: JGLP(i)

- Do mini-bucket with cost-shifting in each bucket: MBE-MM(i)

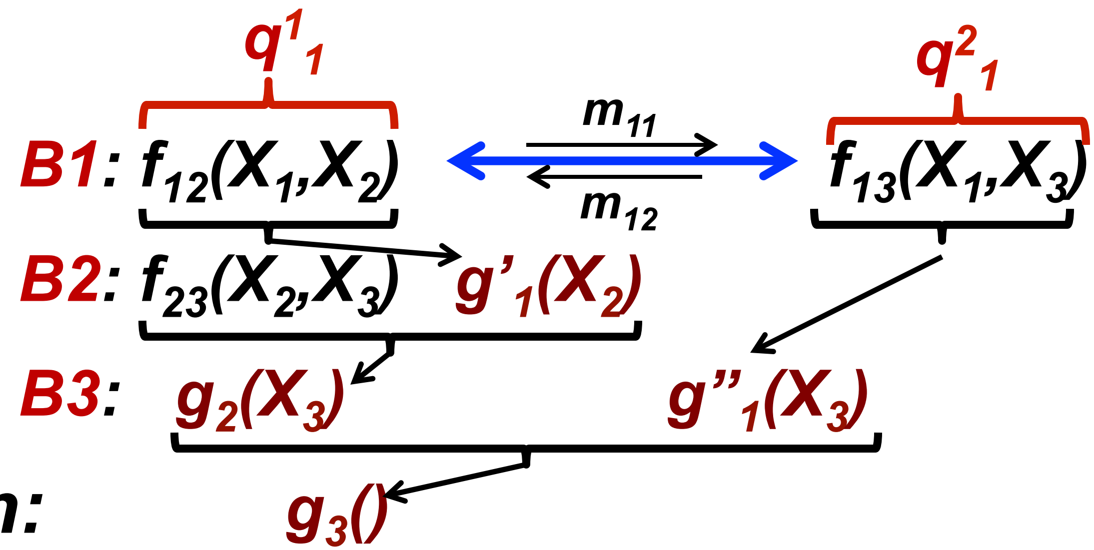
Join Graph Linear Programming (JGLP)



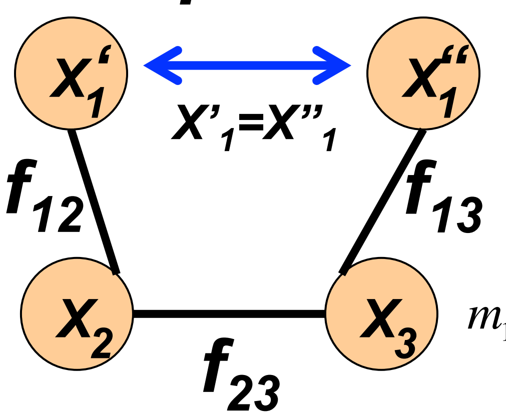
MB defines
A Join Graph

Mini-bucket elimination with moment-matching

Buckets and messages:



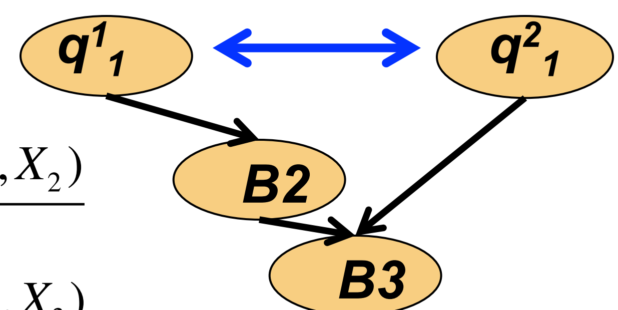
Interpretation:



$$m_{11} = \frac{\max_{X_3} f_{13}(X_1, X_3) - \max_{X_2} f_{12}(X_1, X_2)}{2}$$

$$m_{12} = \frac{\max_{X_2} f_{12}(X_1, X_2) - \max_{X_3} f_{13}(X_1, X_3)}{2}$$

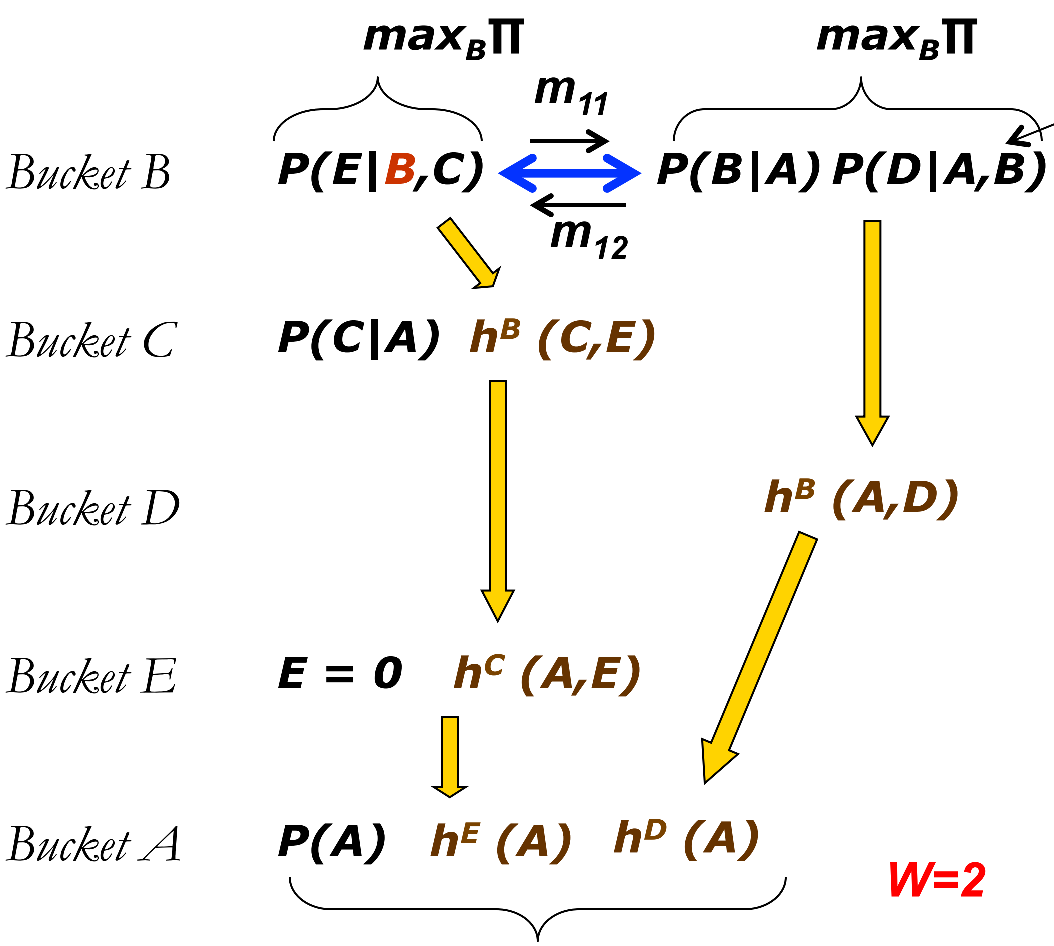
Bucket junction tree:

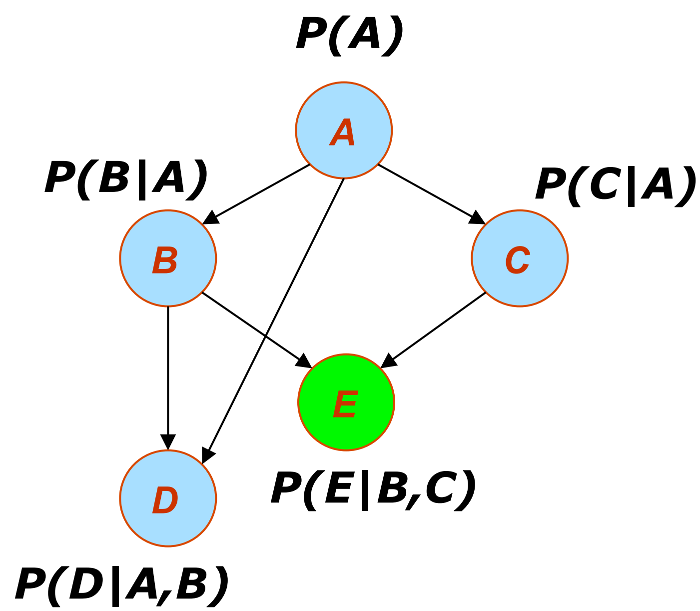


Mini-Bucket Elimination

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

Node duplication, renaming





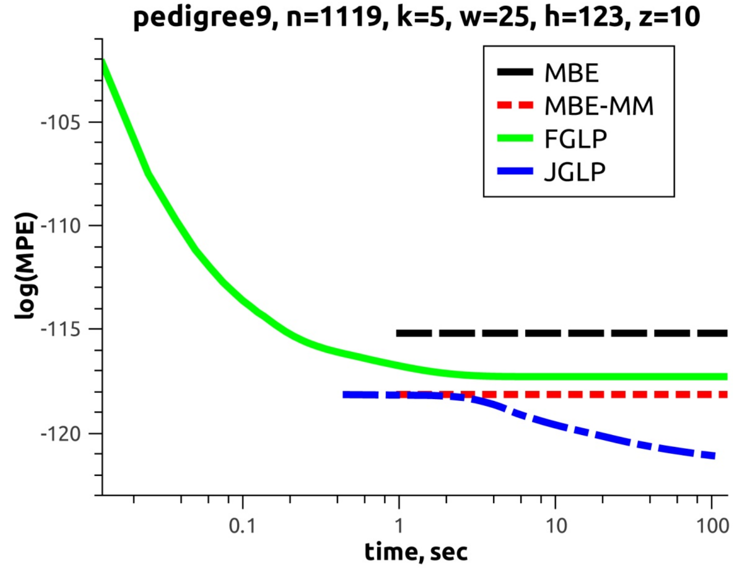
MPE* is an upper bound on MPE --U
Generating a solution yields a lower bound--L

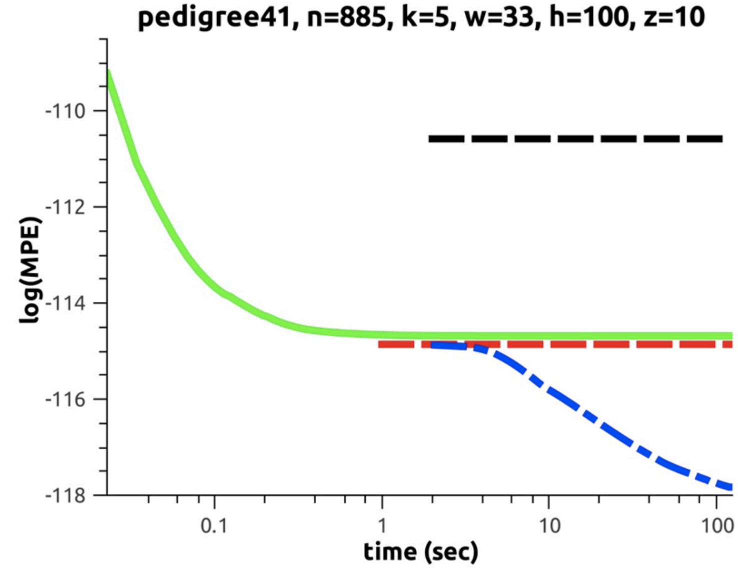
Iterative tightening as bounding schemes

