Approximation Techniques Iterative bounded inference

COMPSCI 276, Fall 2014

Set 11: Rina Dechter

Reading: Class Notes, chapter 9, Darwiche chapters 14

Agenda

- Mini-bucket elimination
- Mini-clustering
- Iterative Belief propagation
- Iterative-join-graph propagation
  - IJGP complexity
  - Convergence and pair-wise consistency
  - Accuracy when converged
  - Belief Propagation and constraint propagation
- Using Mini-bucket as heuristics for optimization

Summary of IJGP so far

A spectrum of approximations.

IBP: results from applying IJGP to the dual joingraph.

Jointree algorithm: results from applying IJGP to a jointree (as a joingraph).

In between these two ends, we have a spectrum of joingraphs and corresponding factorizations, where IJGP seeks stationary points of the KL–divergence between these factorizations and the original distribution.

More On the Power of Belief Propagation

- BP as local minima of KL distance
- BP's power from constraint propagation perspective.

Inference Power of Loopy BP

- Comparison with iterative algorithms in constraint networks
- Zero-beliefs $\Leftrightarrow$ inconsistent assignments
- ε-small beliefs – experimental study

Constraint networks

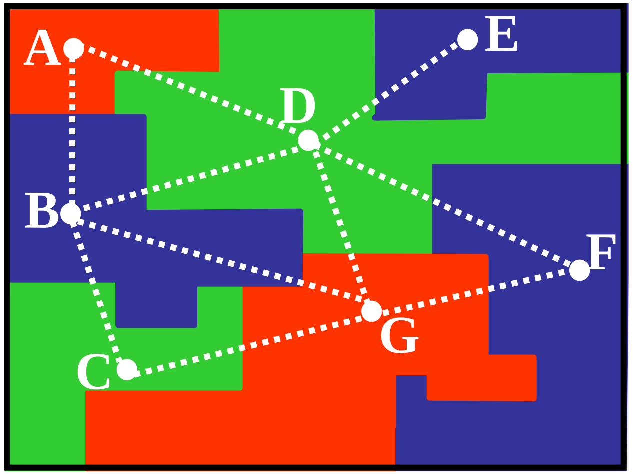
Map coloring

Variables: countries (A B C etc.)

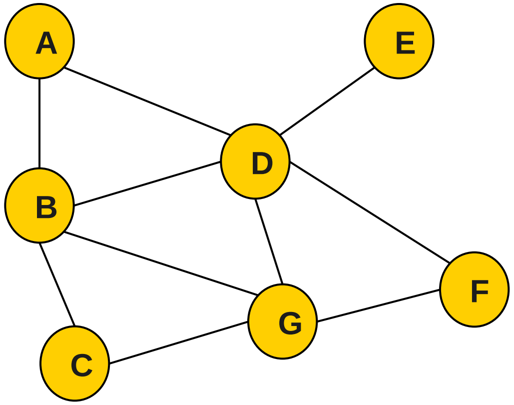
Values: colors (red green blue)

Constraints: **A ≠ B,** **A ≠ D, D ≠ E**, *etc.*

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

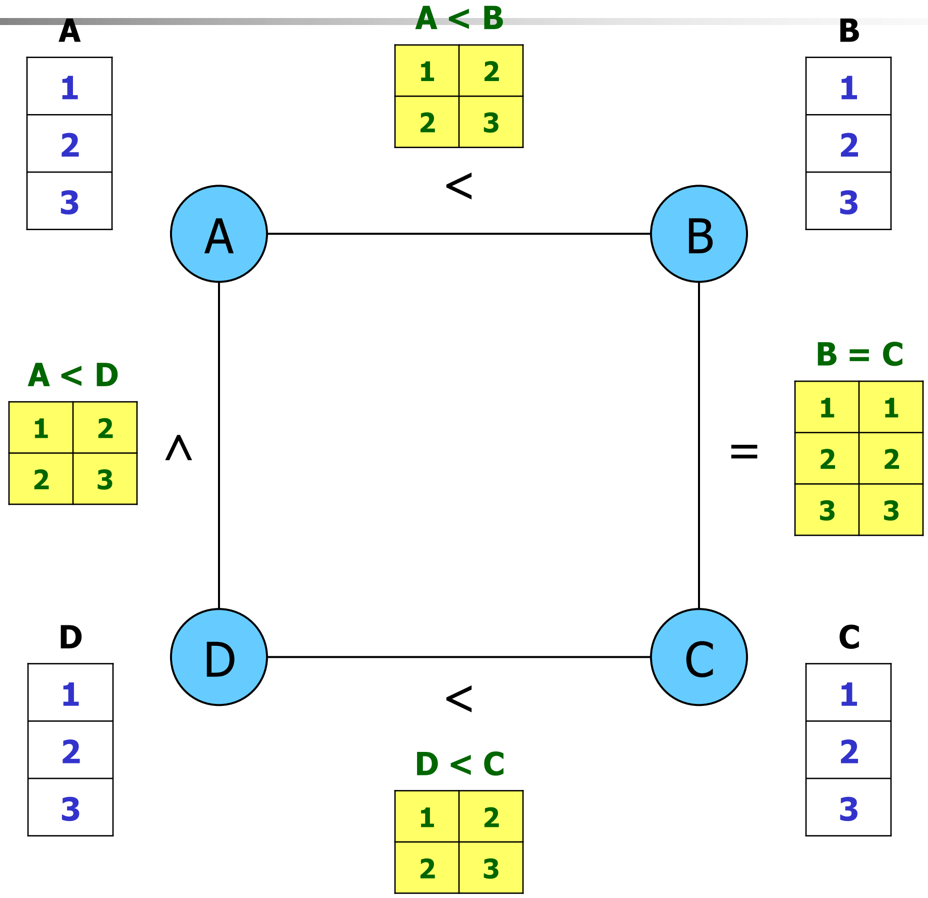


Constraint graph



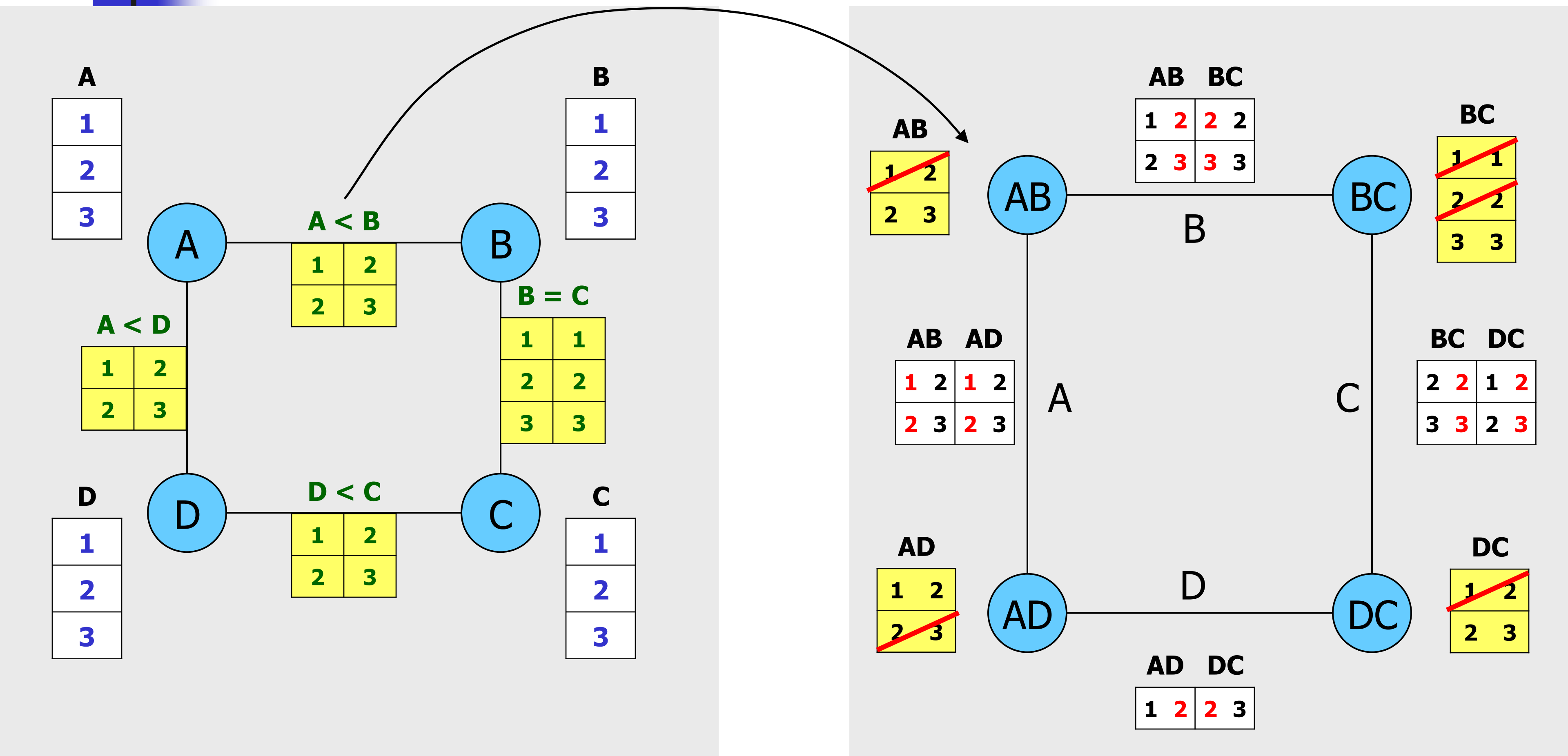
Arc-consistency

- Sound
- Incomplete
- Always converges (polynomial)



Relational Distributed Arc-Consistency

Primal **Dual**

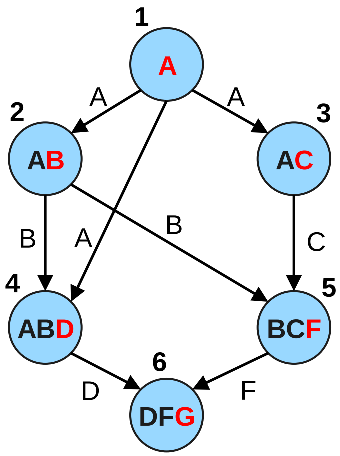


Flattening the Bayesian Network

A	P(A)
1	.2
2	.5
3	.3
…	0

A	B	P(B|A)
1	2	.3
1	3	.7
2	1	.4
2	3	.6
3	1	.1
3	2	.9
…	…	0

A	C	P(C|A)
1	2	1
3	2	1
…	…	0



A	B	D	P(D|A,B)
1	2	3	1
1	3	2	1
2	1	3	1
2	3	1	1
3	1	2	1
3	2	1	1
…	…	…	0

B	C	F	P(F|B,C)
1	2	3	1
3	2	1	1
…	…	…	0

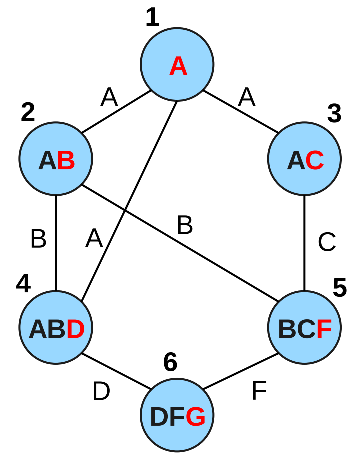
D	F	G	P(G|D,F)
1	2	3	1
2	1	3	1
…	…	…	0

Belief network

A
1
2
3

A	B
1	2
1	3
2	1
2	3
3	1
3	2

A	C
1	2
3	2



A	B	D
1	2	3
1	3	2
2	1	3
2	3	1
3	1	2
3	2	1

B	C	F
1	2	3
3	2	1

D	F	G
1	2	3
2	1	3

Flat constraint network

Belief Zero Propagation = Arc-Consistency

$$h_i^j = \sum_{elim(i,j)} \left(p_i \cdot \left(\prod_{\{k \in ne_j(i)\}} h_k^i \right)\right)$$

$$h_i^j = \pi_{l_{ij}} \left(R_i \bowtie \left(\bowtie_{k \in ne(i)} h_k^i\right)\right)$$

