Constraints and Probabilistic networks: a look at the interface

Rina Dechter

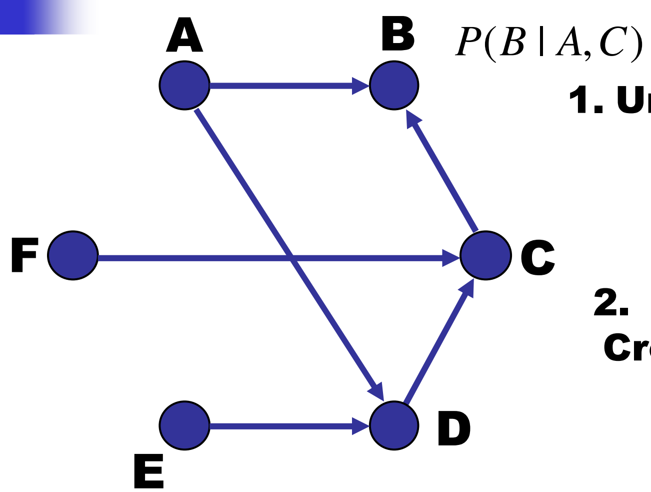
Information and Compute Science

University of California, Irvine

Collaboration with : Kalev Kask, Robert Mateescu, David Larkin

Probabilistic vs Deterministic networks

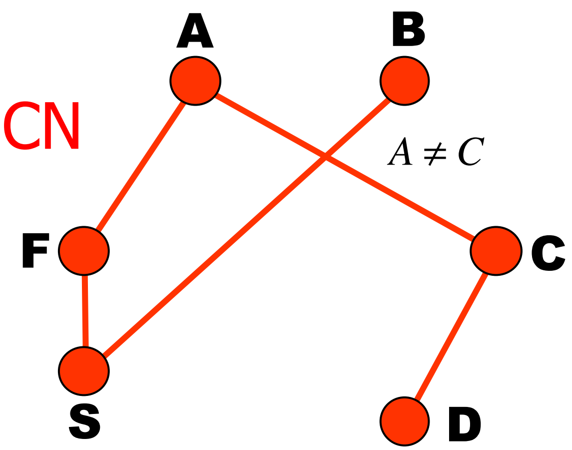
BN



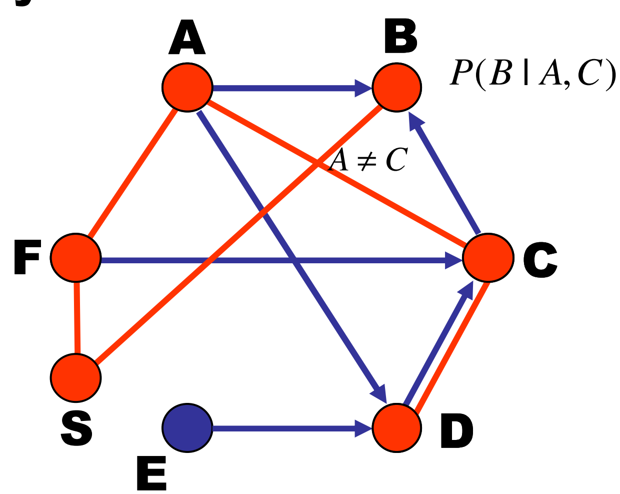
1. Understanding

2. Algorithm Cross Fertilization

CN



3. Hybrids: combine? Subsume?



Semantic?
Algorithms?

Graphical models: probabilistic and deterministic

- Bayesian networks: Directed, probabilistic
- Markov networks: undirected, probabilistic
- Constraint networks: undirected, deterministic
- What is the principle differences?
  - What does directionality mean?
  - What do the numbers mean?
- Should we develop a new model that incorporate several functionalities?
- Focus: Constraint networks vs Bayesian networks