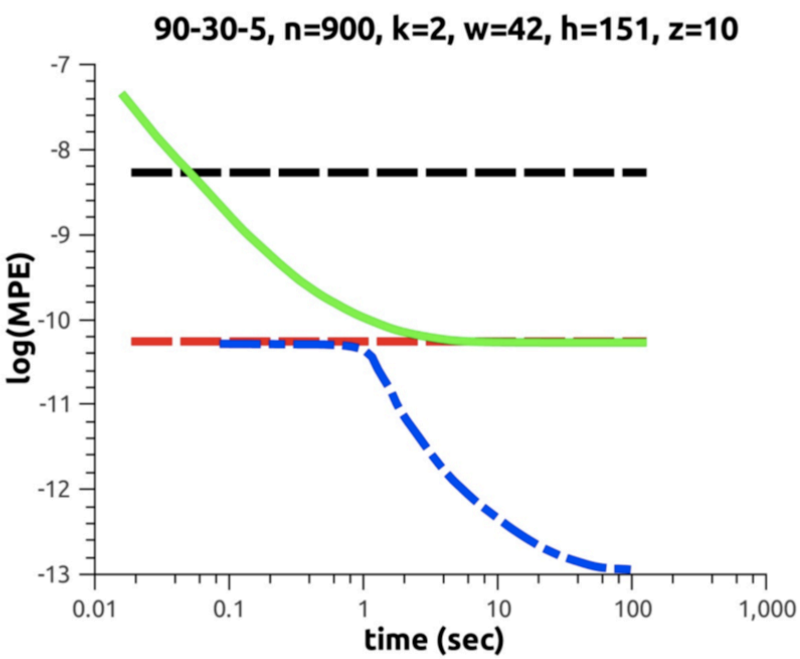
- 4 schemes: MBE, MBE-MM, FGLP, JGLP
  - Iterative schemes (FGLP, JGLP) ran for 5, 300, 3600 seconds

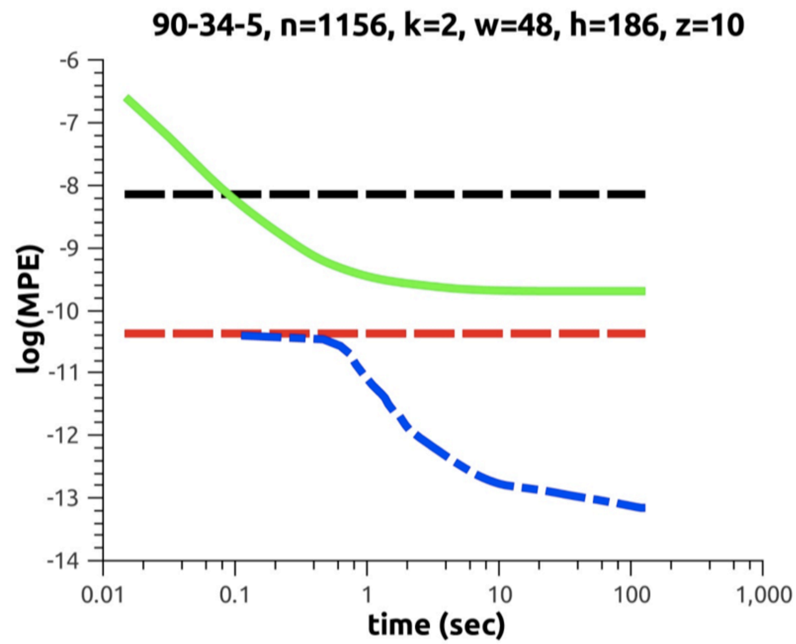
instance name	n	k	w	z	MBE	MBE-MM	FGLP time cut-offs			JGLP time cut-offs		
							5	300	3600	5	300	3600
					UB/time	UB/time	UB	UB	UB	UB	UB	UB
75-25-5	625	2	34	10 20	-15.4553/1 -17.4417/4	-18.4089/1 -20.0576/4	-16.6853	-16.6854	-16.6854	-20.0289 -20.0576	-20.8364 -20.1278	-20.8364 -20.7067
90-30-5	900	2	42	10 20	-8.2481/1 -9.7424/7	-10.2597/1 -11.6004/7	-10.2450	-10.2705	-10.2705	-11.8469 -11.6004	-12.9594 -11.6942	-13.015 -12.5259
90-34-5	1156	2	48	10 20	-8.42007/1 -9.58332/8	-10.3708/1 -12.3670/9	-9.65003	-9.69458	-9.69458	-12.3469 -12.3670	-13.2262 -12.5621	-13.2883 -13.1538
90-42-5	1764	2	60	10 20	-12.7401/1 -14.6136/13	-15.9680/1 -18.5487/14	-15.2480	-15.3653	-15.3653	-18.4100 -18.5487	-20.7714 -18.7679	-20.8136 -19.9705
largeFam4_11_51	1002	4	40	10 20	[illegible] OOM	[illegible] OOM	-201.582	-201.673	-201.673	[illegible] OOM	[illegible] OOM	[illegible] OOM
largeFam4_11_55	1114	4	38	10 20	-229.43/1 OOM	-242.489/1 OOM	-226.075	-226.328	-226.328	-242.657 OOM	-249.551 OOM	-250.453 OOM
largeFam4_12_51	1461	4	56	10 20	-218.229/2 OOM	-239.896/3 OOM	-217.564	-217.740	-217.740	-239.896 OOM	-245.900 OOM	-253.153 OOM
pedigree7	867	4	32	10 20	-105.854/1 -108.011/33	-109.569/1 -111.120/42	-110.179	-110.187	-110.187	-109.960 OOM	-110.810 OOM	-111.293 OOM
pedigree13	888	3	32	10 20	-69.0973/1 -69.8890/8	-70.0999/1 -71.1071/11	-71.8561	-71.8591	-71.8591	-70.4581 -71.1071	-71.9869 -71.1071	-72.0374 -71.3658
pedigree31	1006	5	30	10 20	-125.032/1 OOM	-126.629/1 OOM	-126.667	-126.678	-126.678	-126.644 OOM	-129.158 OOM	-129.277 OOM
pedigree41	885	5	33	10 20	-110.156/1 -112.153/29	-114.858/1 -117.638/37	-114.681	-114.681	-114.681	-115.050 OOM	-118.133 OOM	-118.419 OOM
type4_120_17	4302	5	23	10 20	-1128.22/1 -1235.94/18	-1203.08/1 -1237.95/21	-1049.34	-1049.85	-1049.86	-1203.21 -1237.95	-1221.21 OOM	-1223.69 OOM
type4_170_23	6933	5	21	10 20	-1682.9/1 -1783.18/7	-1747.18/1 -1783.76/7	-1509.96	-1511.61	-1511.65	-1747.22 -1783.76	-1769.96 -1783.76	-1772.16 -1783.76

Iterative tightening as bounding schemes





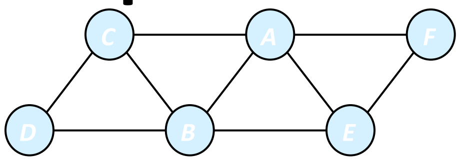




Outline

- Graphical models: reasoning principles
- Inference
- Advancing Search via AND/OR Search
- Lower Bounding schemes for inference
- Lower-bounding heuristic for AND/OR search
- Experiments

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

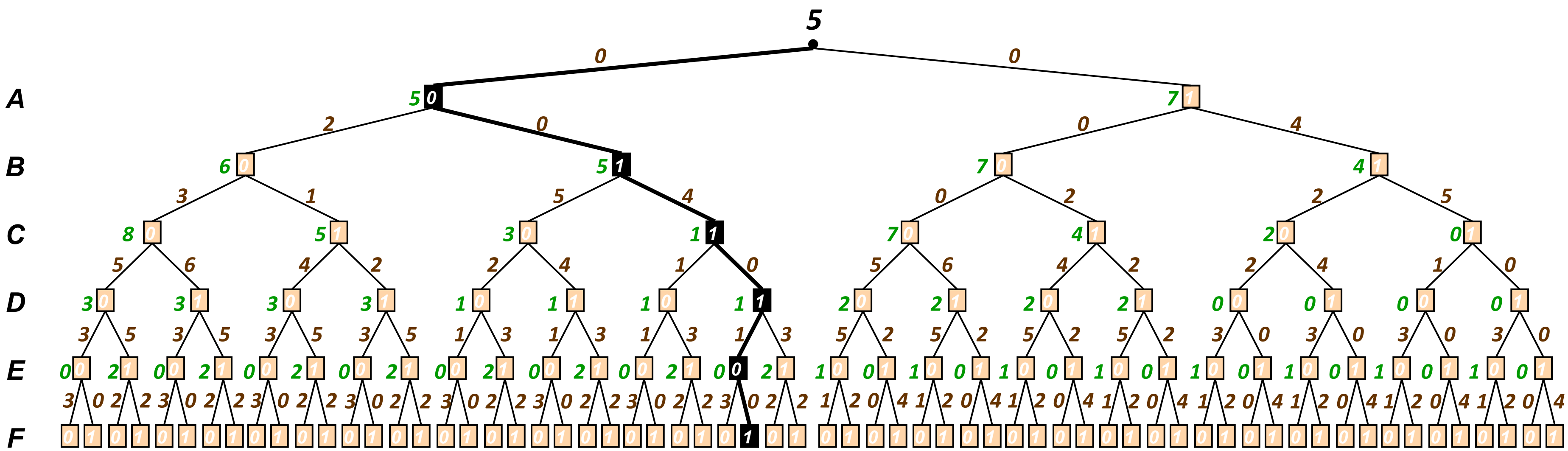
B	E	f_7
0	0	3
0	1	2
1	0	1
1	1	0

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

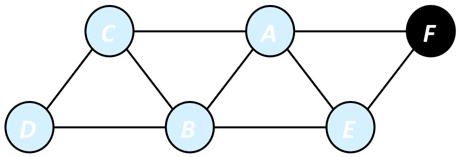
$$f(\mathrm{X}) = \sum_{i=1}^{9} f_i(\mathrm{X})$$

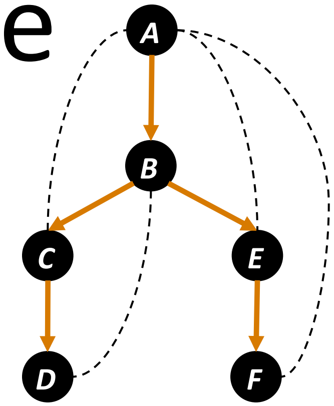
$$\min_{a,b,c,d,e,f} f_1(a,b) + f_2(a,c) + f_3(a,f) + f_4(b,c) + f_5(b,d) + f_6(b,e) + f_7(c,d) + f_8(e,f)$$