A	B	P(B|A)
1	2	>0
1	3	>0
2	1	>0
2	3	>0
3	1	>0
3	2	>0
…	…	0

h_1^2

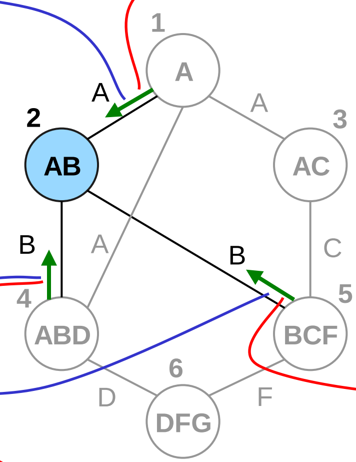
A	$h_1^2(A)$
1	>0
2	>0
3	>0
…	0

h_4^2

B	$h_4^2(B)$
1	>0
2	>0
3	>0
…	0

h_5^2

B	$h_5^2(B)$
1	>0
3	>0
…	0



A	B
1	2
1	3
2	1
2	3
3	1
3	2

h_1^2

A
1
2
3

h_4^2

B
1
2
3

h_5^2

B
1
3

Updated belief:

$$Bel(A,B) = P(B \mid A) \cdot h_1^2 \cdot h_4^2 \cdot h_5^2 =$$

$=$

A	B	Bel (A,B)
1	3	>0
2	1	>0
2	3	>0
3	1	>0
…	…	0

Updated relation:

$$R(A,B) = R(A,B) \bowtie h_1^2 \bowtie h_4^2 \bowtie h_5^2 =$$

$=$

A	B
1	3
2	1
2	3
3	1

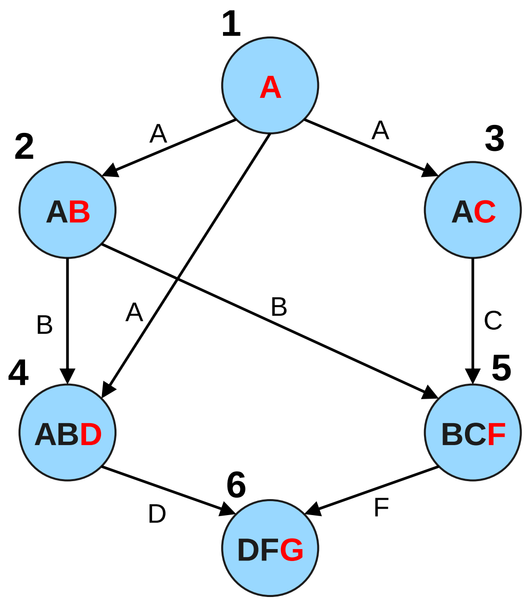
Flat Network - Example

R_1

A	P(A)
1	.2
2	.5
3	.3
…	0

R_2

A	B	P(B|A)
1	2	.3
1	3	.7
2	1	.4
2	3	.6
3	1	.1
3	2	.9
…	…	0



R_3

A	C	P(C|A)
1	2	1
3	2	1
…	…	0

R_4

A	B	D	P(D|A,B)
1	2	3	1
1	3	2	1
2	1	3	1
2	3	1	1
3	1	2	1
3	2	1	1
…	…	…	0

R_5

B	C	F	P(F|B,C)
1	2	3	1
3	2	1	1
…	…	…	0

R_6

D	F	G	P(G|D,F)
1	2	3	1
2	1	3	1
…	…	…	0

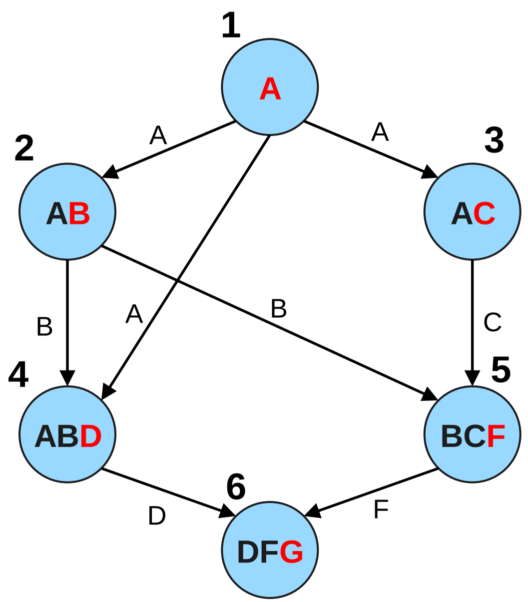
IBP Example – Iteration 1

R_1

A	P(A)
1	>0
3	>0
…	0

R_2

A	B	P(B|A)
1	3	1
2	1	>0
2	3	>0
3	1	1
…	…	0



R_3

A	C	P(C|A)
1	2	1
3	2	1
…	…	0

R_4

A	B	D	P(D|A,B)
1	3	2	1
2	3	1	1
3	1	2	1
3	2	1	1
…	…	…	0

R_5

B	C	F	P(F|B,C)
1	2	3	1
3	2	1	1
…	…	…	0

R_6

D	F	G	P(G|D,F)
2	1	3	1
…	…	…	0

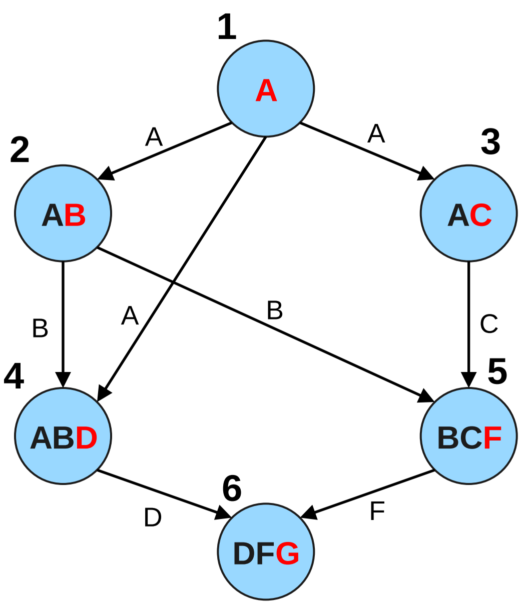
IBP Example – Iteration 2

R_1

A	P(A)
1	>0
3	>0
…	0

R_2

A	B	P(B|A)
1	3	1
3	1	1
…	…	0



R_3

A	C	P(C|A)
1	2	1
3	2	1
…	…	0

R_4

A	B	D	P(D|A,B)
1	3	2	1
3	1	2	1
…	…	…	0

R_5

B	C	F	P(F|B,C)
3	2	1	1
…	…	…	0

R_6

D	F	G	P(G|D,F)
2	1	3	1
…	…	…	0

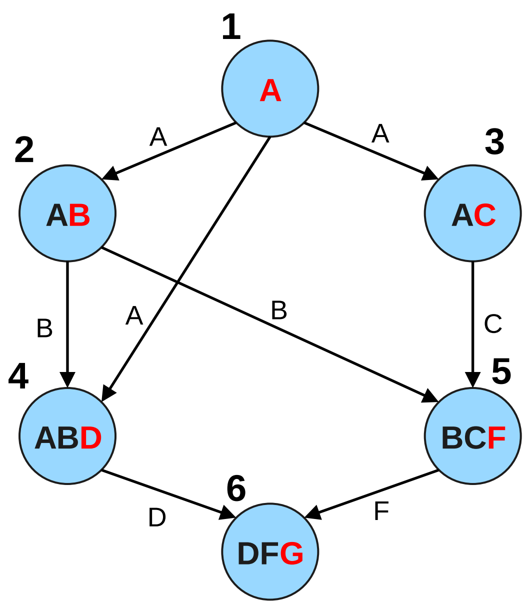
IBP Example – Iteration 3

R_1

A	P(A)
1	>0
3	>0
…	0

R_2

A	B	P(B|A)
1	3	1
…	…	0



R_3

A	C	P(C|A)
1	2	1
3	2	1
…	…	0

R_4

A	B	D	P(D|A,B)
1	3	2	1
3	1	2	1
…	…	…	0

R_5

B	C	F	P(F|B,C)
3	2	1	1
…	…	…	0

R_6

D	F	G	P(G|D,F)
2	1	3	1
…	…	…	0

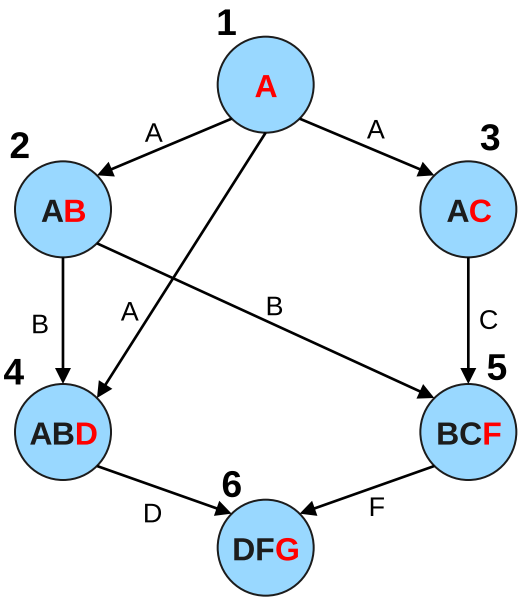
IBP Example – Iteration 4

R_1

A	P(A)
1	1
…	0

R_2

A	B	P(B|A)
1	3	1
…	…	0



R_3

A	C	P(C|A)
1	2	1
3	2	1
…	…	0

R_4

A	B	D	P(D|A,B)
1	3	2	1
…	…	…	0

R_5

B	C	F	P(F|B,C)
3	2	1	1
…	…	…	0

R_6

D	F	G	P(G|D,F)
2	1	3	1
…	…	…	0

IBP Example – Iteration 5

R_1

A	P(A)
1	1
…	0

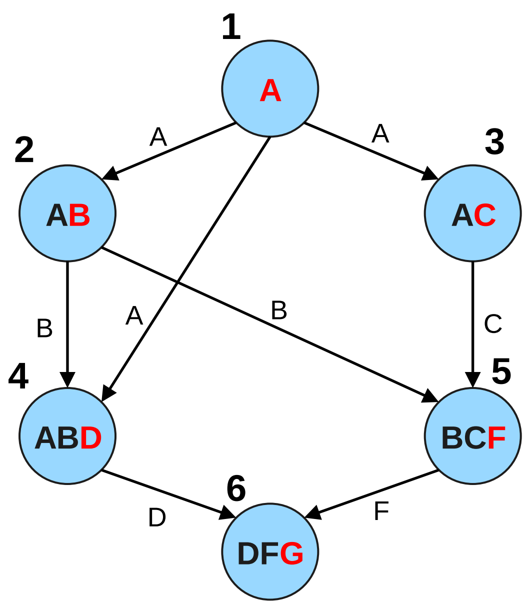
A	B	C	D	F	G	Belief
1	3	2	2	1	3	1
…	…	…	…	…	…	0

R_2

A	B	P(B|A)
1	3	1
…	…	0

R_3

A	C	P(C|A)
1	2	1
…	…	0



R_4

A	B	D	P(D|A,B)
1	3	2	1
…	…	…	0

R_5

B	C	F	P(F|B,C)
3	2	1	1
…	…	…	0

R_6

D	F	G	P(G|D,F)
2	1	3	1
…	…	…	0

IBP – inference power for zero beliefs

- **Theorem:**

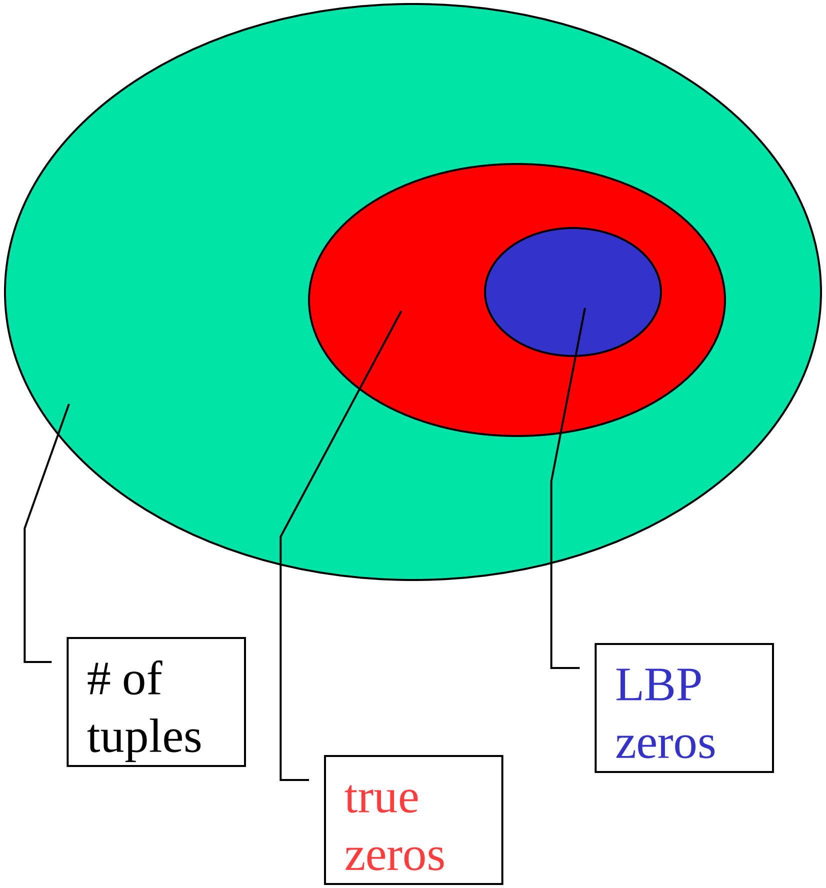
  Trace of zero beliefs of Iterative Belief Propagation =
  Trace of invalid tuples of arc-consistency on flat network

- **Soundness:**
  - The inference of zero beliefs by IBP converges in a finite number of iterations
  - all the inferred zero beliefs are correct

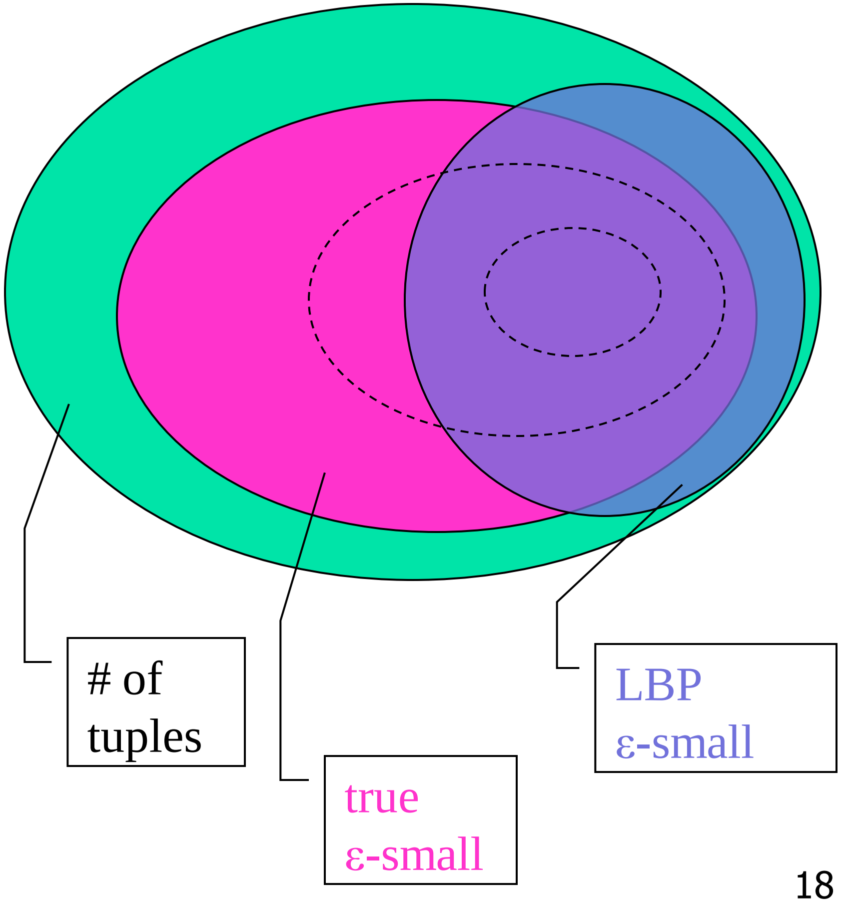
- **Incompleteness:**
  - IBP may not infer all the true zero beliefs