Constraint Satisfaction

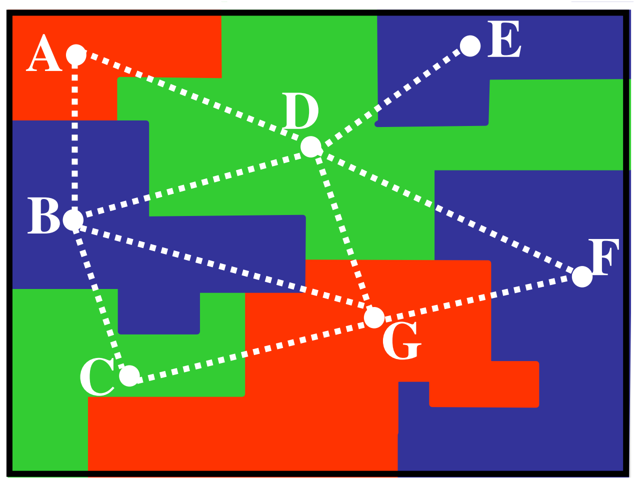
Example: map coloring

Variables (X) - countries (A,B,C,etc.)

Values (D) - colors (e.g., red, green, yellow)

Constraints (C): **A ≠ B, A ≠ D, D ≠ E**, *etc.*

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



Semantics: set of all solutions

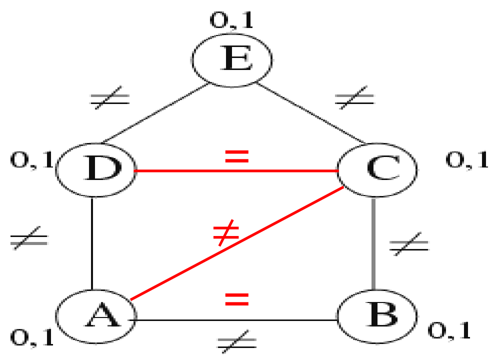
Primary task: find a solution

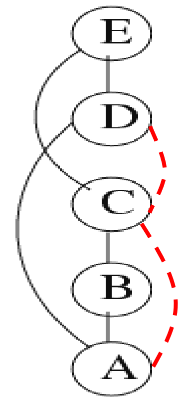
Two primary approaches

- Inference:
  - Variable elimination, tree-clustering,
- Search:
  - Backtracking, conditioning
- Hybrids of search and inference

Bucket Elimination

Variable elimination





Bucket E: $E \neq D,\ E \neq C$ → **D = C**

Bucket D: $D \neq A$ → **A ≠ C**

Bucket C: $C \neq B$ → **B = A**

Bucket B: $B \neq A$ → **contradiction**

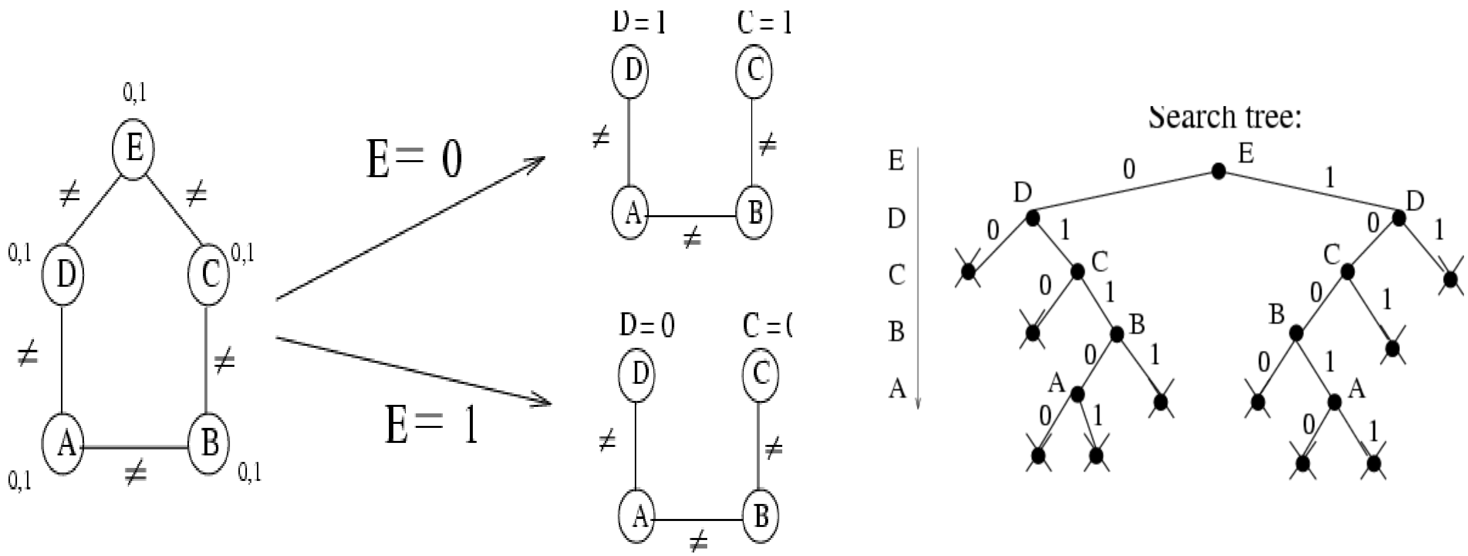
Bucket A:

Complexity : $O(n\,exp(w^{*}))$

w^{*} - ***induced width, tree - width***

trees are easy : w* = 1

The Idea of Conditioning

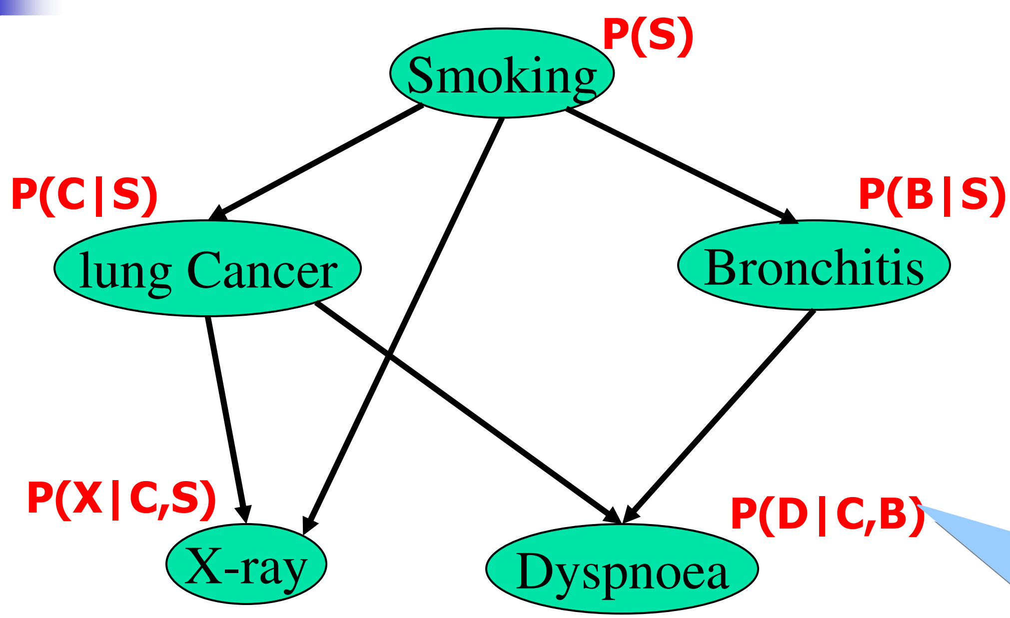


Complexity : ***exponential time, linear space***

Refined complexity : a) exponential in cycle - cutset size

b) in depth of dfs tree

Probabilistic Networks



$BN = (G, \Theta)$

CPD:

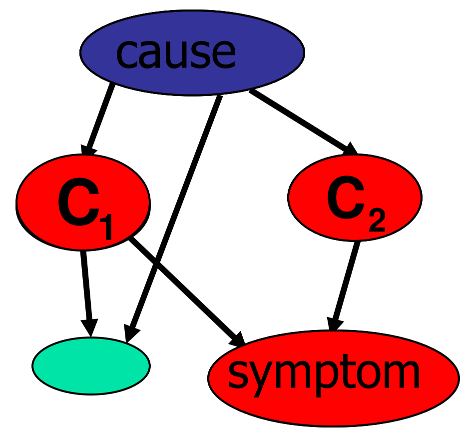
C	B	D=0	D=1
0	0	0.1	0.9
0	1	0.7	0.3
1	0	0.8	0.2
1	1	0.9	0.1

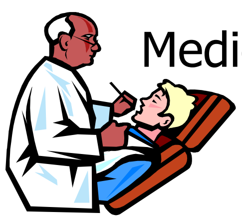
$$P(S, C, B, X, D) = P(S)\, P(C|S)\, P(B|S)\, P(X|C,S)\, P(D|C,B)$$

Conditional Independencies ➡ Efficient Representation

What are they good for?