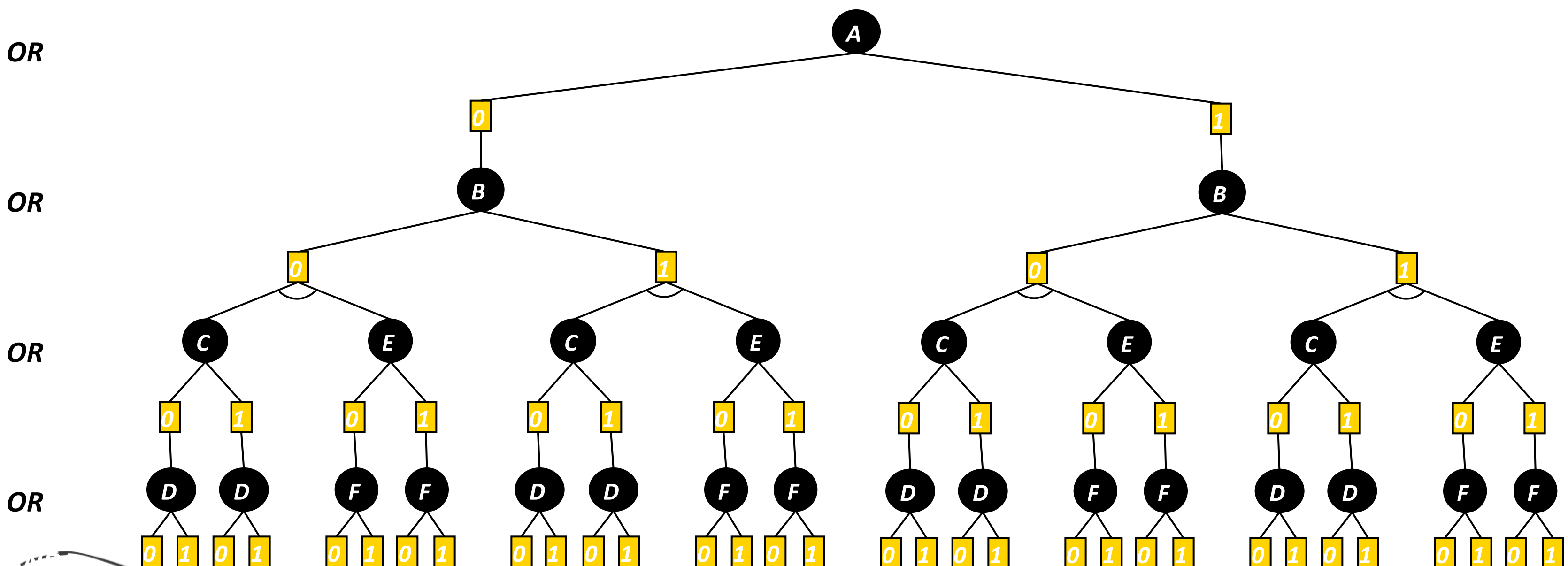
An optimal assignment is A=0, B=1, C=1, D=1, E=0, F=1 with cost 5