Zero and ε-Small Beliefs

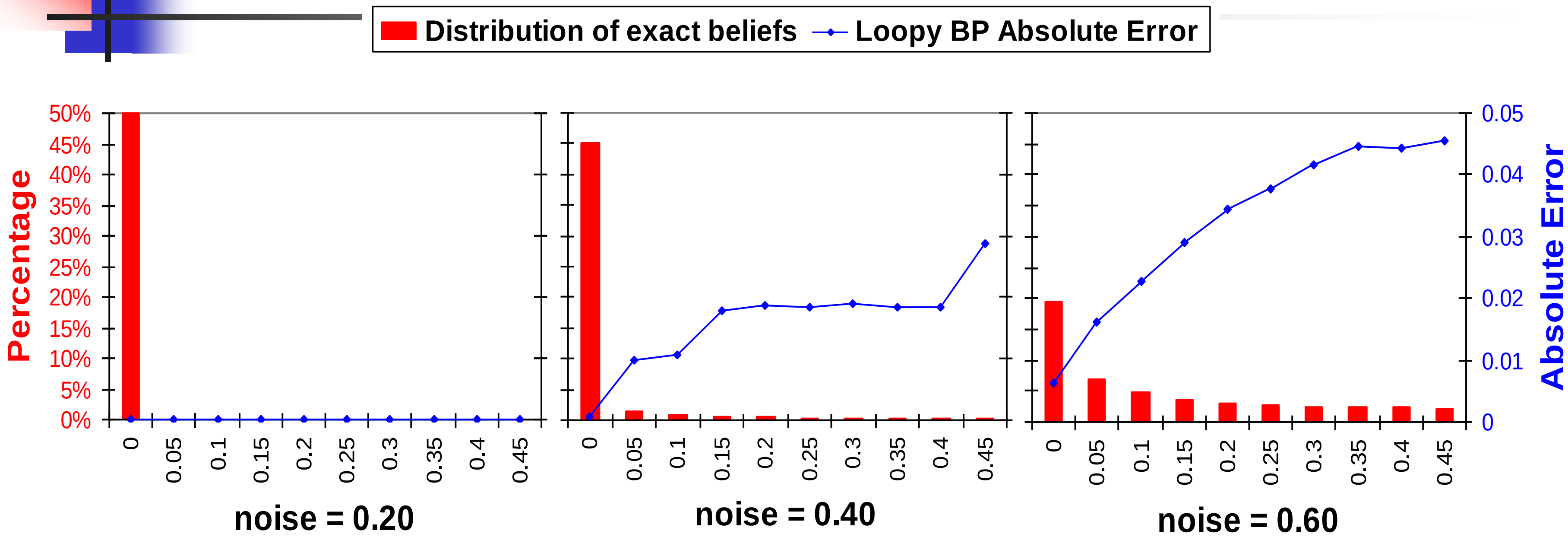
Zero beliefs



ε-small beliefs

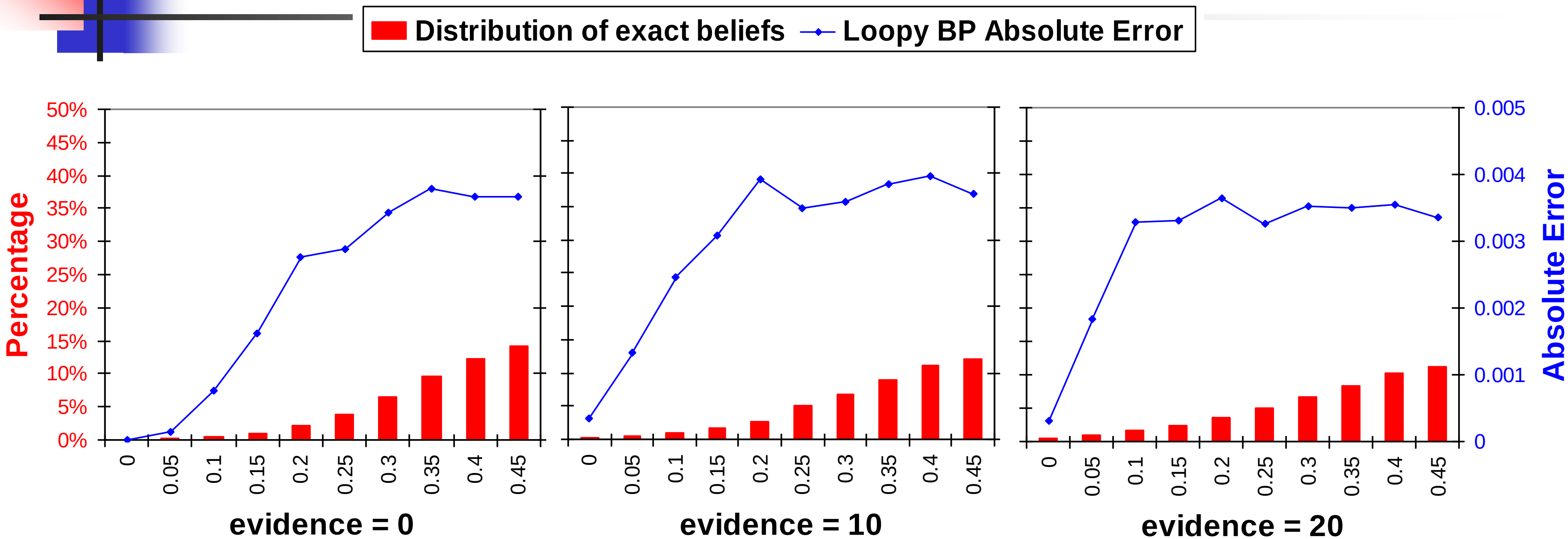


Coding Networks



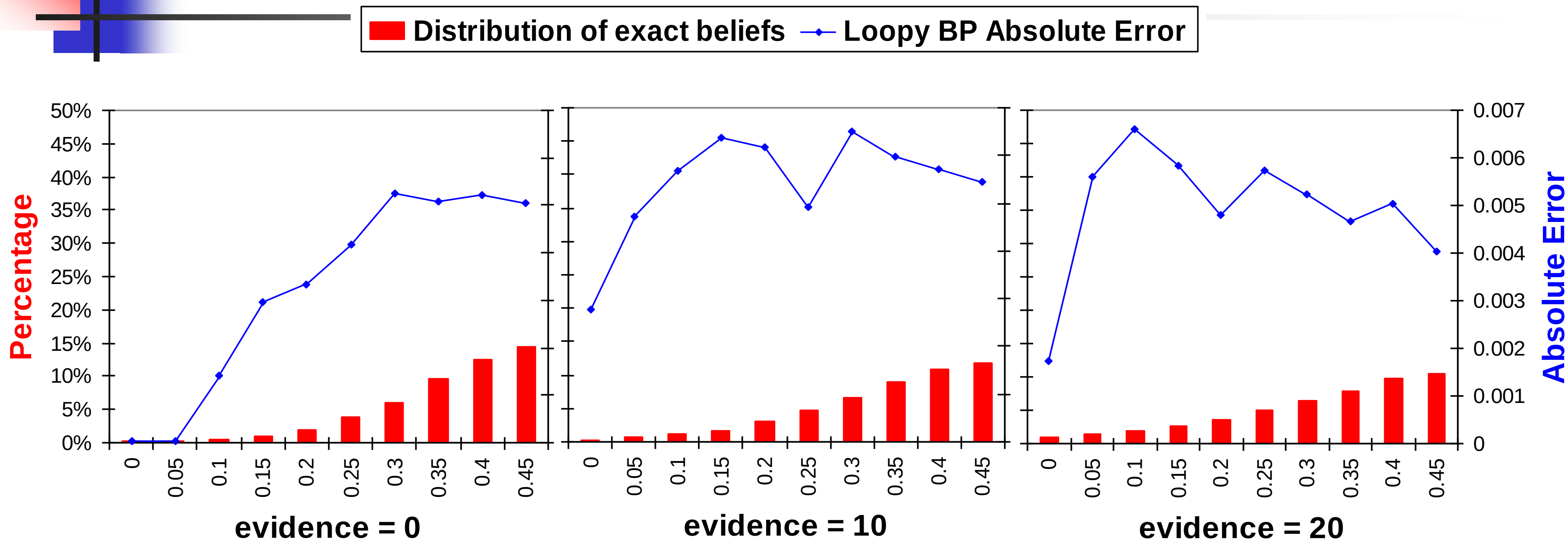
N=200, 1000 instances, treewidth=15

10x10 Grids



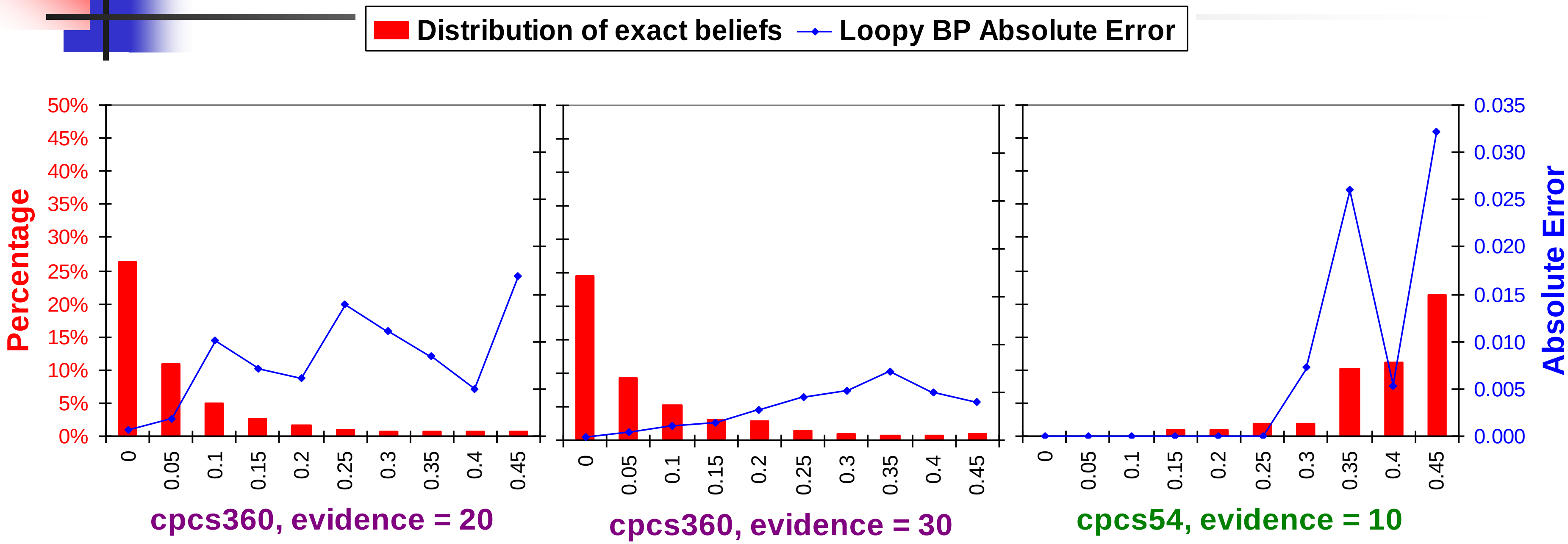
N=100, 100 instances, w*=15

Random Networks



N=80, 100 instances, w*=15

CPCS 54, CPCS360



CPCS360: 5 instances, w*=20

CPCS54: 100 instances, w*=15

Experimental Results

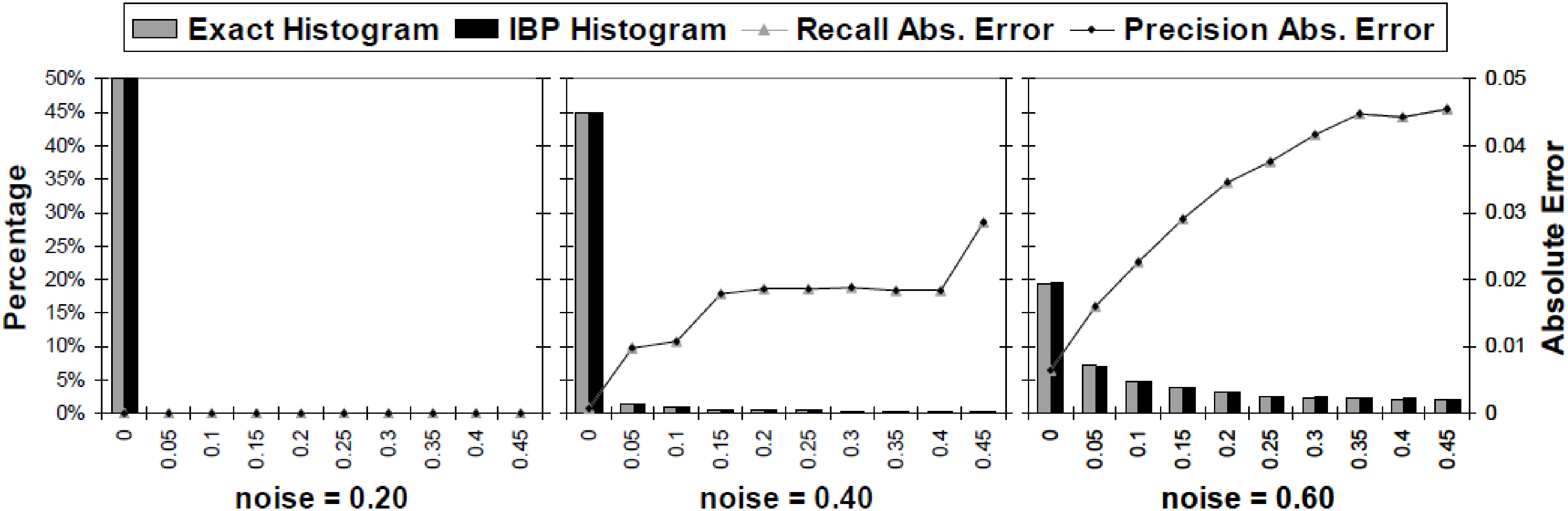
We investigated empirically if the results for zero beliefs extend to ε-small beliefs (ε > 0)

Have determinism?