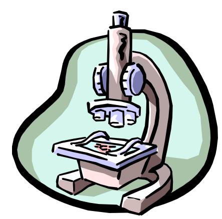
- Diagnosis: P(cause|symptom)=?
- Prediction: P(symptom|cause)=?
- Classification: $\max_{class}$ P(class|data)
- Decision-making (given a cost function)





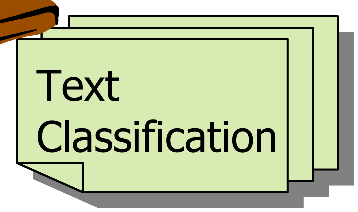
Medicine

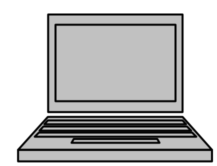
Speech recognition



Bio-informatics

Stock market

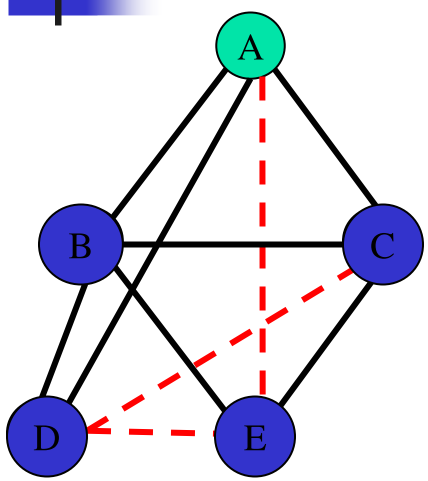




Computer troubleshooting

CP-2002

Belief updating: P(X|evidence)=?



"Moral" graph

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

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

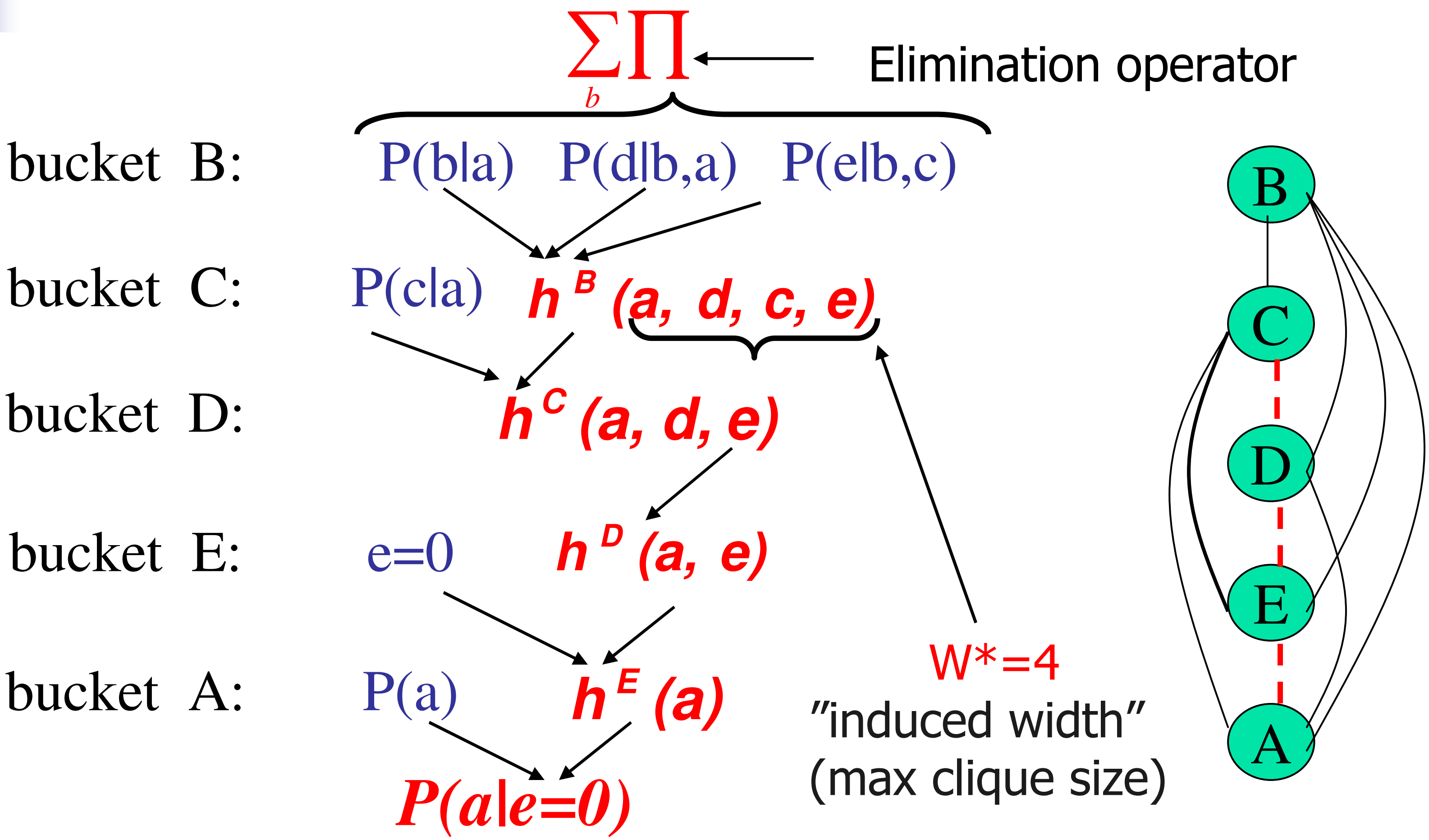
$$P(a)\sum_{e=0}\sum_{d}\sum_{c}P(c|a)\sum_{b}P(b|a)P(d|b,a)P(e|b,c)$$