The AND/OR Search Tree



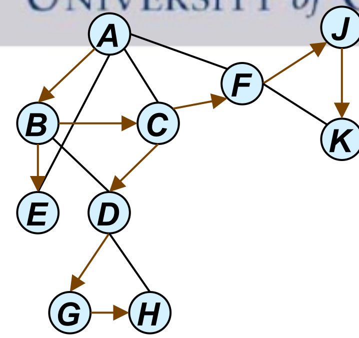


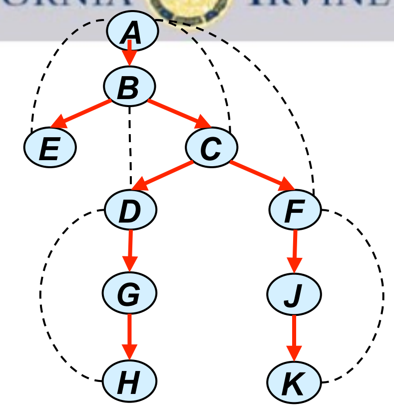
Pseudo tree

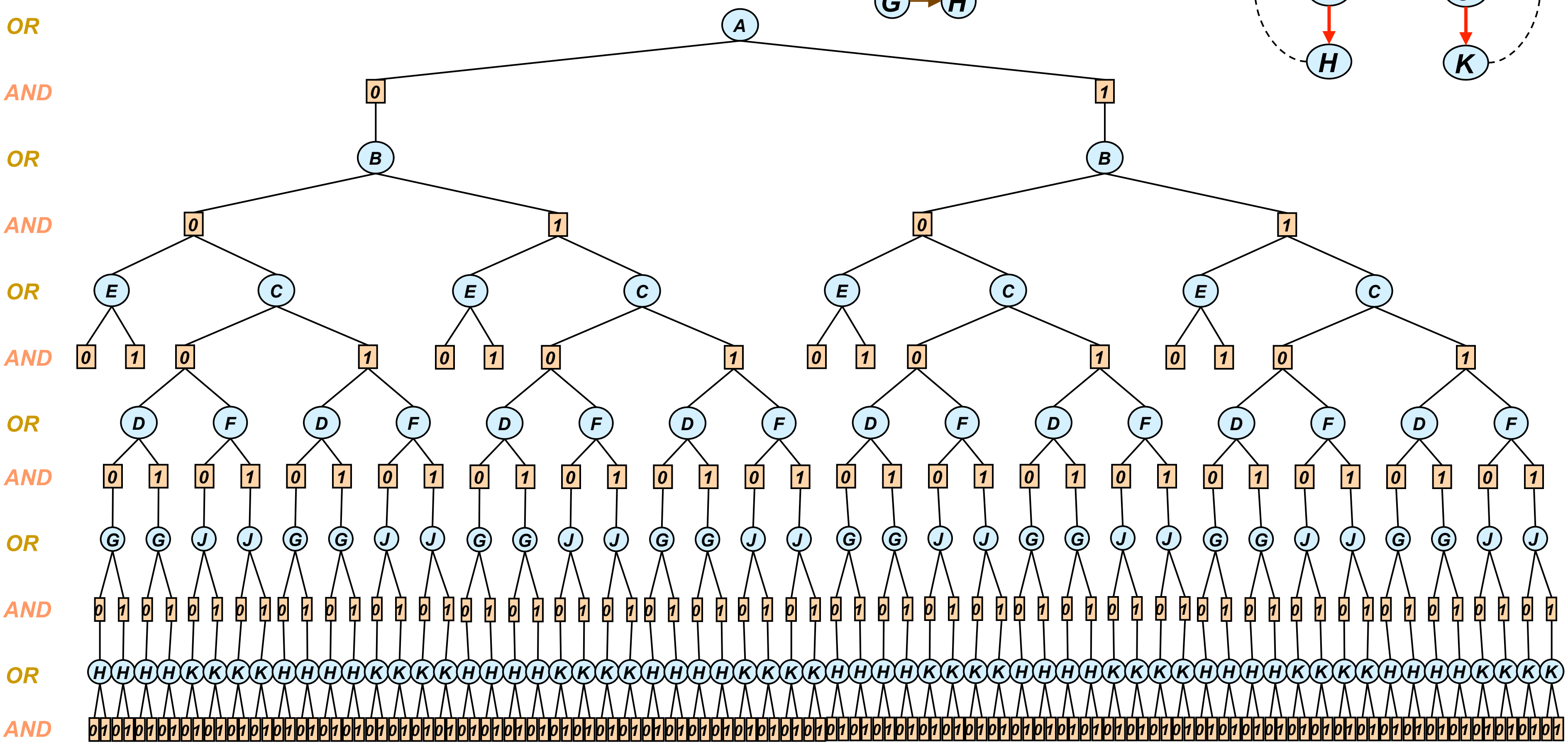


A solution subtree is

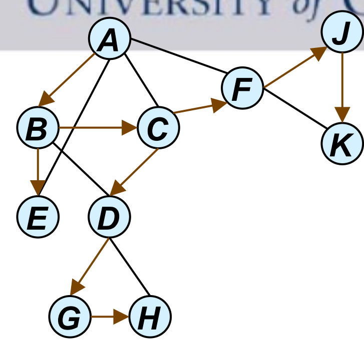
From AND/OR Tree

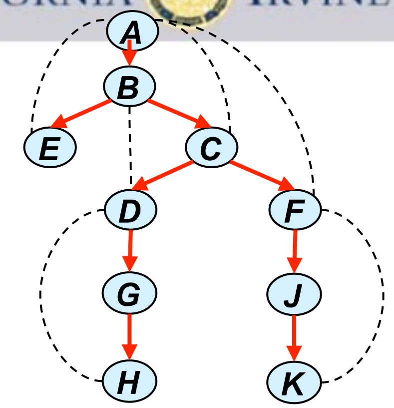


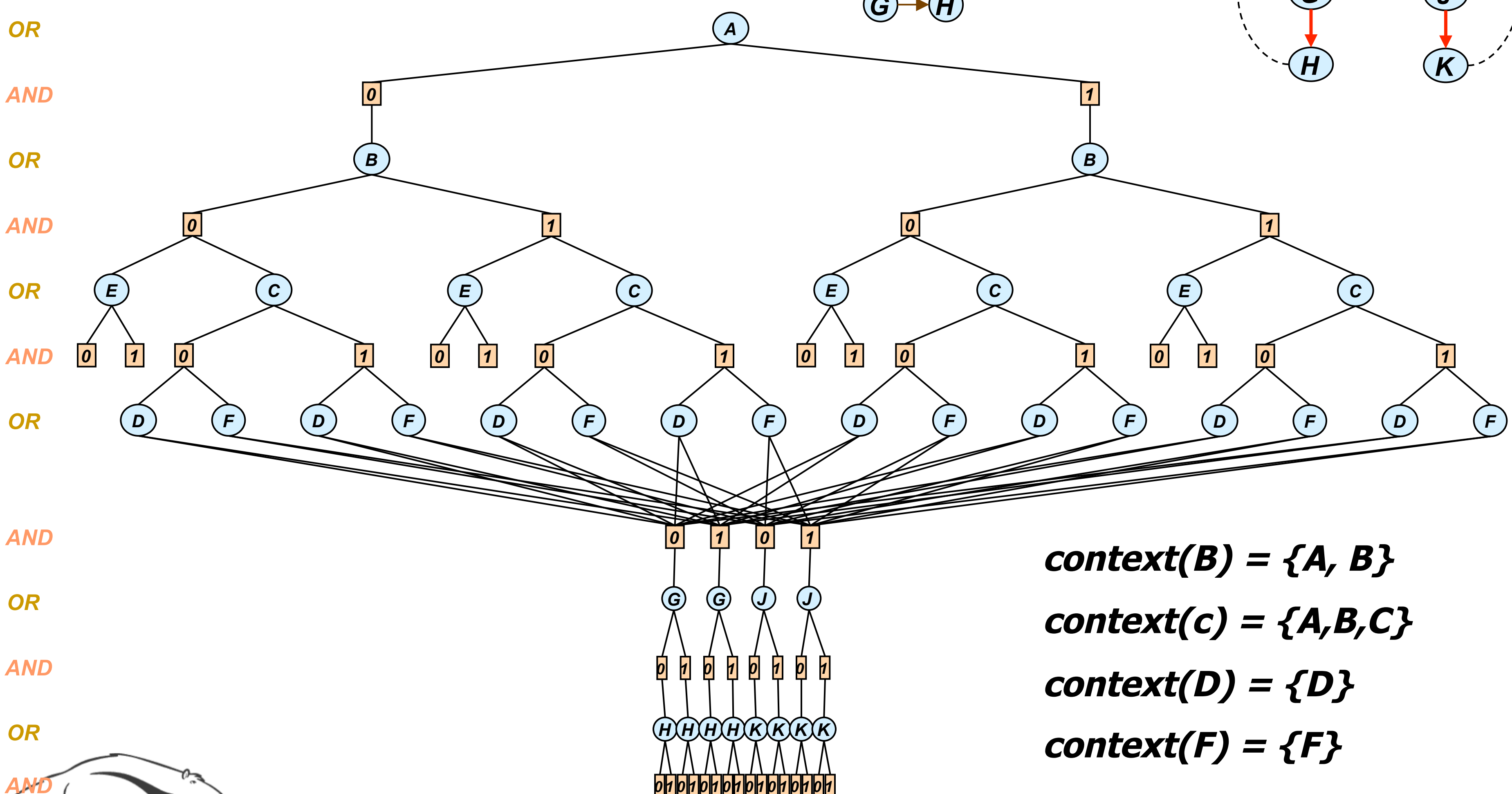




An AND/OR Graph







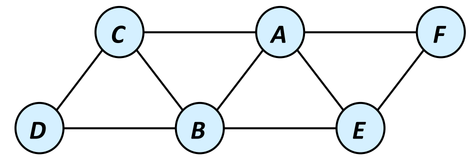
context(B) = {A, B}

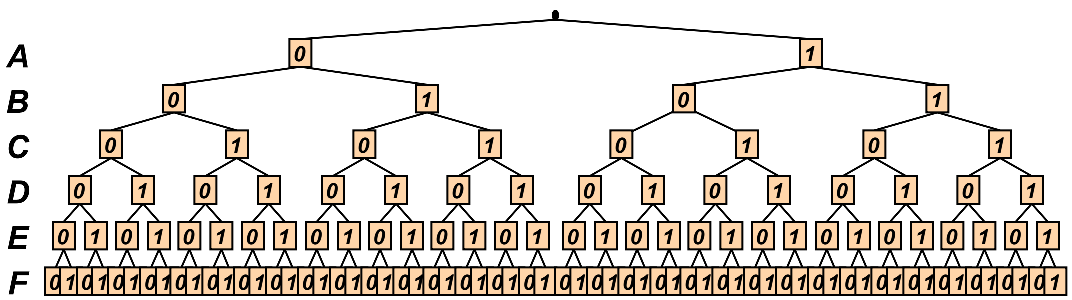
context(c) = {A,B,C}

context(D) = {D}

context(F) = {F}

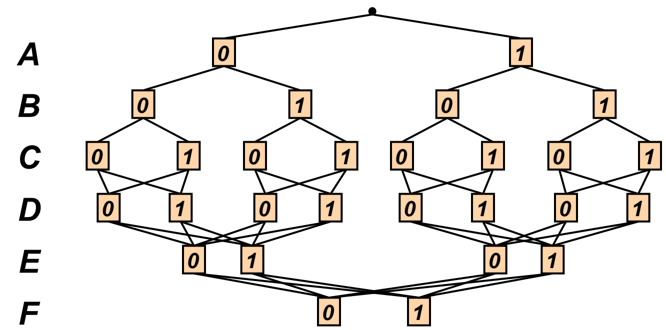
All Four Search Spaces





Full OR search tree

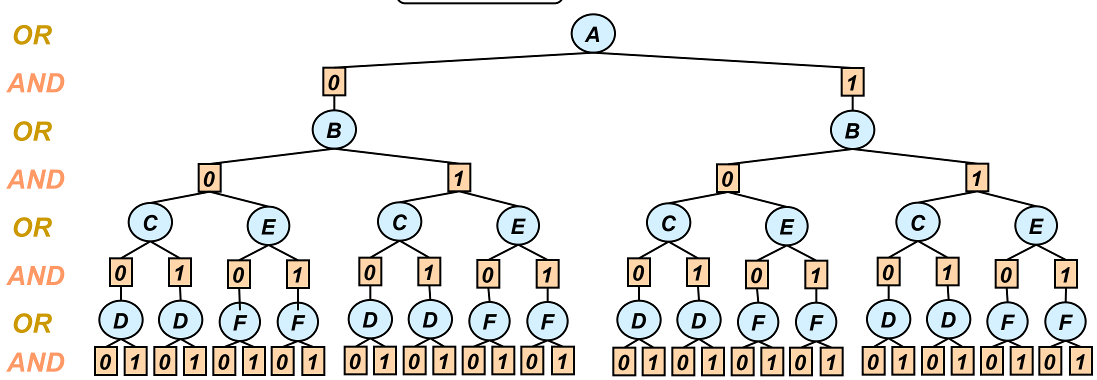
126 nodes



Context minimal OR search graph

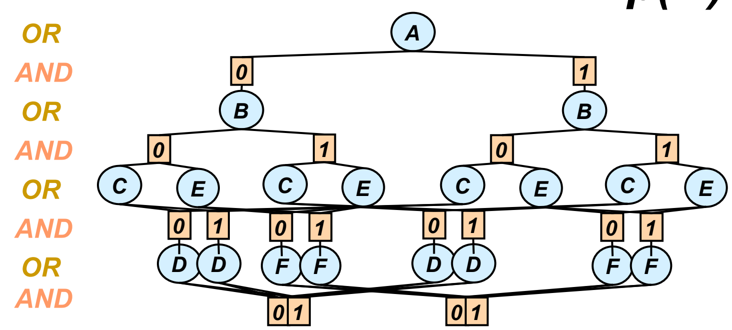
28 nodes

Exp(w)



Full AND/OR search tree

54 AND nodes



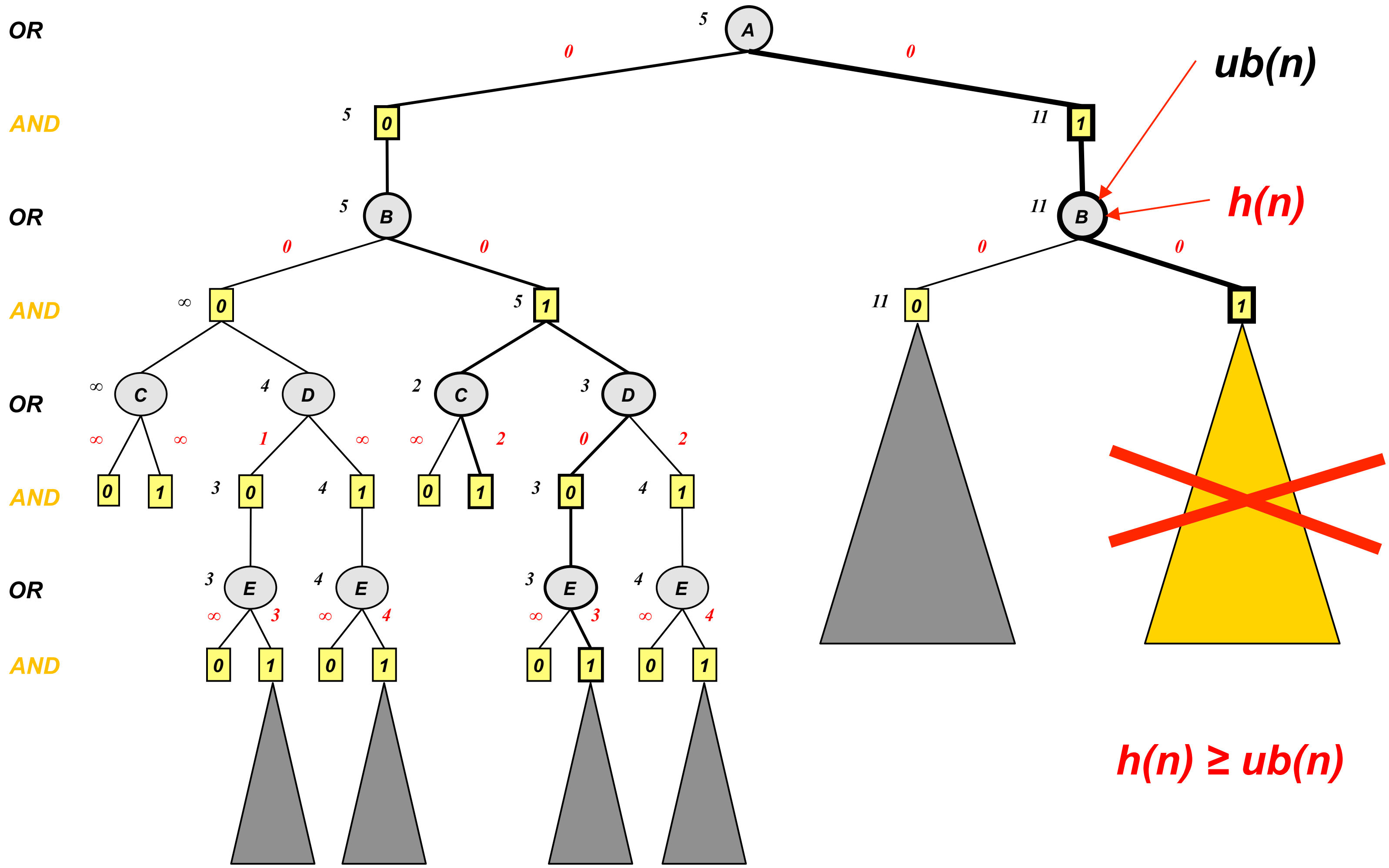
Context minimal AND/OR search graph

18 AND nodes

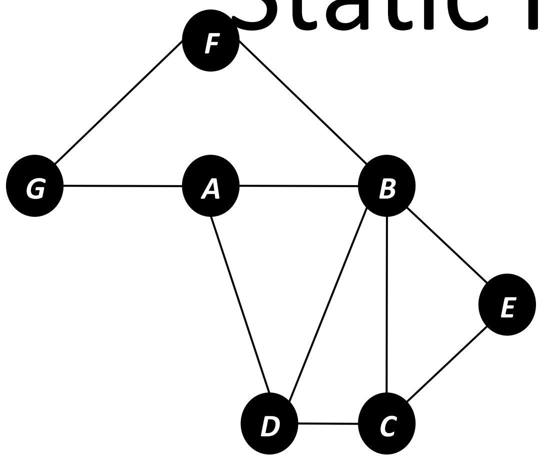
Any query is best computed
Over the c-minimal AO space

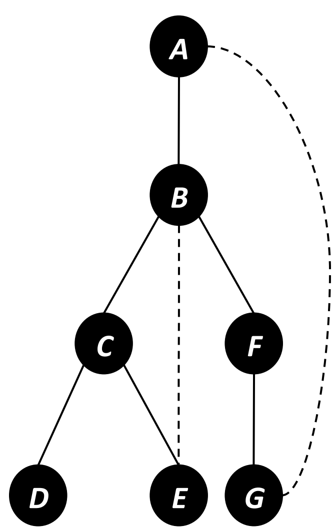
AND/OR Branch-and-Bound

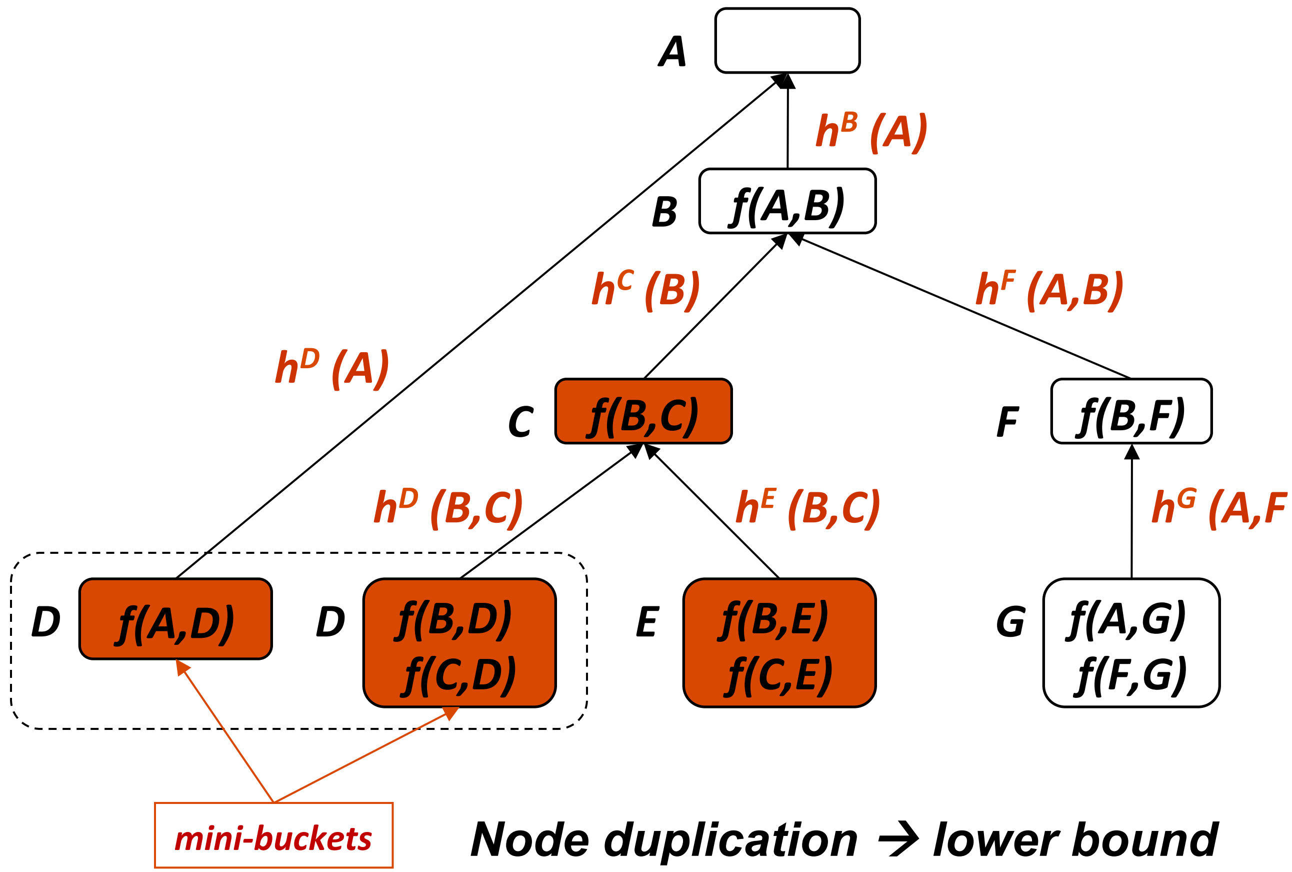
Kask and Dechter 2001
Marinescue and Dechter, 2005-2009