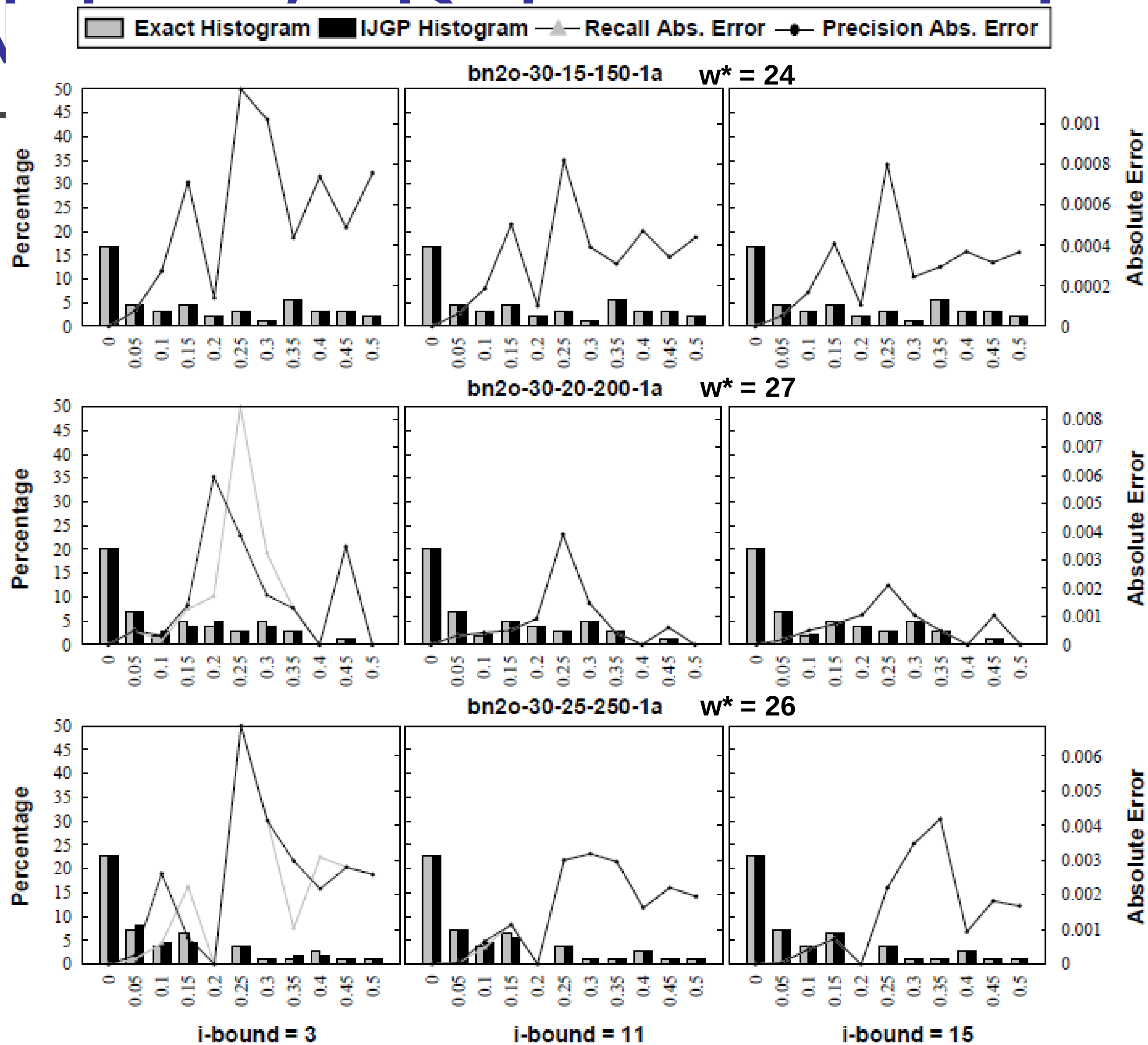
- Network types:
  - YES
    - Coding
    - Linkage analysis*
    - Grids*
  - NO
    - Two-layer noisy-OR*
    - CPCS54, CPCS360
- Measures:
  - Exact/IJGP histogram
  - Recall absolute error
  - Precision absolute error
- Algorithms:
  - IBP
  - IJGP

* Instances from the UAI08 competition

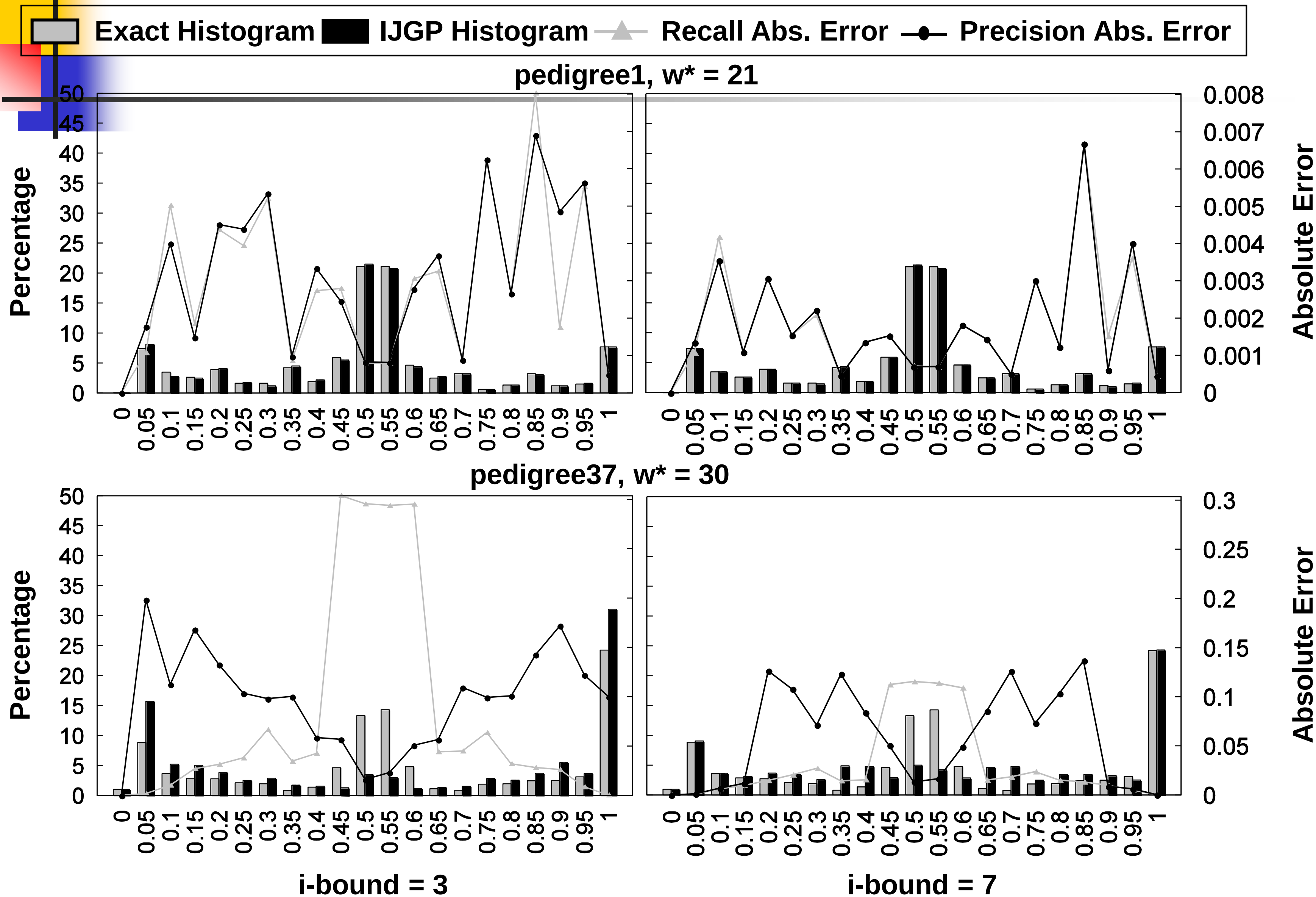
Networks with Determinism: Coding



N=200, 1000 instances, w*=15

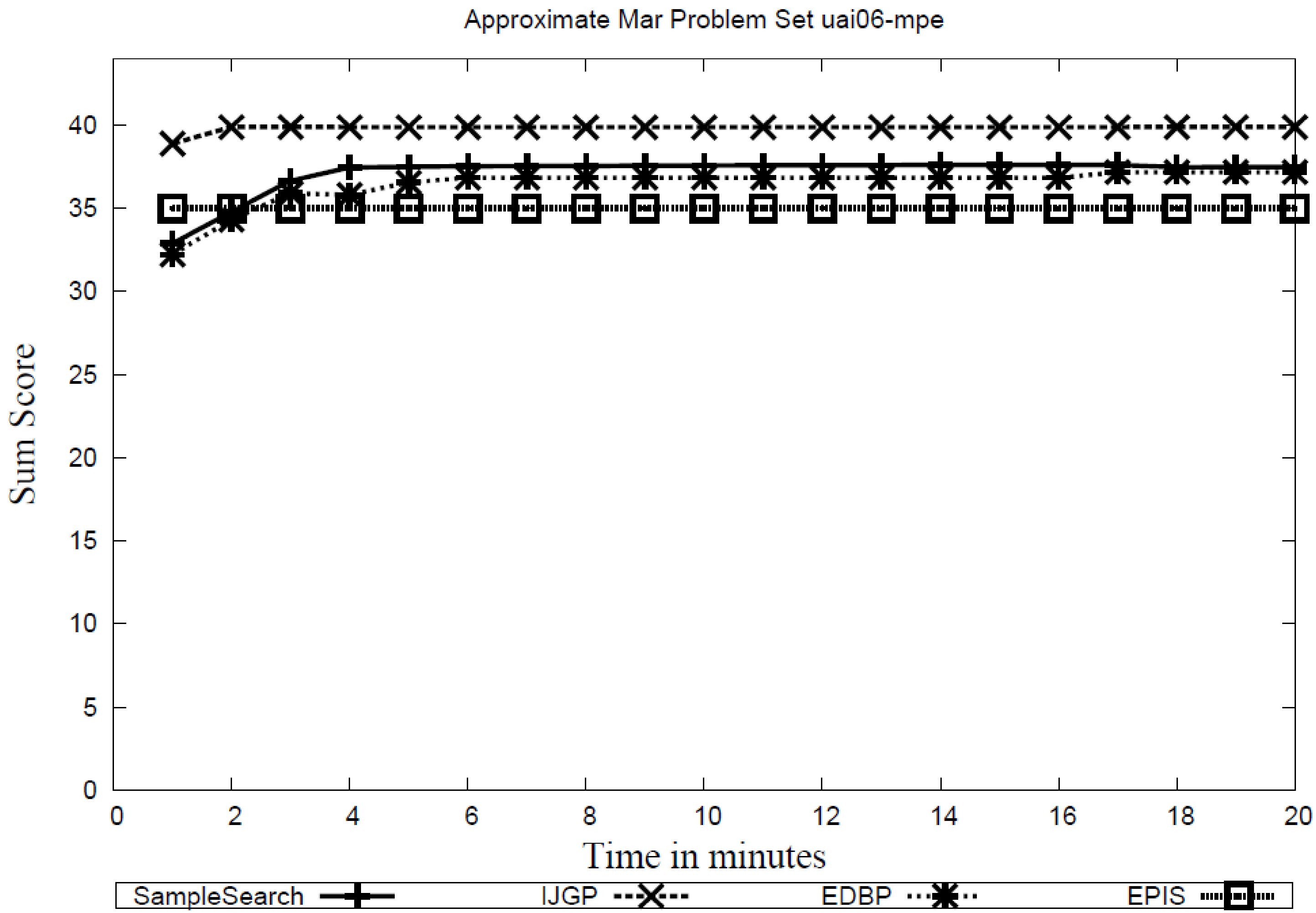

Exact Histogram
IJGP Histogram
Recall Abs. Error
Precision Abs. Error
bn2o-30-15-150-1a
w* = 24
bn2o-30-20-200-1a
w* = 27
bn2o-30-25-250-1a
w* = 26
Percentage
Absolute Error
i-bound = 3
i-bound = 11
i-bound = 15