Variable Elimination

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

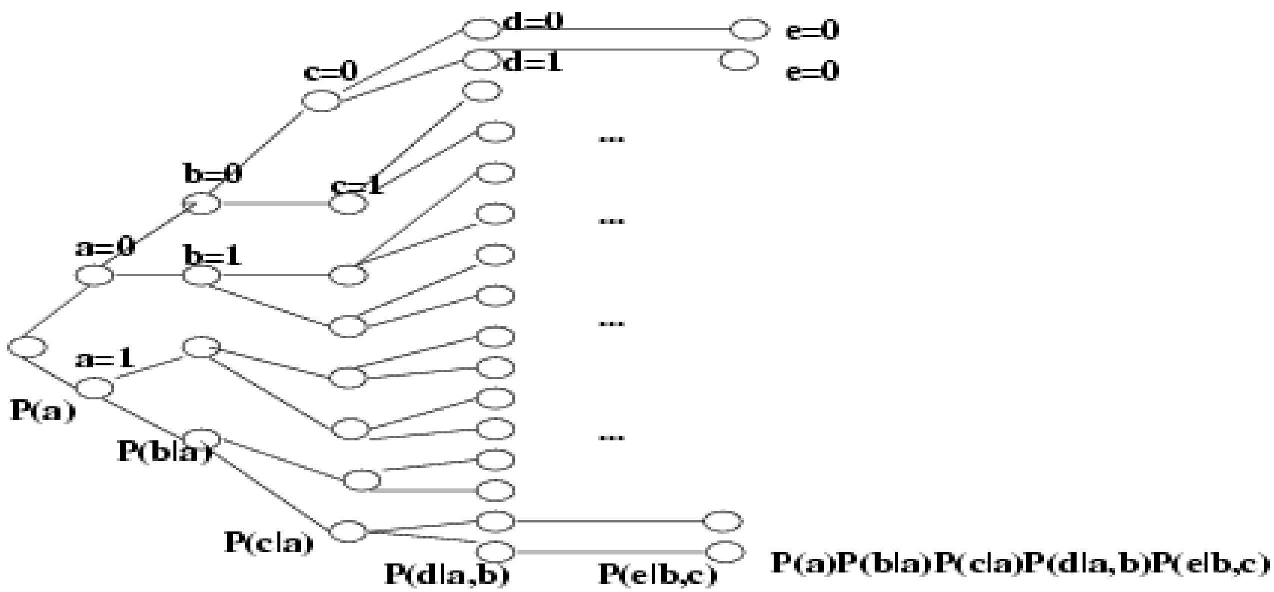
Bucket elimination

Algorithm *elim-bel* (Dechter 1996),
Join-tree clustering (Spigelhalter et. Al. 1988)