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)

Empirical evaluation

Experimental settings:

- **4 benchmarks:**
  - Pedigrees (10 instances)
  - Type4 (10 instances)
  - LargeFam (40 instances)
  - n-by-n grid networks (32 instances)

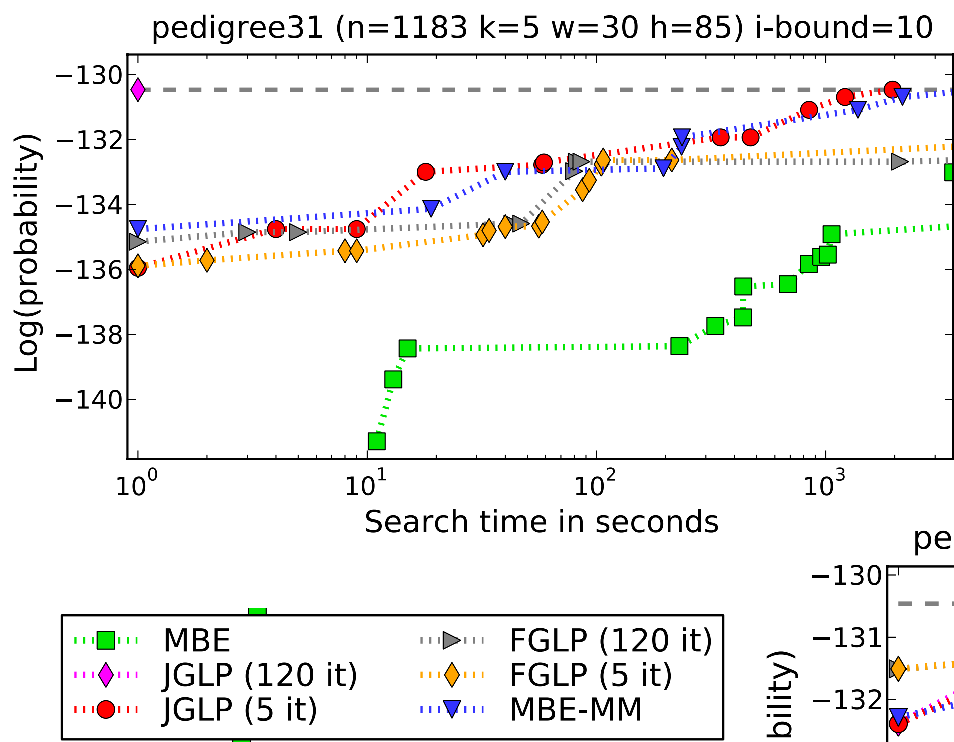
  genetic linkage analysis networks (Pedigrees, Type4, LargeFam)
- Algorithms as bounding schemes
- Algorithms as heuristic generators

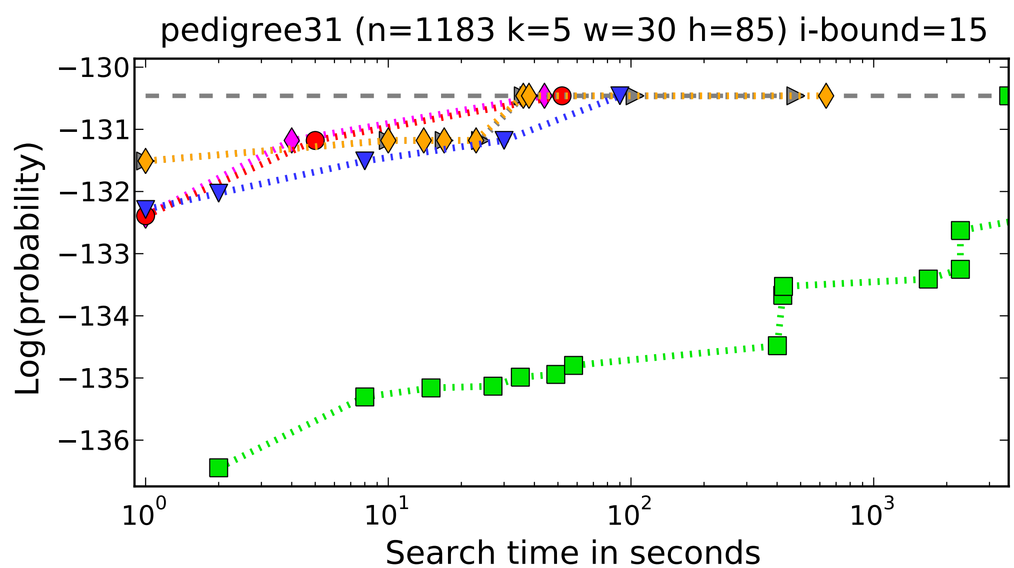
Iterative tightening as heuristic generators

- 4 schemes used:
  - AOBB guided by pure MBE heuristics (**AOBB-MBE**)
  - AOBB guided by MBE and max- marginal matching heuristics (**AOBB-MBE-MM**)
  - AOBB whose heuristics are generated from FGLP followed by MBE (**AOBB-FGLP+MBE**)
  - AOBB guided by JGLP-produced heuristics (**AOBB-JGLP**)
- FGLP, JGLP ran for 30 seconds
- Total search time bound 24 h
- Memory limit 3 Gb
- Mini-bucket z-bounds={10,15,20}