Nets with Determinism: Linkage

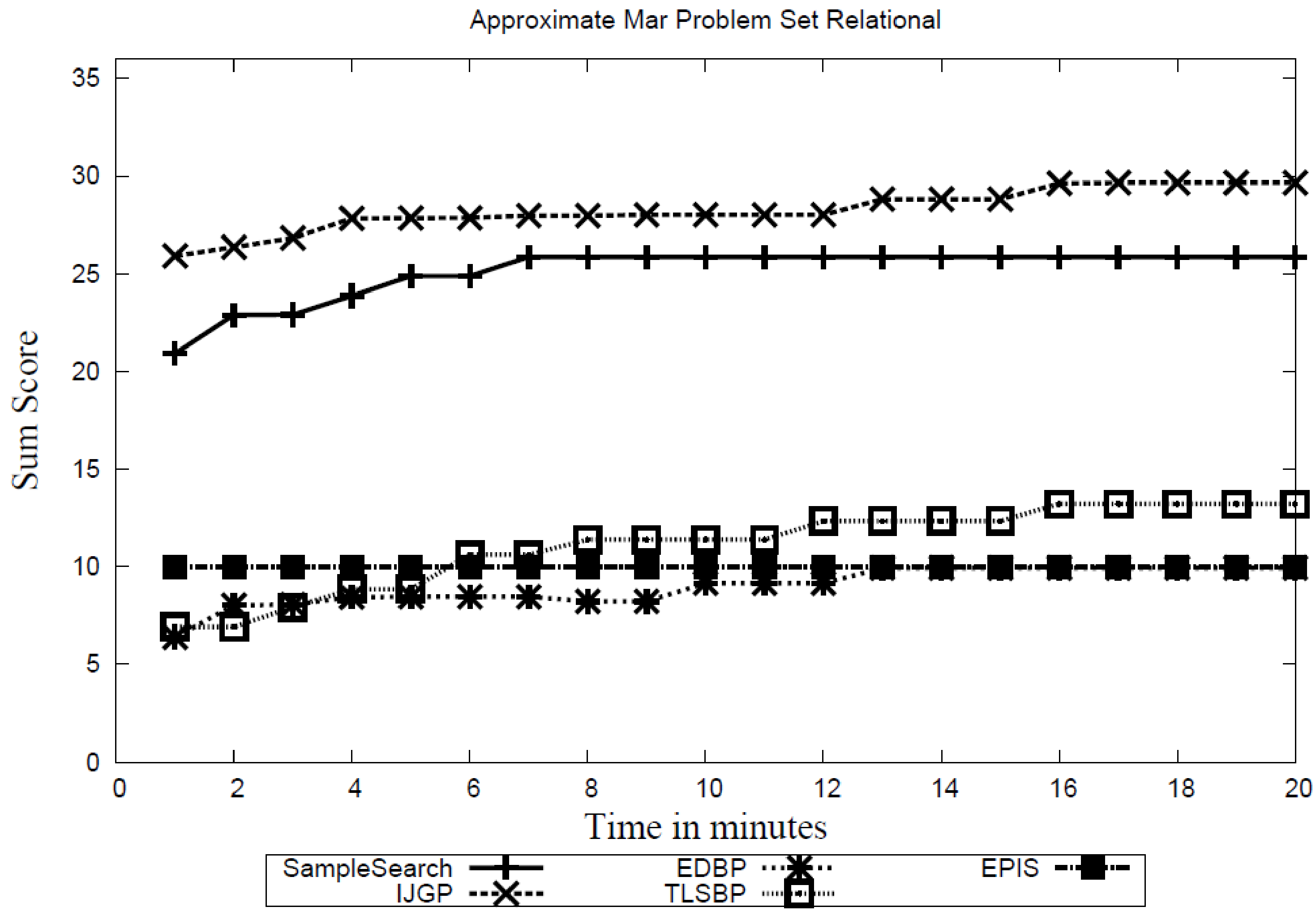


Some competition comparison

IJGP on UAI06 problems



IJGP on Set Relational

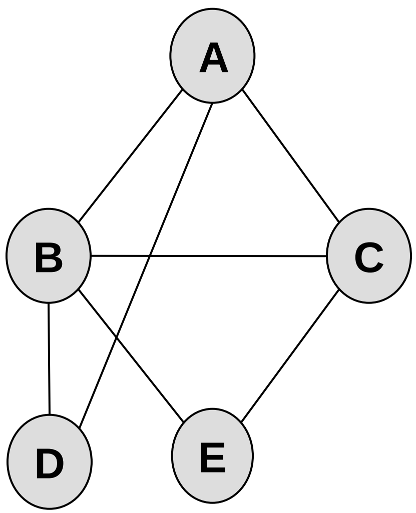


Agenda

- Mini-bucket elimination
- Mini-clustering
- Iterative Belief propagation
- Iterative-join-graph propagation
  - IJGP complexity
  - Convergence and pair-wise consistency
  - Accuracy when converged
  - Belief Propagation and constraint propagation
- Using Mini-bucket as heuristics for optimization

(did not go beyond this slides)

Mini-Bucket can be used to guide more than one solution



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: e = 0 ***h^C(e,a)*** ***h^D(e,a)***

bucket A: ***h^E(a)***

L = lower bound

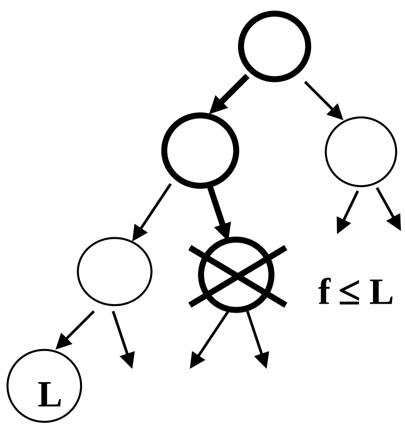
Basic Heuristic Search Schemes

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

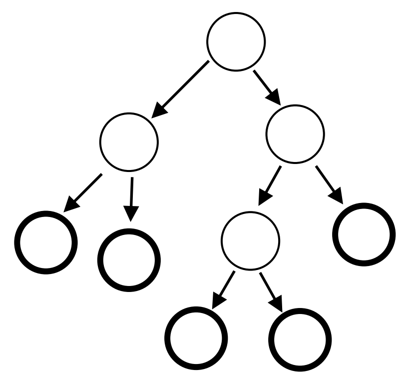
Linear space (or more)



2. Best-First Search

Always expand the node with the highest heuristic value **$f(x^p)$**

Needs lots of memory



Heuristic search

- Mini-buckets record upper-bound heuristics
- The evaluation function over $\overline{\mathrm{x}}_{\mathrm{p}} = (x_1,\ldots x_p)$

$$f(\bar{x}_p) = g(\bar{x}_p)h(\bar{x}_p)$$

$$g(\bar{x}_p) = \prod_{i=1}^{p-1} P(x_i \mid pa_i)$$

$$h(\bar{x}_p) = \prod_{h_j \in bucket_p} h_j$$

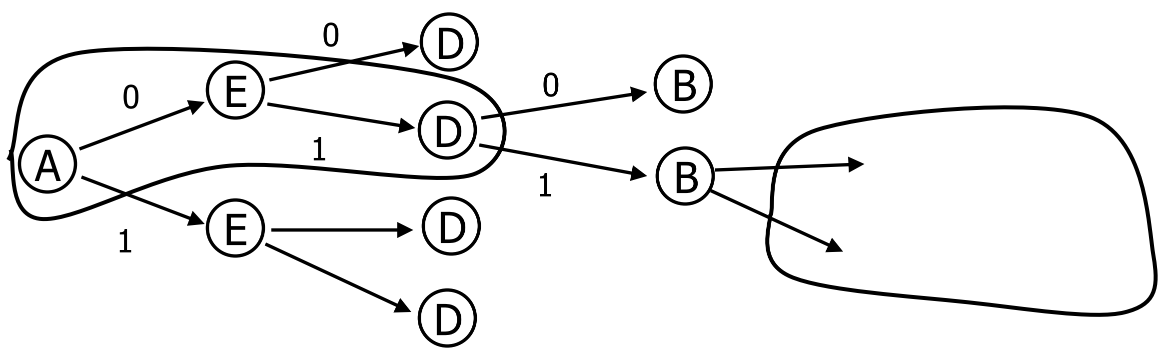
- **Best-first:** expand a node with maximal evaluation function
- **Branch and Bound:** prune if f <= upper bound
- **Properties:**
  - an exact algorithm
  - Better heuristics lead to more pruning

Heuristic Function

Given a cost function

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

Define an evaluation function over a partial assignment as the probability of it's best extension

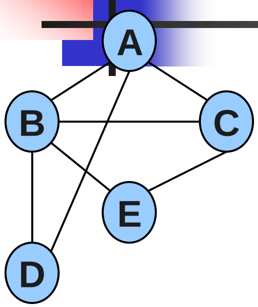


$$f^*(a,e,d) = \max_{b,c} P(a,b,c,d,e) =$$

$$= \underbrace{P(a)}_{g(a,e,d)} \bullet \underbrace{\max_{b,c} P(b|a) \bullet P(c|a) \bullet P(e|b,c) \bullet P(d|a,b)}_{H^*(a,e,d)}$$

$$= g(a,e,d) \bullet H^*(a,e,d)$$

MBE Heuristics



Cost Network

- 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

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

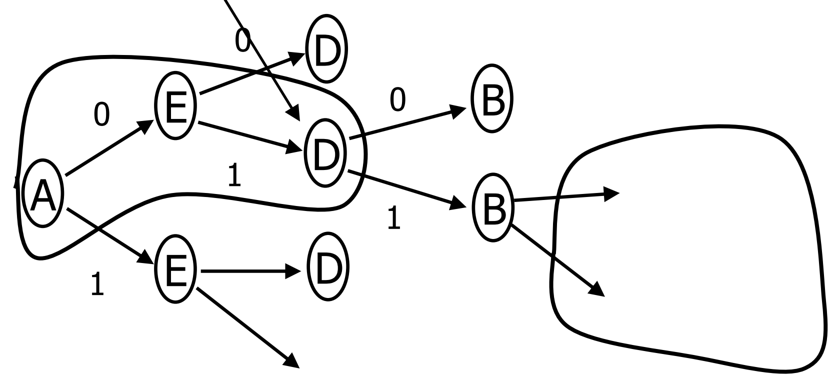
C: F(C,A) ***$h^B(E,C)$***

D: ***$h^B(D,A)$***

E: ***$h^C(E,A)$***

A: F(A) ***$h^E(A)$*** ***$h^D(A)$***

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

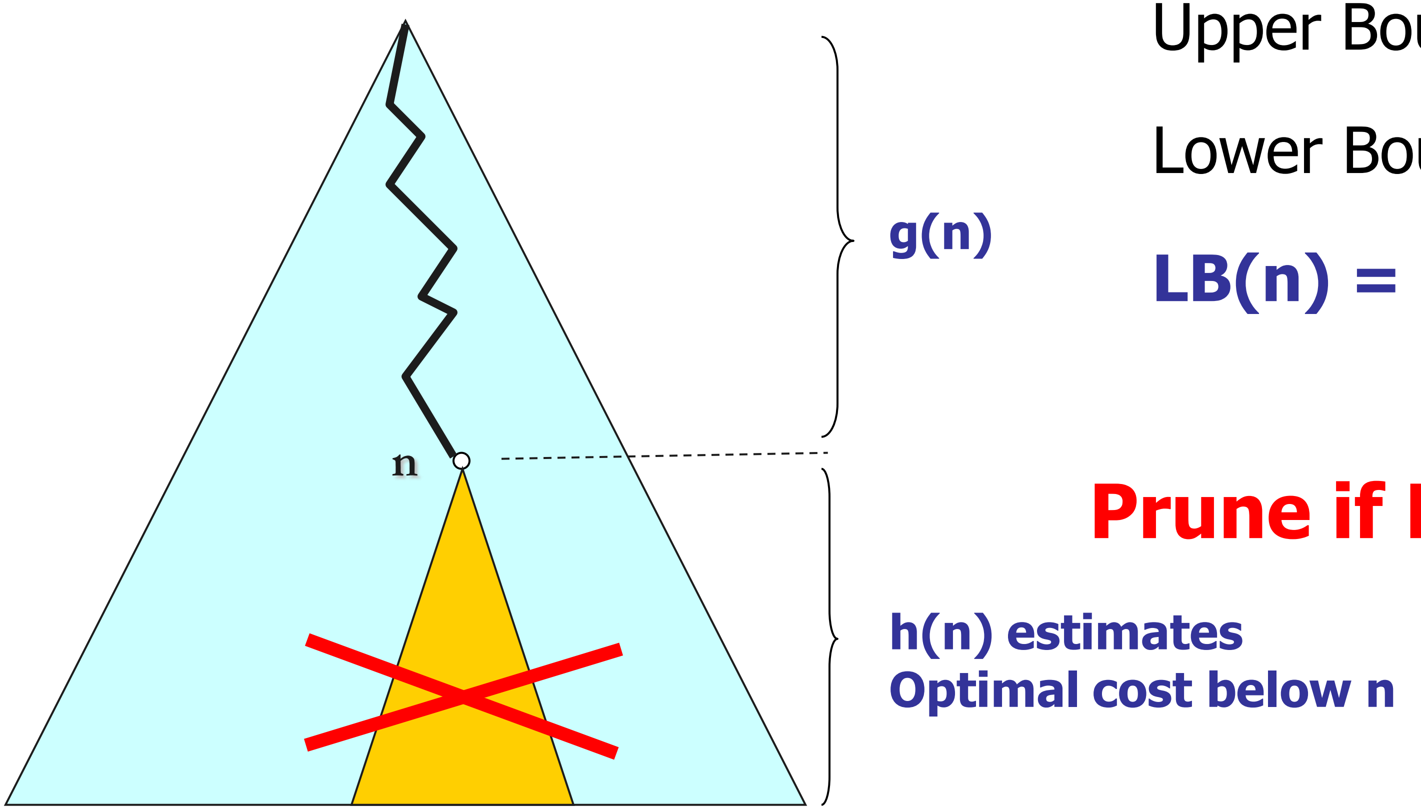


$$f(a,e,D) = \underbrace{F(a)}_{g} + \underbrace{h^B(D,a) + h^C(e,a)}_{h \text{ – is admissible}}$$

Properties

- Heuristic is consistent/monotone
- Heuristic is admissible
- Heuristic is computed in linear time
- IMPORTANT:
  - Mini-buckets generate heuristics of varying strength using control parameter – bound i
  - Higher bound -> more preprocessing -> stronger heuristics -> less search
  - Allows controlled trade-off between preprocessing and search