Conditioning generates the probability tree

$$P(a, e=0) = P(a)\sum_b P(b \mid a)\sum_c P(c \mid a)\sum_b P(d \mid a, b)\sum_{e=0} P(e \mid b, c)$$



Complexity: exponential time, linear space
Refined complexity: exponential in loop-cutest size,
Linear space.

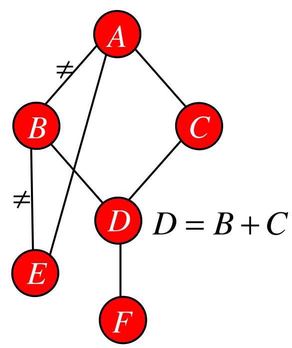
Exact Techniques: Complexity

	Search	Variable Elimination
Worst-case time	$O(\exp(n))$ $O(\exp(cutset))$ $O(\exp(dfs-depth))$	$O(n\exp(w^*))$ $w^* \leq n$
Average time	Better than worst-case	Same as worst-case
Space	$O(n)$	$O(n\exp(w^*))$ $w^* \leq n$
Output	One solution	Knowledge compilation

Queries of CN vs BN

Constraint networks

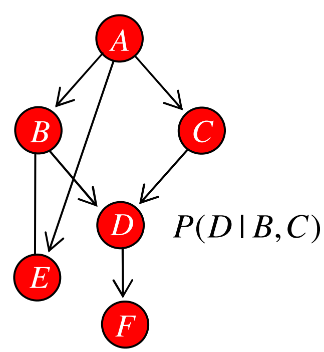
- **Is it consistent?**
- **Find solution**
  - NP-complete
- Count solutions
  - #P-complete
- unminimal const
- Solved by search
- Use constraint propagation



represents
$sol\,(A, B, C, D, E, F)$

Probability networks

- Always consistent
- Find t s.t P(t)>0
  - Easy: backtrack-free
- **Find P(X|e)?**
  - **#P-complete**
- Explicit minimal tables
- Solved by variable elimination
- No propagation