Empirical Evaluation: Haplotype problems

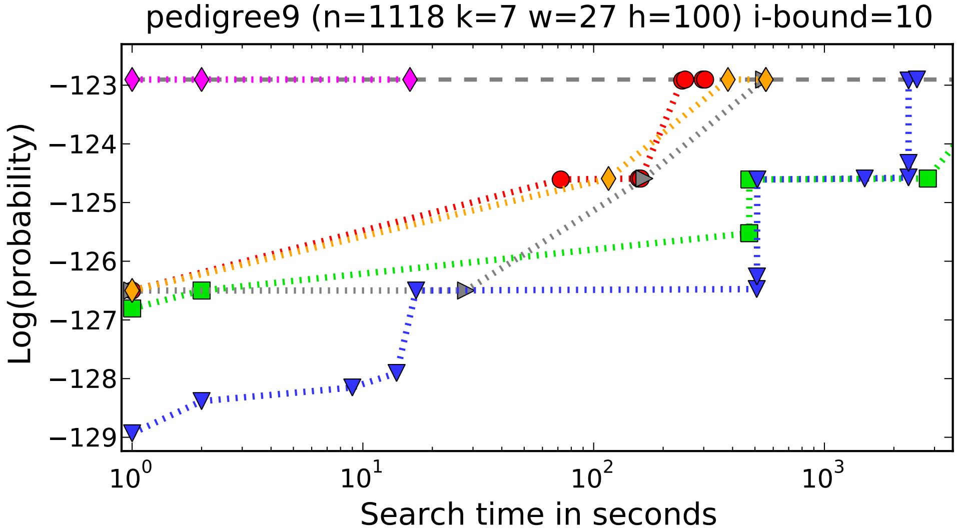
Time bound – 24 h

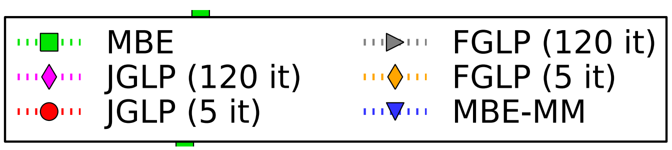


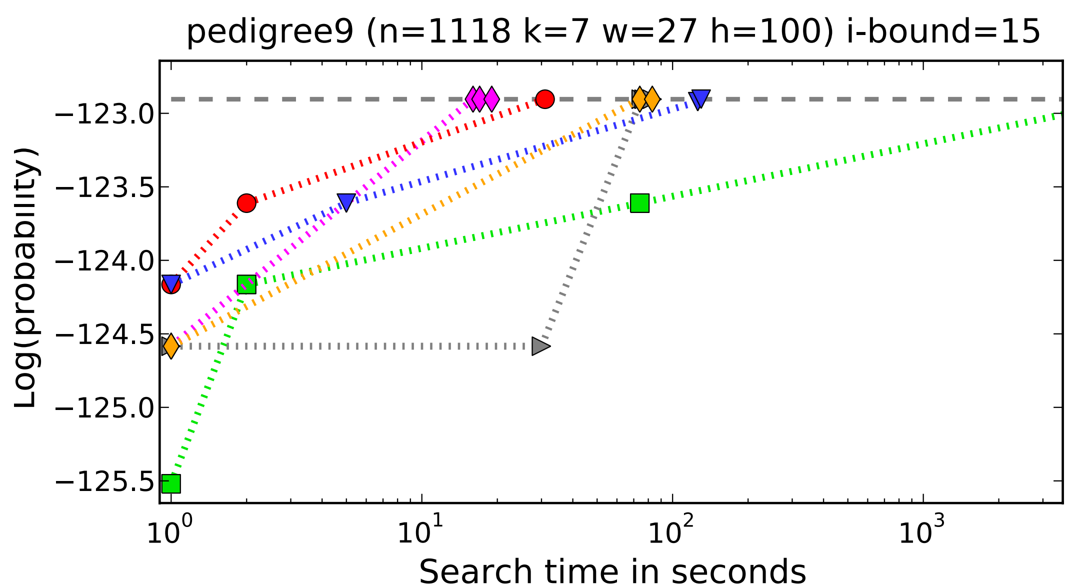


Empirical Evaluation: Haplotype problems

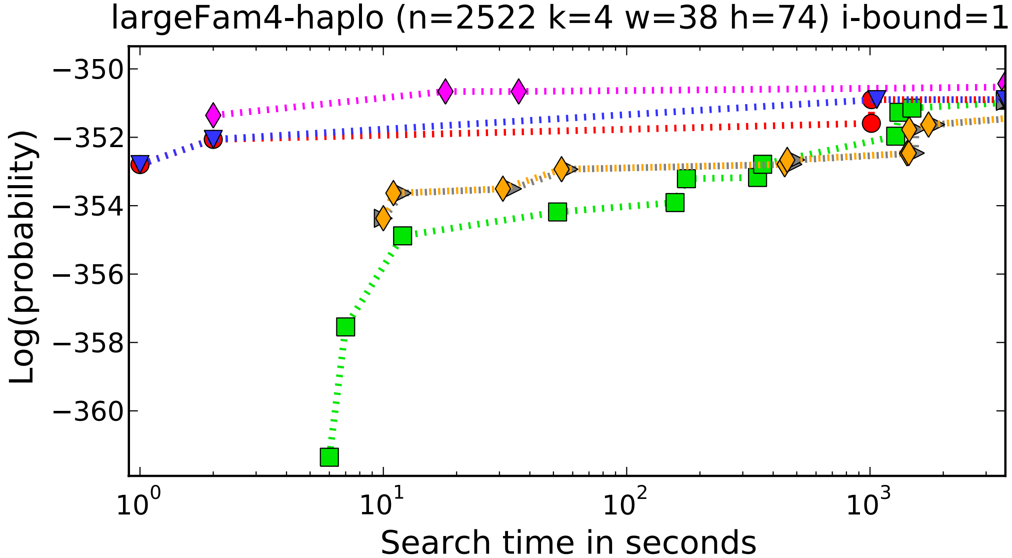
Time bound – 24 h



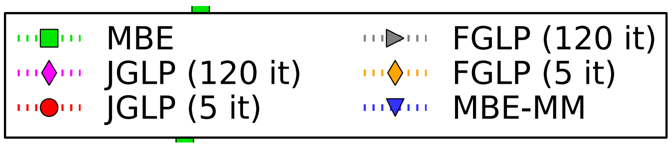


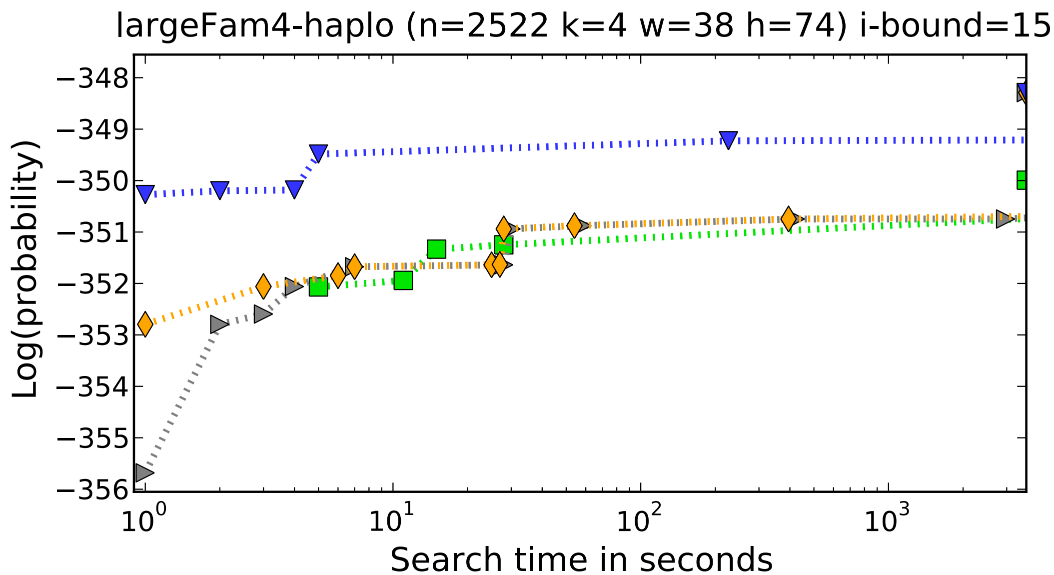


Empirical Evaluation: Large families

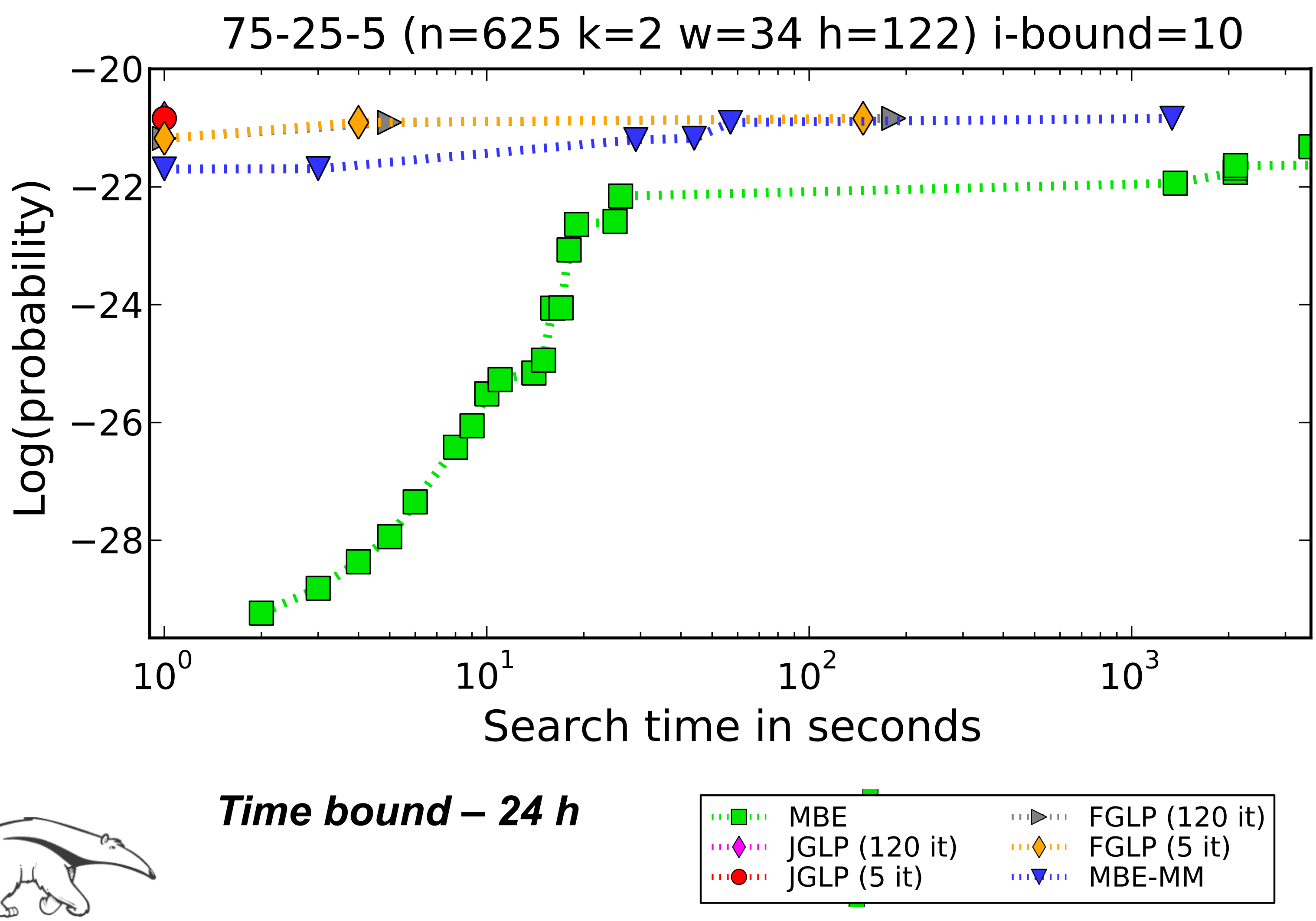


Time bound – 24 h



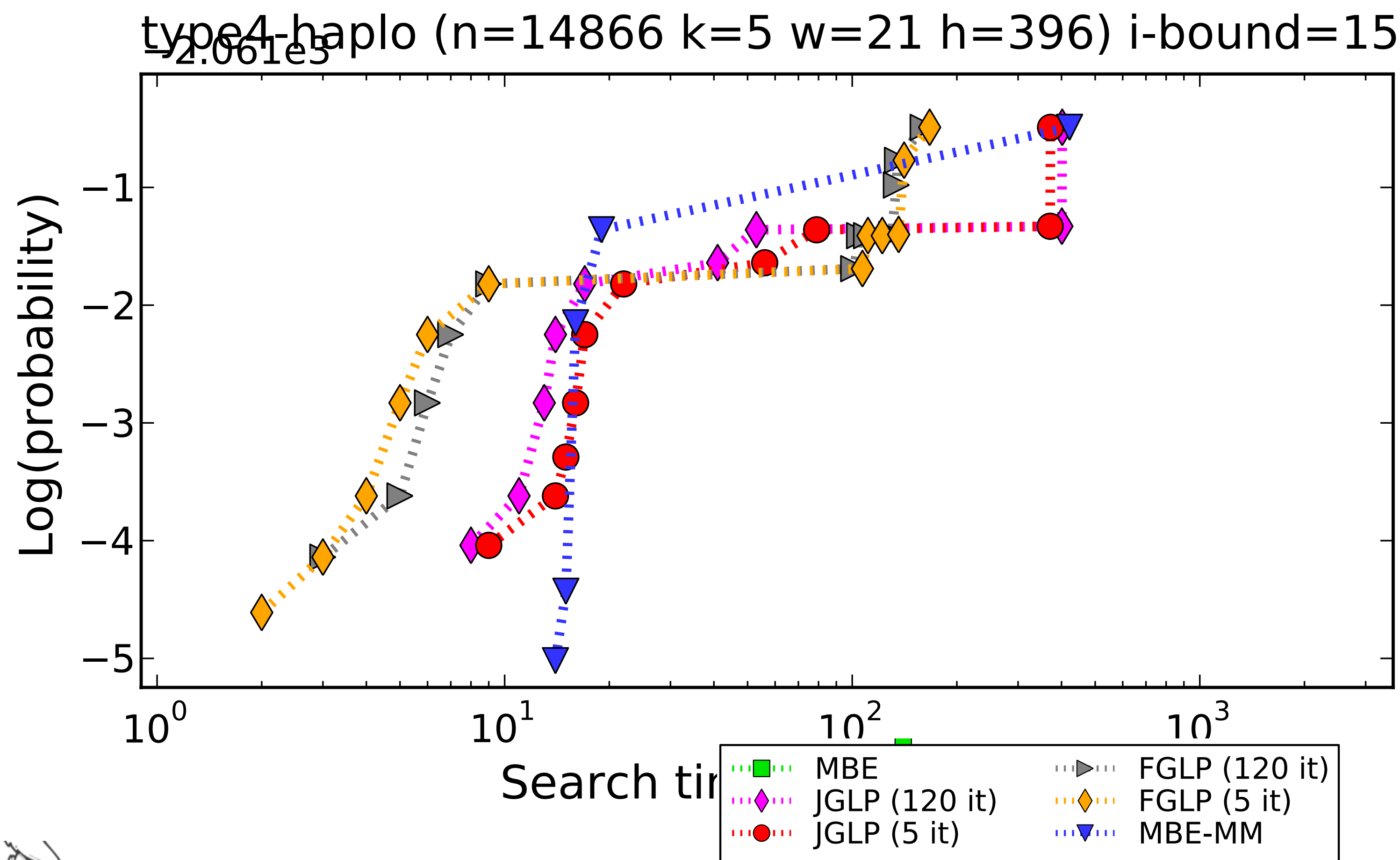


Empiricial Evaluation; Grid networks



Time bound – 24 h

Iterative tightening as heuristic generators



BREN:ICS INFORMATION AND COMPUTER SCIENCES

UNIVERSITY of CALIFORNIA IRVINE

PASCAL 2012 Inference Challenge

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

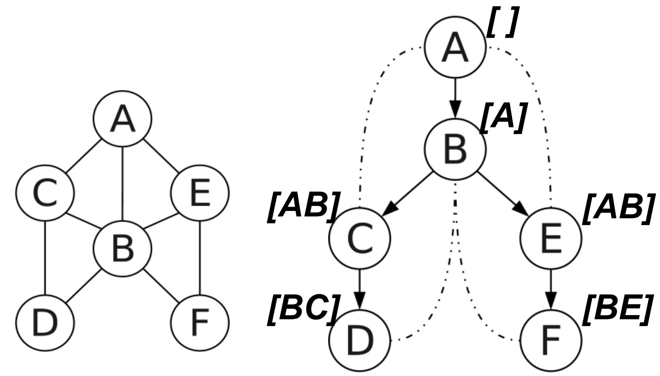
Lars Otten, Alexander Ihler,
Kalev Kask, Rina Dechter