Classic Branch-and-Bound



OR Search Tree

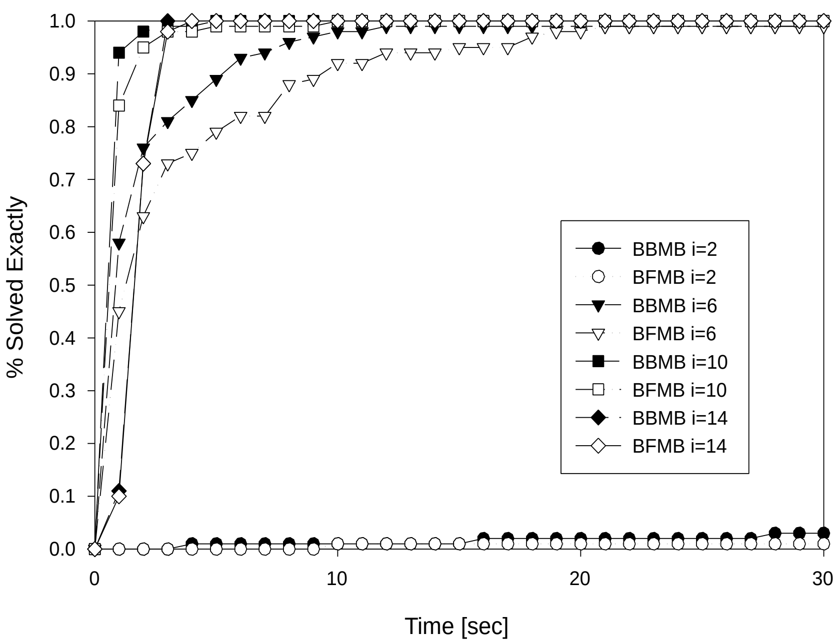
Upper Bound **UB**

Lower Bound **LB**

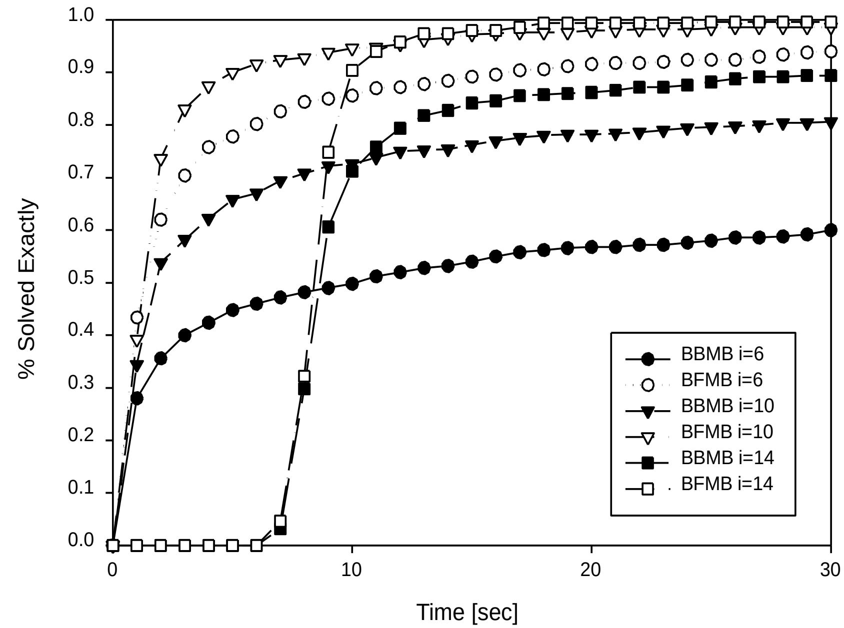
LB(n) = g(n) + h(n)

Prune if LB(n) ≥ UB

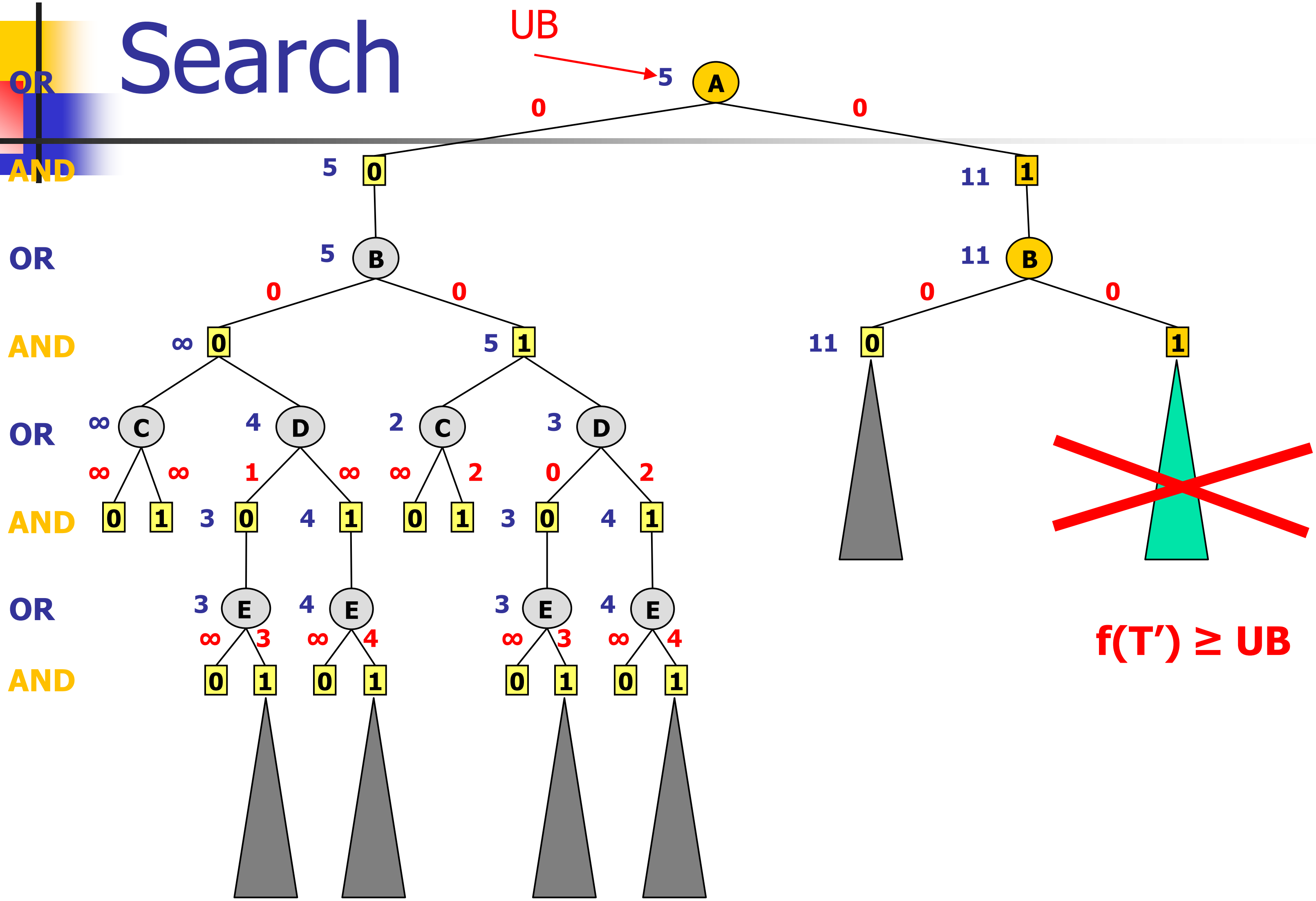
Empirical Evaluation of mini-bucket heuristics (Kask and Dechter, 1999)



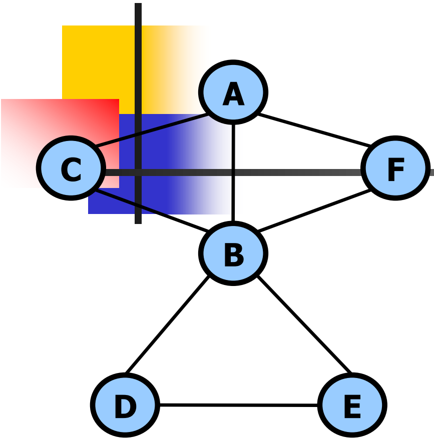
Random Coding, K=100, noise 0.32



AND/OR Branch-and-Bound Search



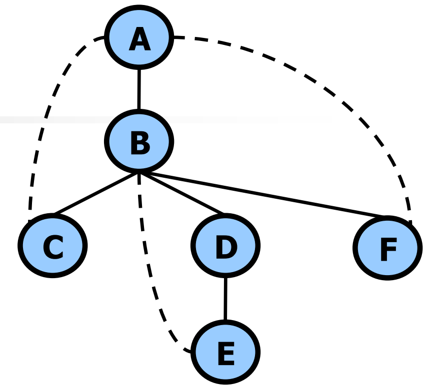
Heuristic Evaluation Function

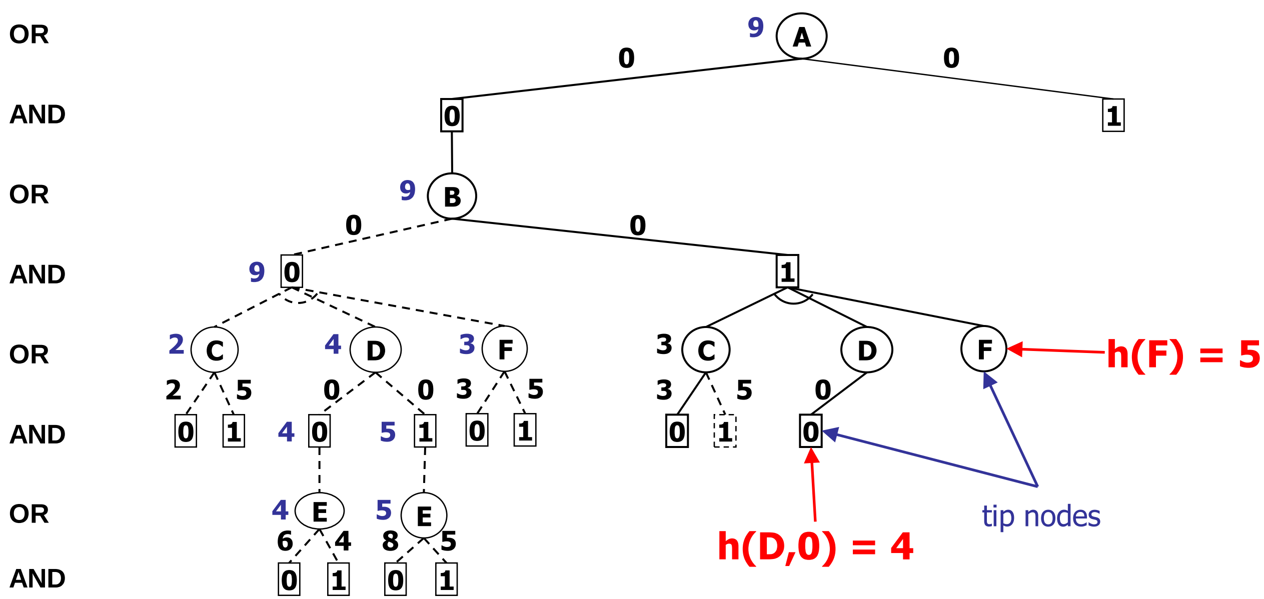


A	B	C	$f_1(ABC)$
0	0	0	2
0	0	1	5
0	1	0	3
0	1	1	5
1	0	0	9
1	0	1	3
1	1	0	7
1	1	1	2

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





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

Software & Competitions

- **How to use the software**
  - http://graphmod.ics.uci.edu/group/Software
  - http://mulcyber.toulouse.inra.fr/projects/toulbar2
- **Reports on competitions**
  - UAI-2006, 2008, 2010 Competitions
    - PE, MAR, MPE tasks
  - CP-2006 Competition
    - WCSP task

Toulbar2 and aolib

- toulbar2

  http://mulcyber.toulouse.inra.fr/gf/project/toulbar2

  (Open source WCSP, MPE solver in C++)

- aolib

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

  (WCSP, MPE, ILP solver in C++, inference and counting)

- Large set of benchmarks

  http://carlit.toulouse.inra.fr/cgi-bin/awki.cgi/SoftCSP

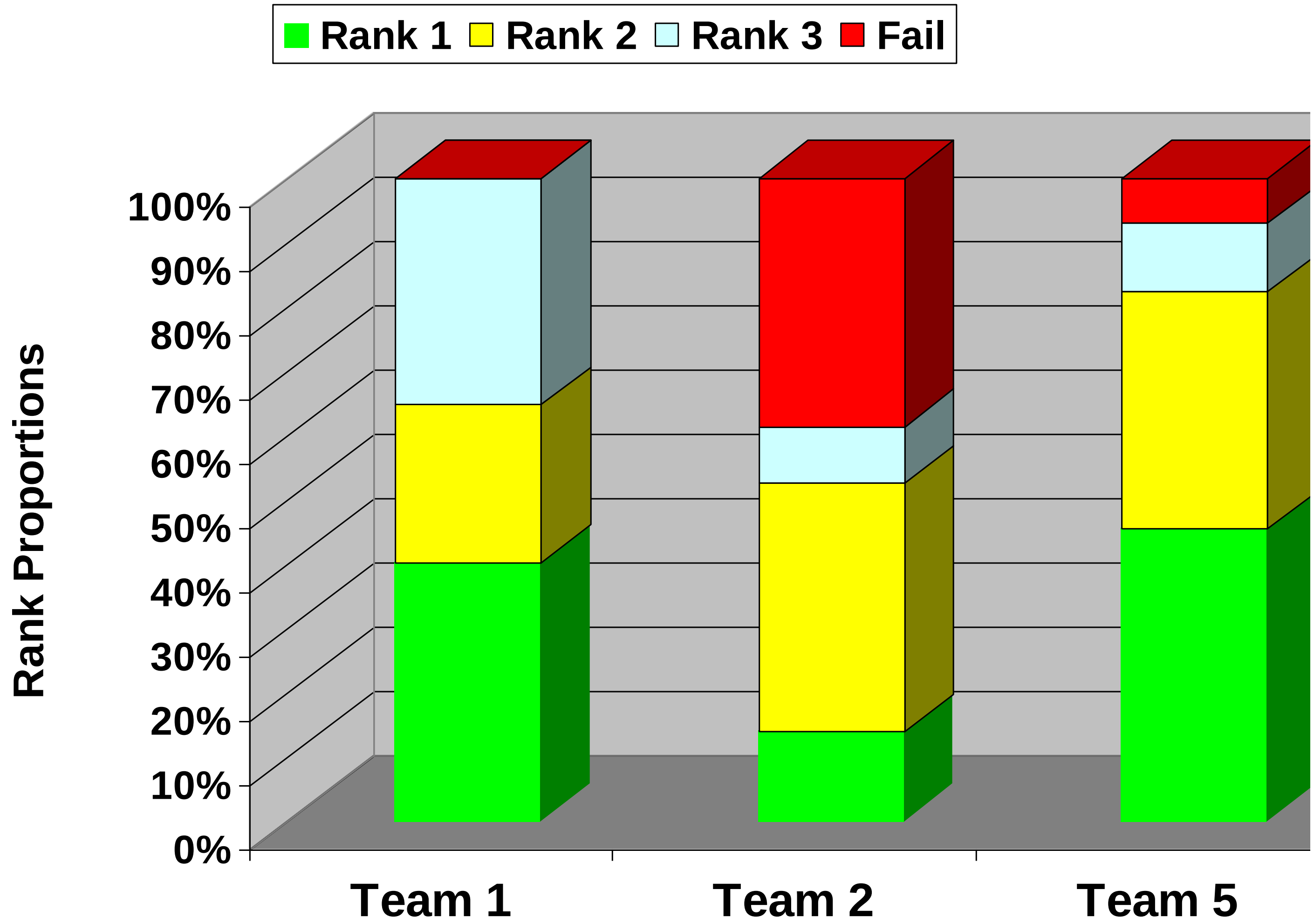
  http://graphmod.ics.uci.edu/group/Repository

UAI-2006 Competition

- **Team 1 (UCLA)**
  - David Allen, Mark Chavira, Arthur Choi, Adnan Darwiche
- **Team 2 (IET)**
  - Masami Takikawa, Hans Dettmar, Francis Fung, Rick Kissh
- **Team 5 (UCI)**
  - Radu Marinescu, Robert Mateescu, Rina Dechter
  - Used **AOBB-C+SMB(i)** solver for MPE

UAI-2006 Results

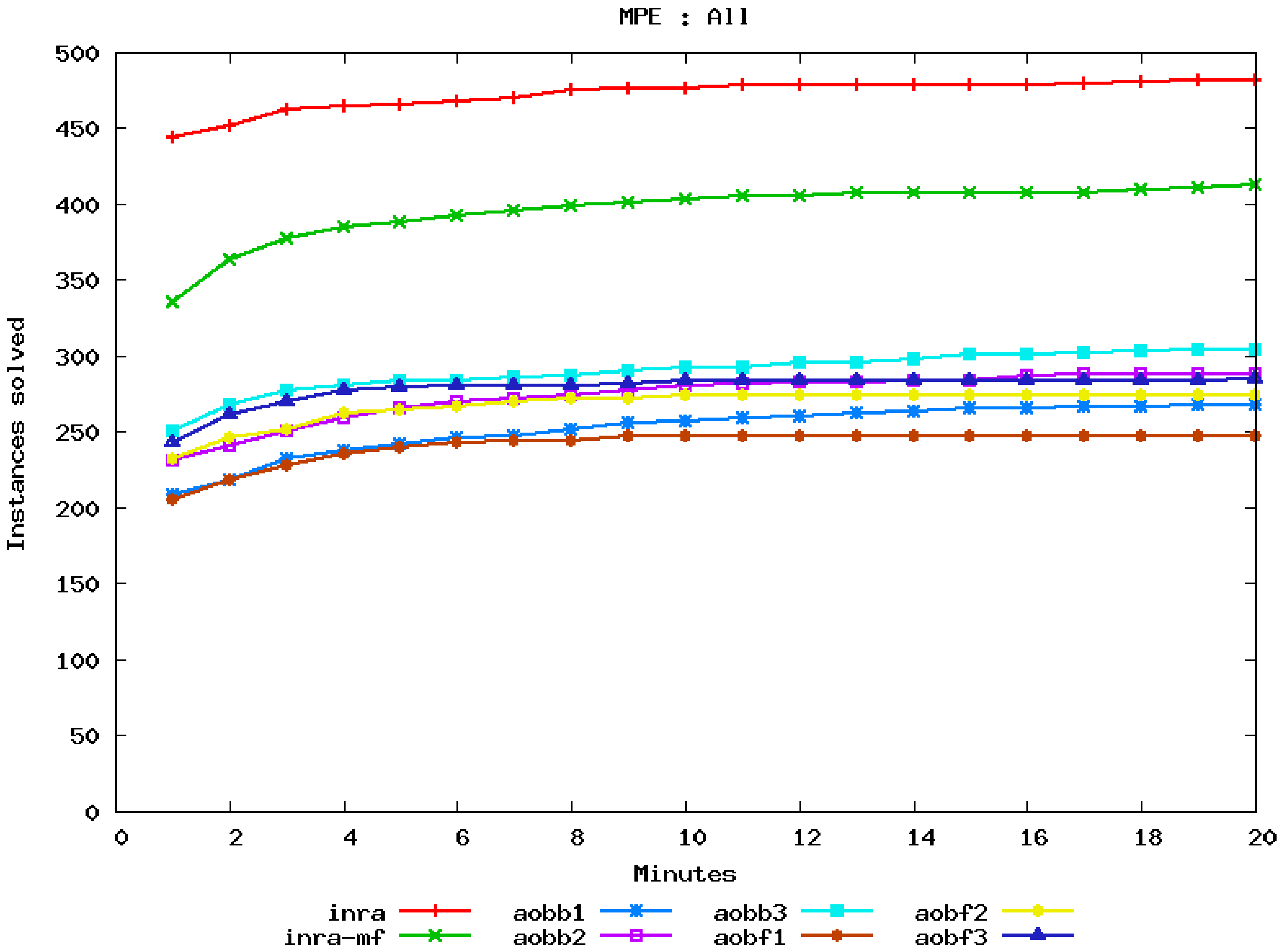
Rank Proportions (how often was each team a particular rank, rank 1 is best)



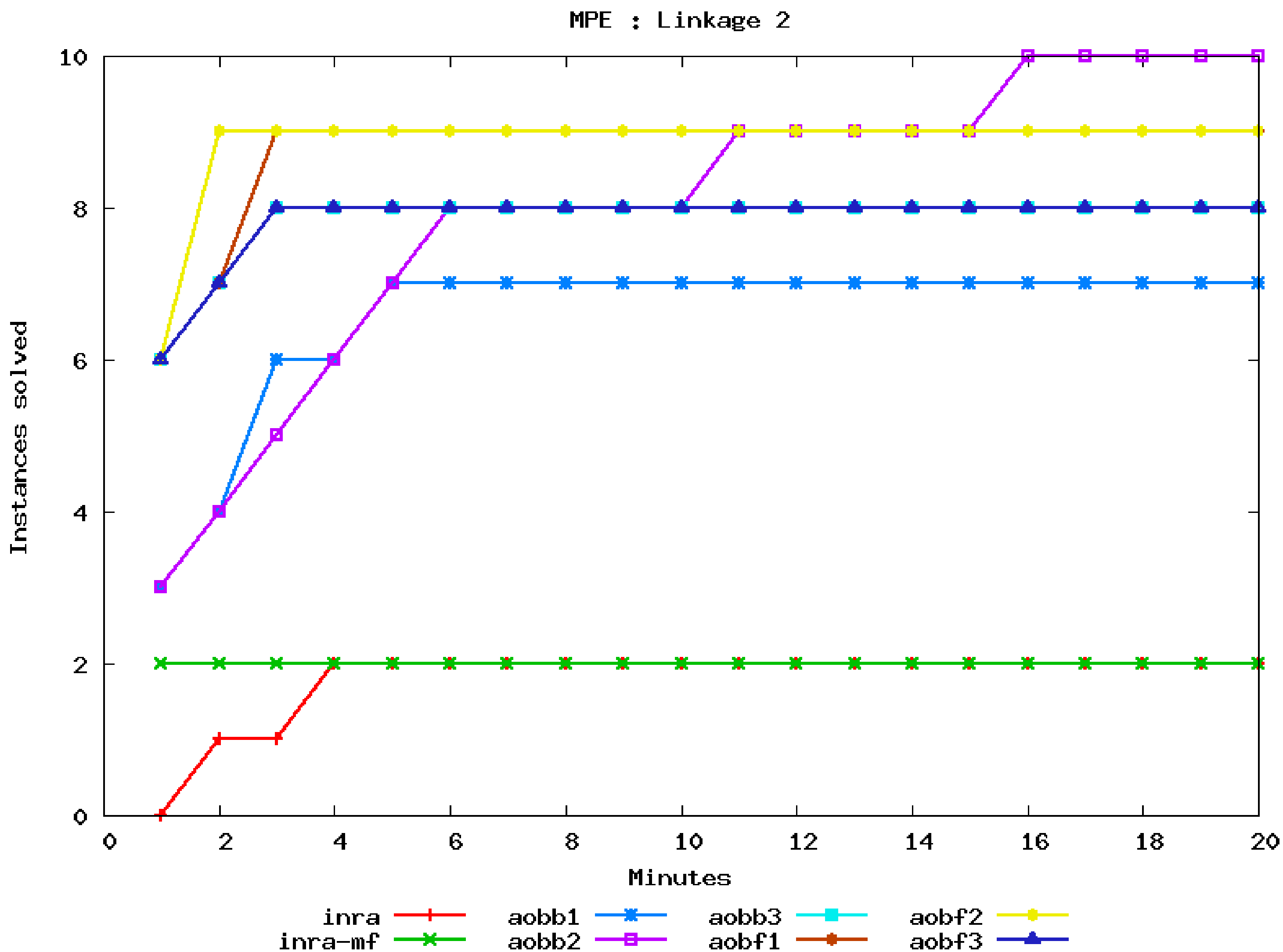
UAI-2008 Competition

- **AOBB-C+SMB(i) – (i = 18, 20, 22)**
  - AND/OR Branch-and-Bound with pre-compiled mini-bucket heuristics (i-bound), full caching, static pseudo-trees, constraint propagation
- **AOBF-C+SMB(i) – (i = 18, 20, 22)**
  - AND/OR Best-First search with pre-compiled mini-bucket heuristics (i-bound), full caching, static pseudo-trees, no constraint propagation
- **Toulbar2**
  - OR Branch-and-Bound, dynamic variable/value orderings, EDAC consistency for binary and ternary cost functions, variable elimination of small degree (2) during search
- **Toulbar2/BTD**
  - DFBB exploiting a tree decomposition (AND/OR), same search inside clusters as toulbar2, full caching (no cluster merging), combines RDS and EDAC, and caching lower bounds

UAI-2008 Results



UAI-2008 Results (contd.)



UAI-2010 Competition

- Tasks
  - PR: probability of evidence
  - MAR: posterior marginals
  - MPE: most probable explanation
- 3 tracks: 20 sec, 20 min, 1 hour
  - PR, MAR - 204 instances; MPE - 442 instances
    - CSP, grids, image alignment, medical diagnosis, object detection, pedigree, protein folding, protein-protein interaction, relational model, segmentation
- Exact and approximate solvers

UAI-2010 Results

- MAR task (Mateescu et al, JAIR2010), (Dechter et al, UAI2002)
  - **1st place** (20 min, 1 hour) – (impl. by Vibhav Gogate)
  - Anytime **IJGP(i**) with randomized orderings and SAT based domain pruning
- PR task (Gogate, Domingos and Dechter UAI2010)
  - **1st place** (20 min, 1 hour) – (impl. by Vibhav Gogate)
  - Formula **SampleSearch** with IJGP(3) based importance distribution, w-cutset sampling, minisat based search, rejection control
- MPE task (Marinescu and Dechter, AIJ2009), (Otten and Dechter, ISAIM2010)
  - **3rd place** (all tracks) – (impl. by Lars Otten)
  - **AND/OR BnB** with mini-buckets, randomized min-fill based pseudo tree, LDS based search for initial upper bound

DISCML 2012 – NIPS Workshop

Winning the PASCAL 2011 MAP Challenge with Enhanced AND/OR Branch-and-Bound

Lars Otten, Alexander Ihler,
Kalev Kask, Rina Dechter

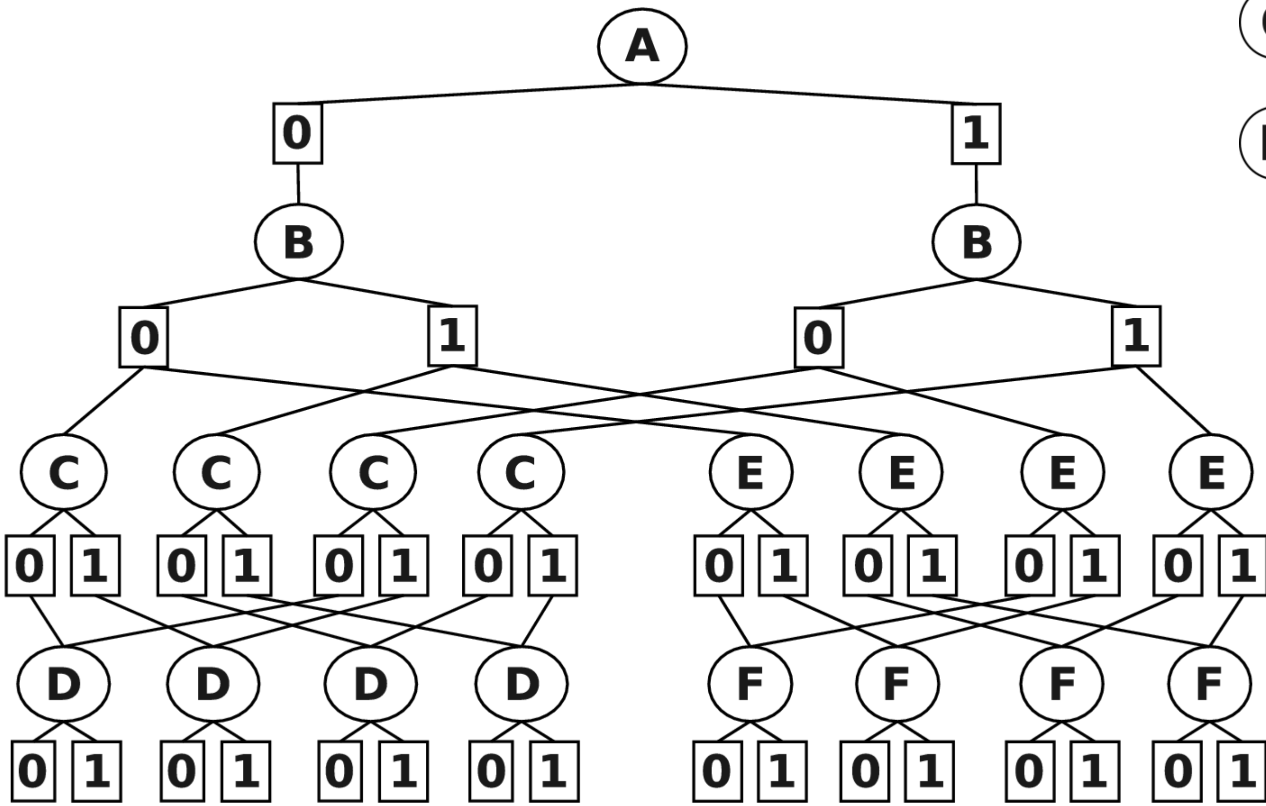
Dept. of Computer Science
University of California, Irvine

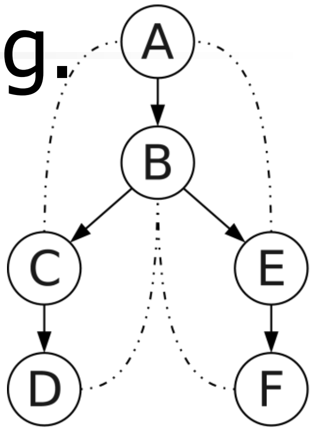
Overview

- Placed 1st in all three MPE tracks.
  - Close competition, congratulations to runner-ups!
- Baseline: AND/OR Branch-and-Bound with mini-bucket heuristic .
  - 3rd place for MPE at UAI 2010 Evaluation.
- Our solver DAOOPT is AOBB “on steroids”:
  - Several enhancements / extensions.
    - All useful in themselves, but hard to quantify.
- Source code available online:
  - **`http://github.com/lotten/daoopt`**

AND/OR Branch-and-Bound

- Problem decomposition and caching.
- Mini-bucket heuristic for pruning.