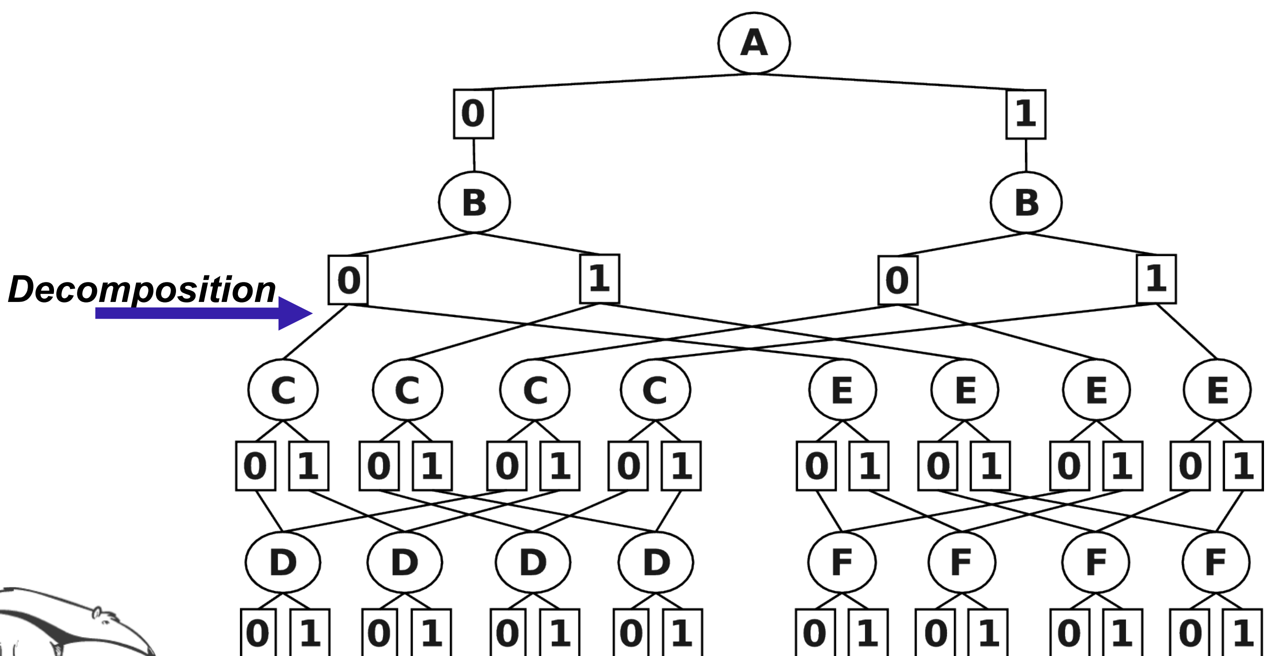
Dept. of Computer Science
University of California, Irvine

Harvard, 10/12

AND/OR Branch and Bound

- Guided by pseudo tree:
  - Subproblem decomposition.

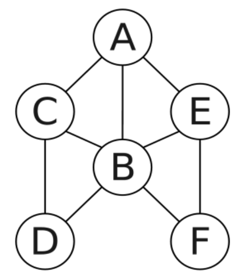


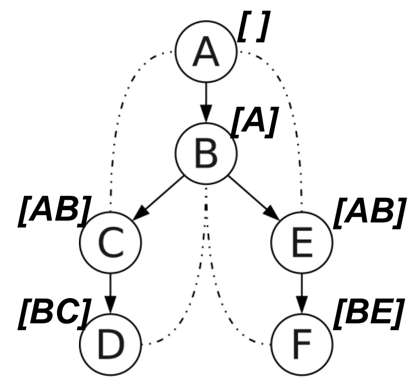


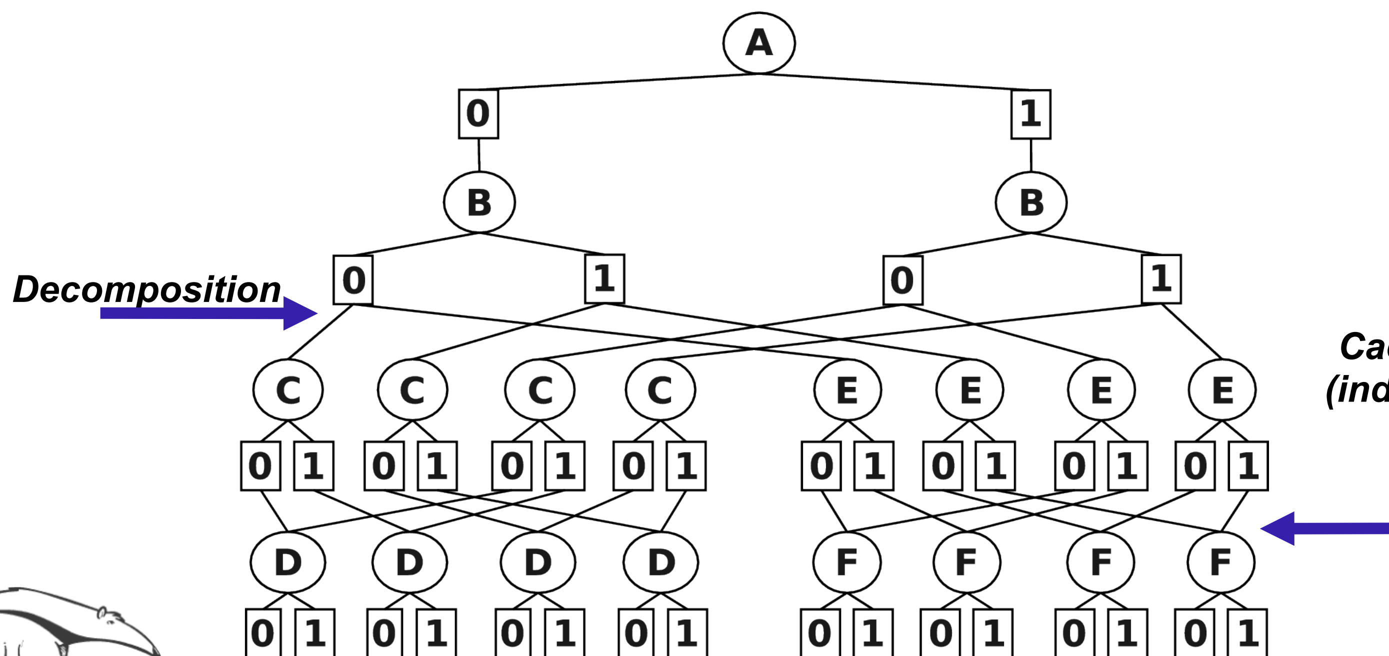
[Marinescu & Dechter, AIJ'09]

AND/OR Branch and Bound

- Guided by pseudo tree:
  - Subproblem decomposition.
  - Merge unifiable subproblems.







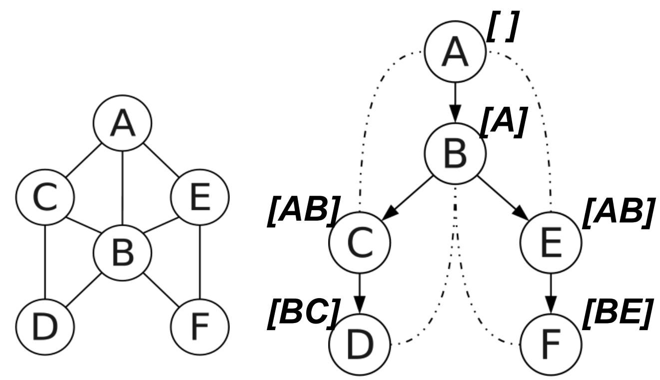
Cache table for F (independent of A)

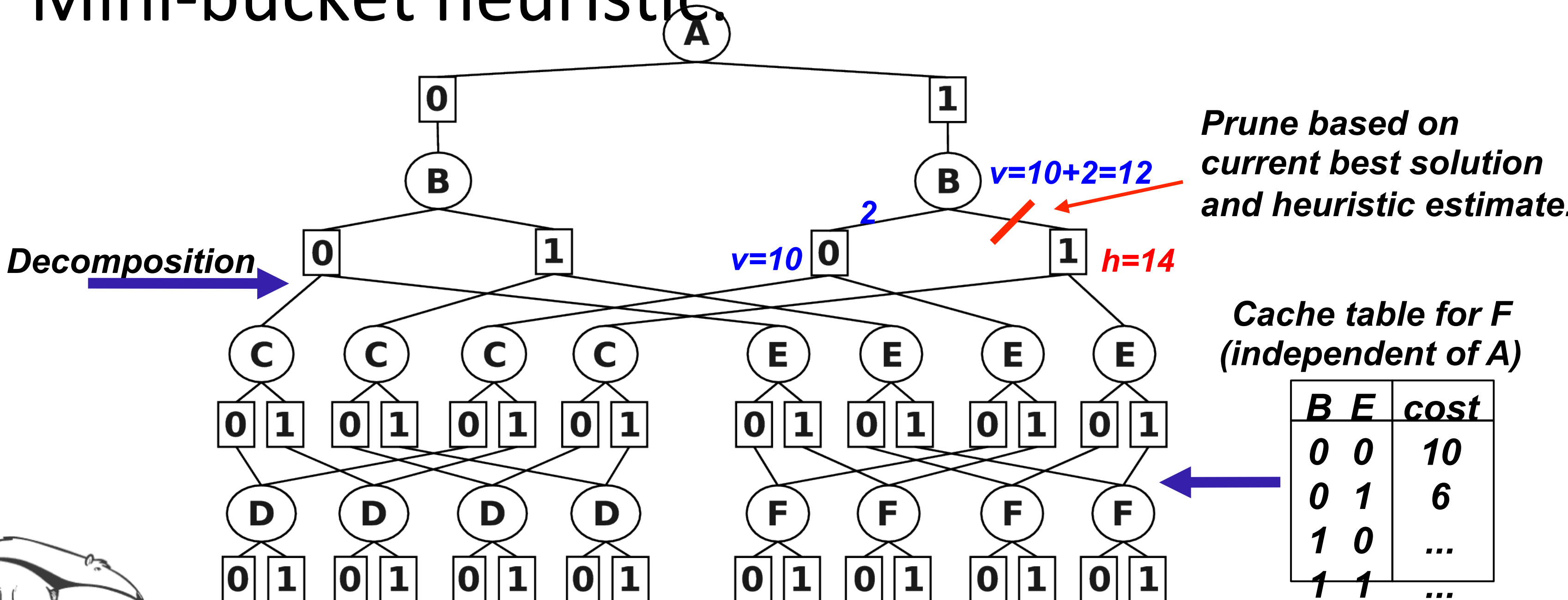
B	E	cost
0	0	10
0	1	6
1	0	...
1	1	...

[Marinescu & Dechter, AIJ'09]

AND/OR Branch and Bound

- Guided by pseudo tree:
  - Subproblem decomposition.
  - Merge unifiable subproblems.
- Mini-bucket heuristic.