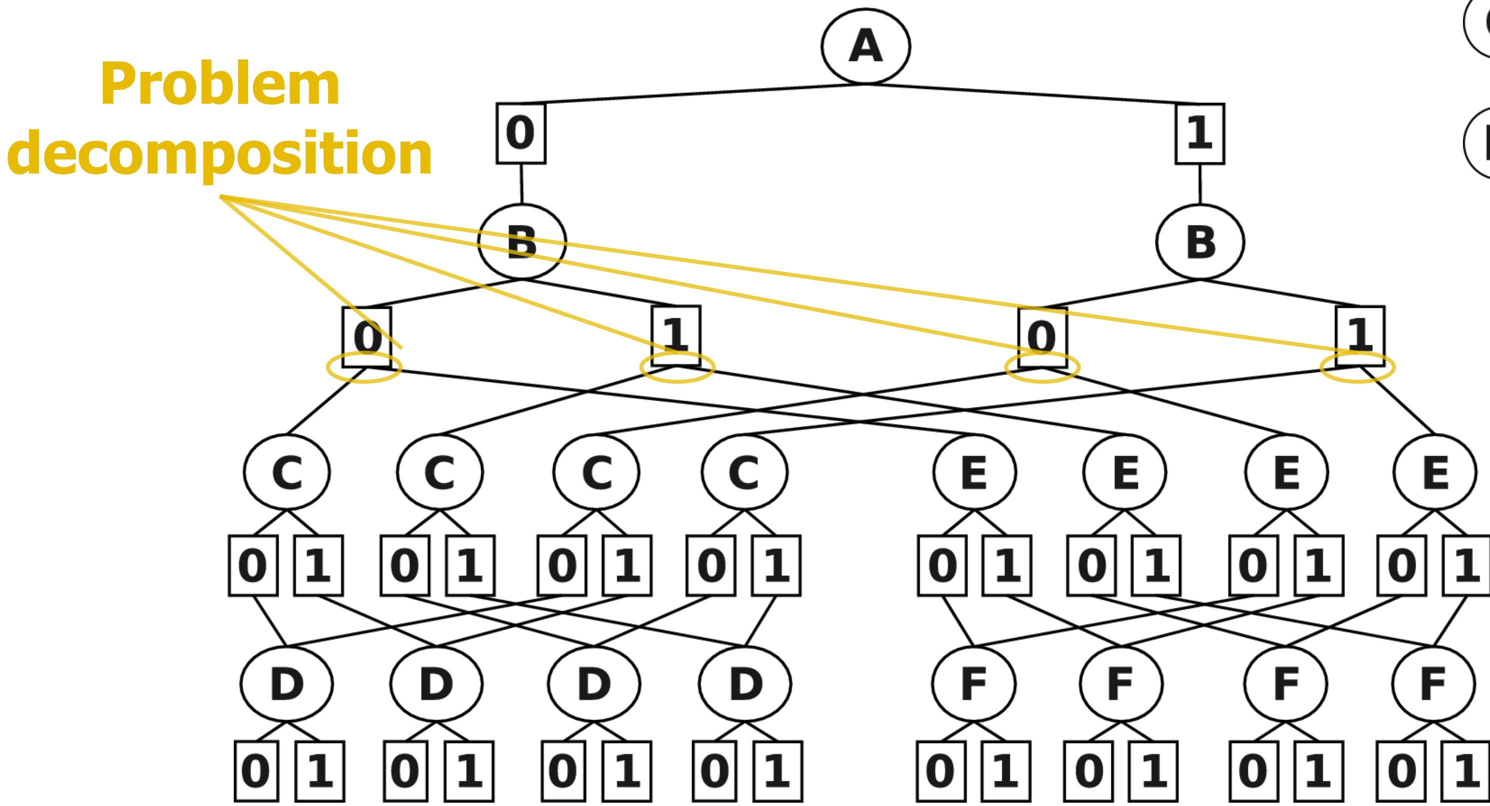


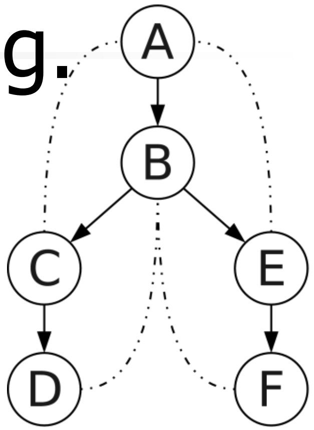


Guided by pseudo tree

AND/OR Branch-and-Bound

- Problem decomposition and caching.
  - Mini-bucket heuristic for pruning.

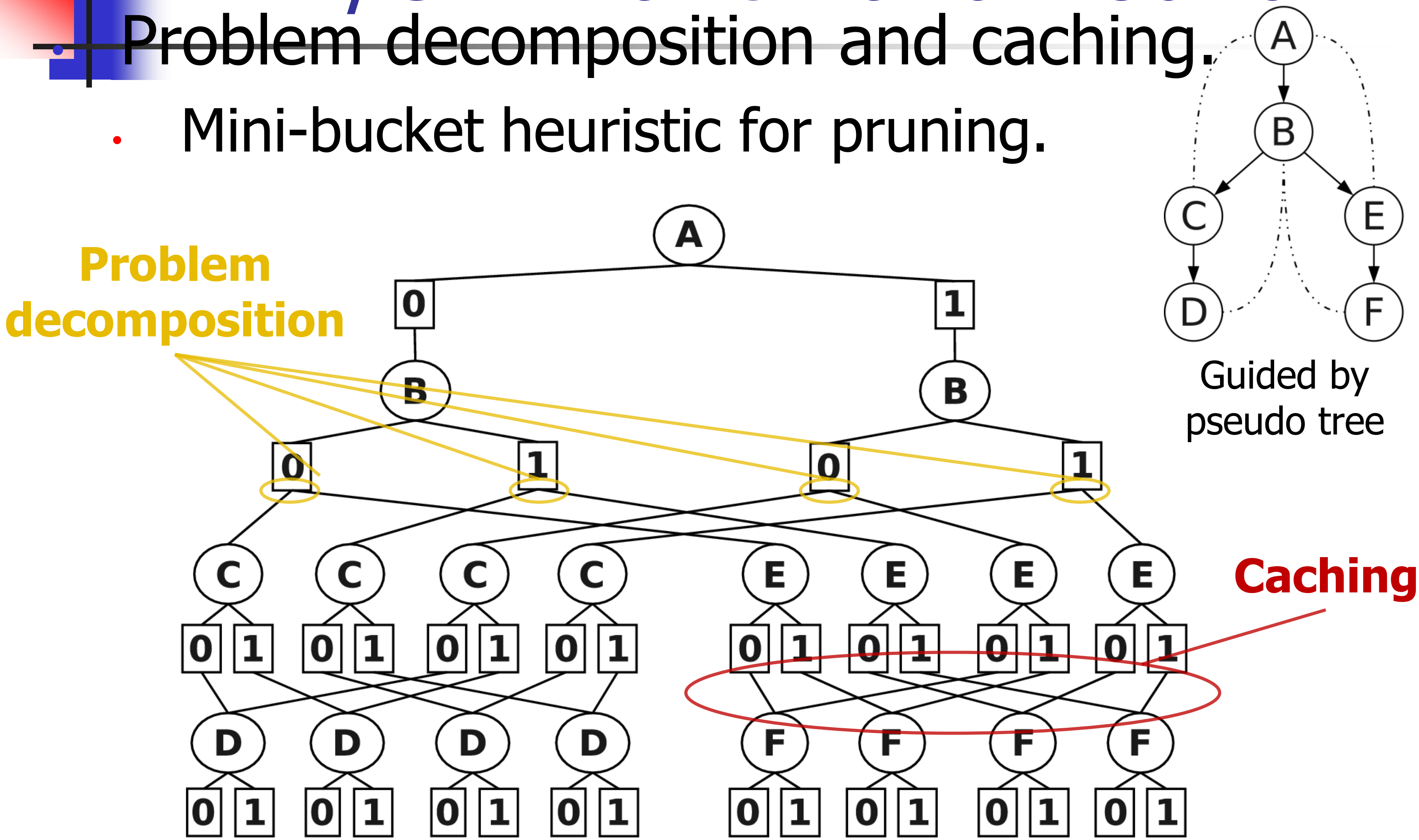




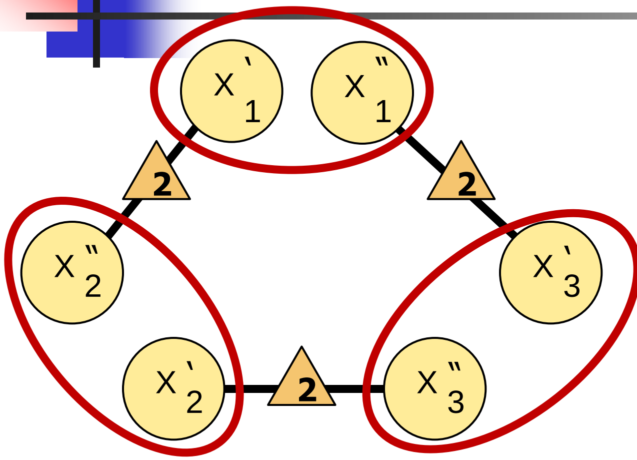
Guided by pseudo tree

AND/OR Branch-and-Bound

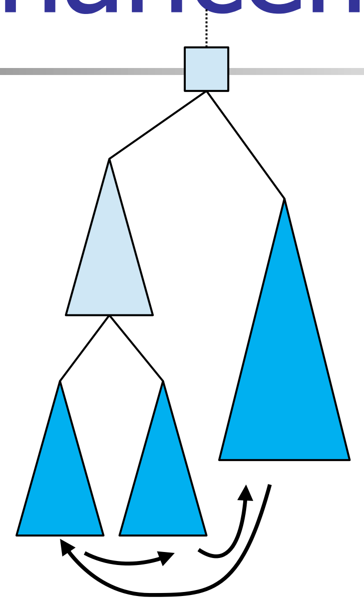
- Problem decomposition and caching.
- Mini-bucket heuristic for pruning.

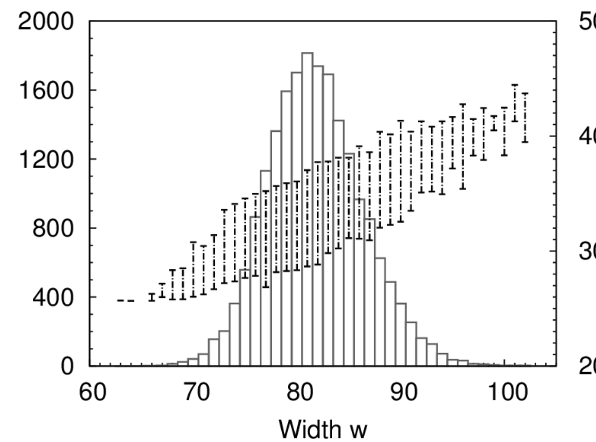


Central Enhancements



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





Cost-shifting (MPLP) Re-parametrization

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

Breadth-First Subproblem Rotation

Improved anytime performance through interleaved processing of independent subproblems.

Enhanced Variable Ordering Schemes

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

Competition Results

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

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

Join-graph decomposition

DEFINITION **1 (join-graph decompositions)** *A* join-graph decomposition *JG for* $\mathcal{M} =< X, D, F, \bigotimes, \Downarrow>$ *is a triple* $\mathcal{JG} =< G, \chi, \psi >$*, where* $G = (V, E)$ *is a graph, and* χ *and* ψ *are labeling functions which associate each vertex* $v \in V$ *with two sets,* $\chi(v) \subseteq X$ *and* $\psi(v) \subseteq F$ *such that:*

I. For each $f \in F$*, there is* exactly *one vertex* $v \in V$ *such that* $f \in \psi(v)$*, and* $scope(f) \subseteq \chi(v)$.

II. (connectedness) For each variable $X_i \in X$*, the set* $\{v \in V | X_i \in \chi(v)\}$ *induces a connected subgraph of* G*. The connectedness requirement is also called the running intersection property.*