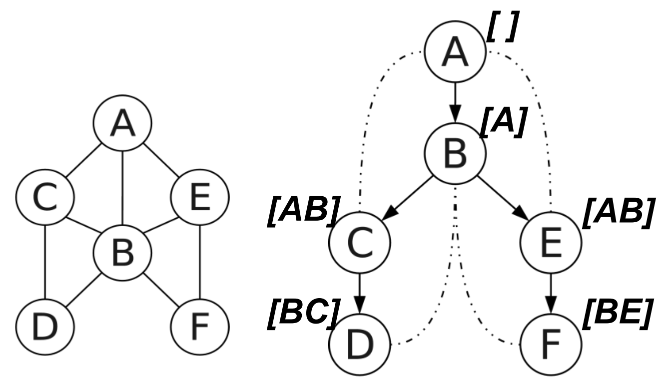


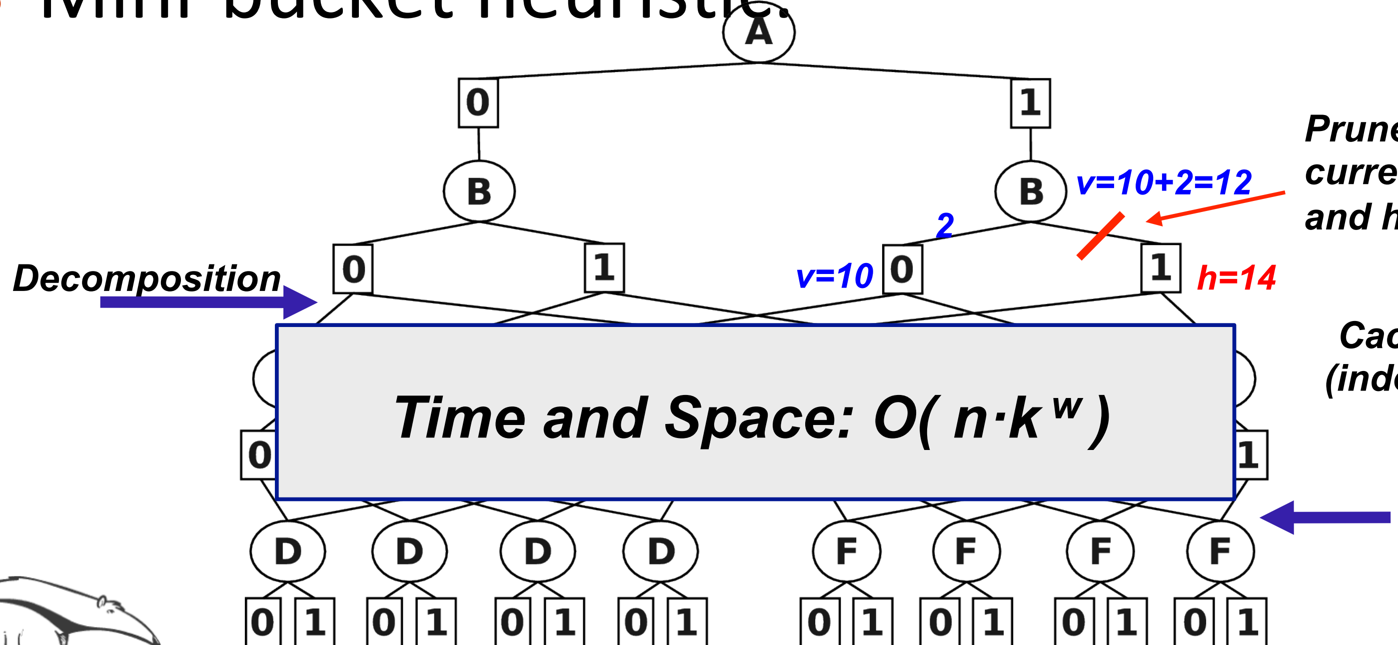
B	E	cost
0	0	10
0	1	6
1	0	...
1	1	...

[Marinescu & Dechter, AIJ'09]

AND/OR Branch and Bound

- Guided by pseudo tree:
  - Subproblem decomposition.
  - Merge unifiable subproblems.
- Mini-bucket heuristic.



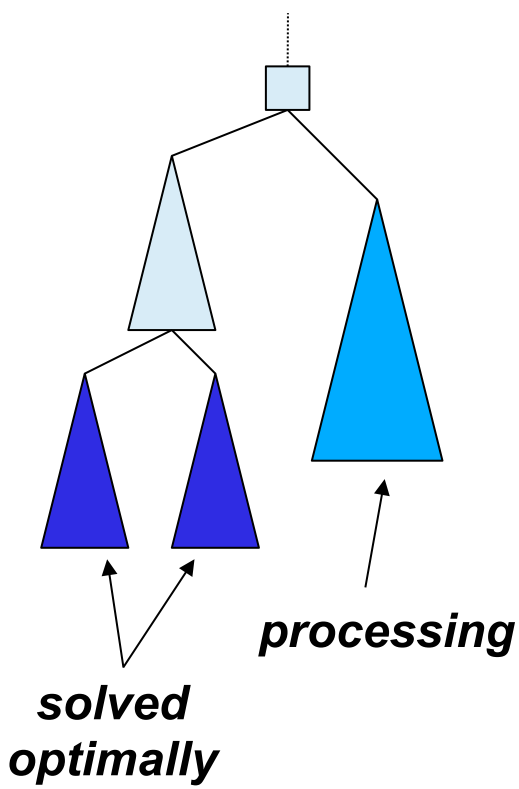


B	E	cost
0	0	10
0	1	6
1	0	...
1	1	...

[Marinescu & Dechter, AIJ'09]

Anytime Performance

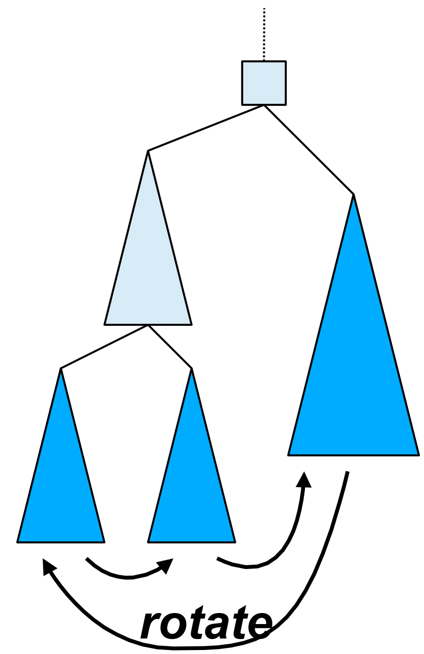
- OR Branch-and-Bound is anytime.
- But AND/OR breaks anytime behavior of depth-first scheme:
  - First anytime solution delayed until last subproblem starts processing.



[Otten & Dechter, AICom '12]

Anytime Performance

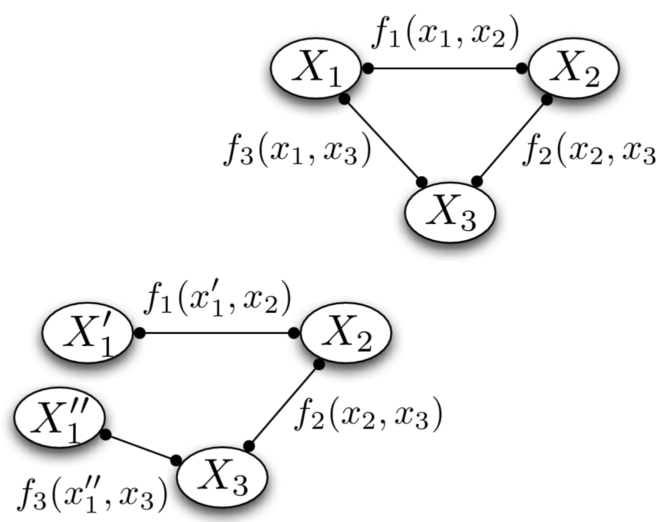
- OR Branch-and-Bound is anytime.
- But AND/OR breaks anytime behavior of depth-first scheme:
  - First anytime solution delayed until last subproblem starts processing.
- **Breadth-Rotating AOBB:**
  - Take turns processing subproblems.
    - Limit number of expansions per visit.
  - Solve each subproblem depth-first.
    - Maintain favorable complexity bounds.

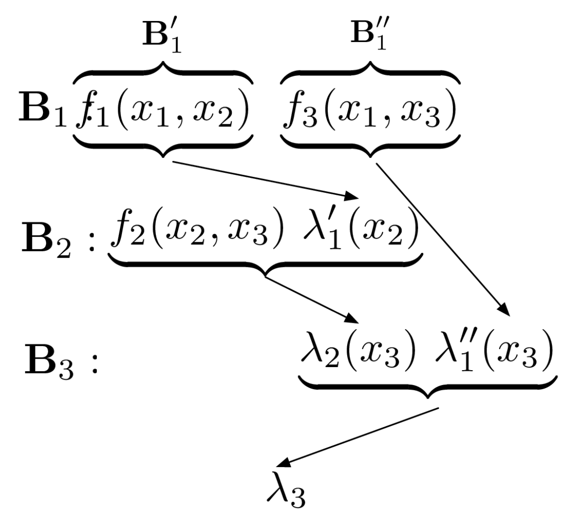


[Otten & Dechter, AICom '12]

Mini-buckets with Moment Matching

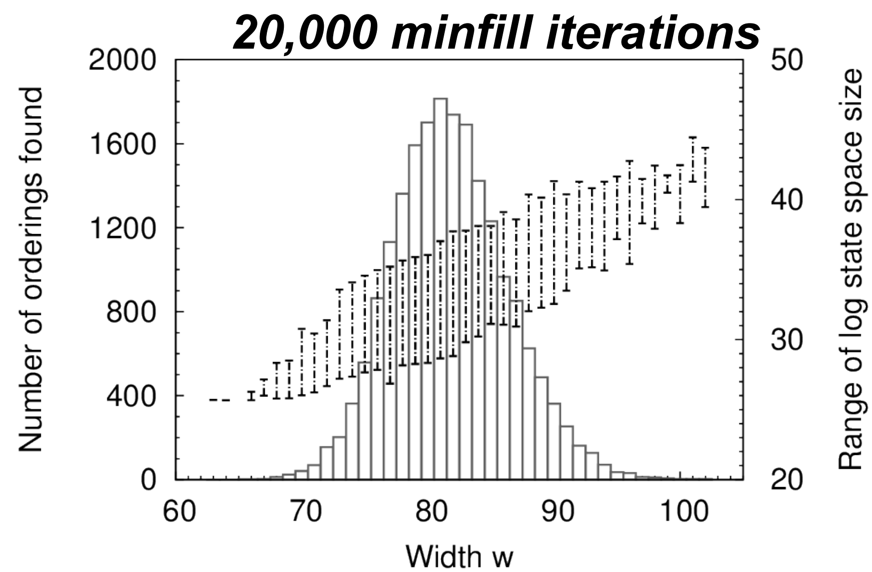
- Heuristic function generated from mini buckets.
  - Apply exact variable elimi-nation to relaxed problem.
  - *i*-bound parameter controls accuracy / complexity.
- Augmented with MPLP cost-shifting.
  - Max-marginal matching per bucket for tighter bounds.





Stochastic Variable Orderings

- AOBB complexity: $O(nk^w)$
  - High variance in width of orderings.
- Our implementation:
  - Minfill heuristic.
  - Random tie-breaking.
  - Allow deviation from heuristic optimum.
  - Highly optimized data structures, early termination.
    - Can do many thousands of iterations.



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

Putting It All Together

- Steps in competition entry for 1 hour track:
  1. MPLP cost-shifting on original graph (1 min / 2,000 iter).
  2. Stochastic local search (3 min).
  3. Find variable ordering (3 min / 30,000 iter).
  4. Join-graph MPLP cost-shifting (1 min / 1,000 iter).
  5. Compute mini-buckets with moment matching.
     - Highest possible *i*-bound for given memory limit.
  6. Run limited discrepancy search ($d_{max} = 2$).
  7. Run complete Breadth-Rotating AOBB.
- Final result in 1 hour category:
  - Daoopt: **-8.3214**, ficolofo: **-8.3196** ($\delta = 0.0018$)

BREN:ICS INFORMATION AND COMPUTER SCIENCES

UNIVERSITY of CALIFORNIA IRVINE

Thank you!

For publication see:

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

Kalev Kask
Irina Rish
Bozhena Bidyuk
Robert Mateescu
Radu Marinescu
Vibhav Gogate
Emma Rollon
Lars Otten
Natalia Flerova