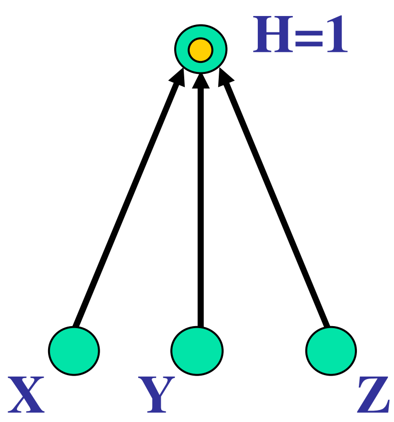


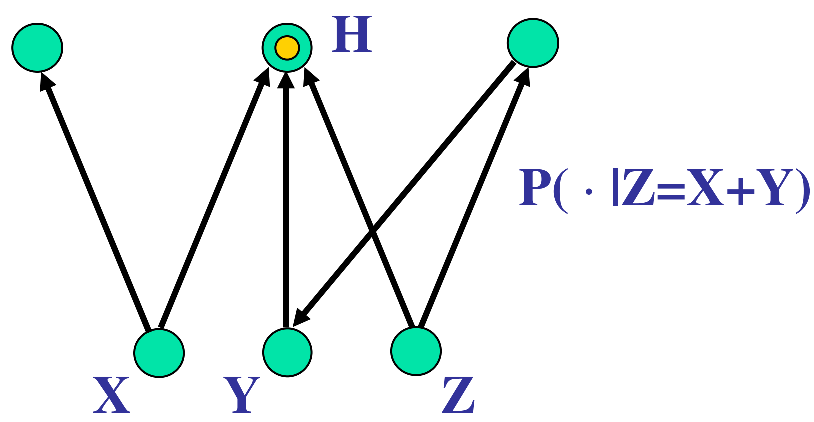
represents
$P(A, B, C, D, E, F)$

Constraints as CPTs

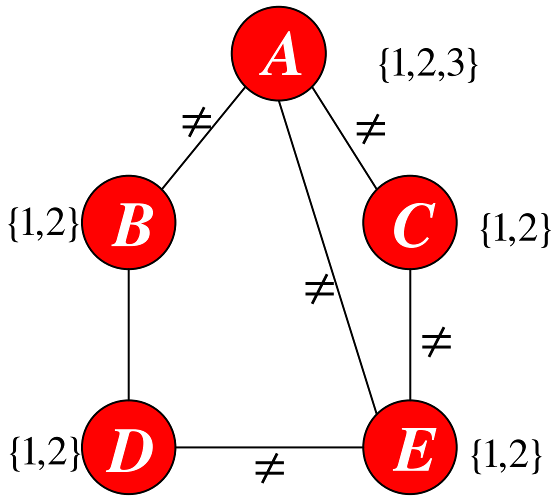
Express each constraint as a probability table using a new child variable.

- X+Y = Z expressed as
- P(H | X,Y,Z) = 1 iff Z = X+Y

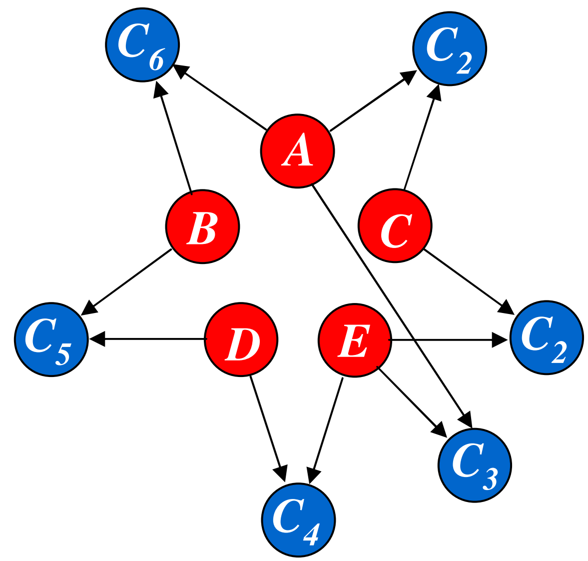




Modeling C*N* as B*N*



$\Rightarrow$



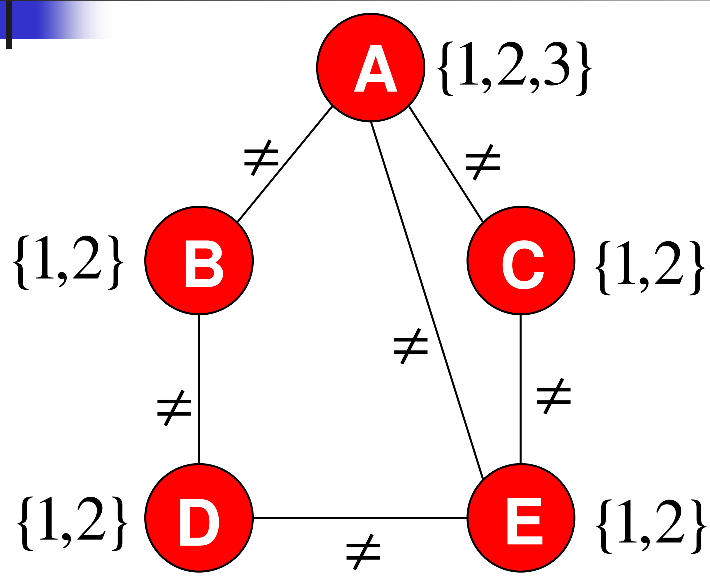
Is the network consistent? $P(C_1,\dots,C_6) > 0)\,?$

Find a solution. $\mathit{find}\;\; x, s.t., P(x \mid C_1,\dots,C_6) > 0$

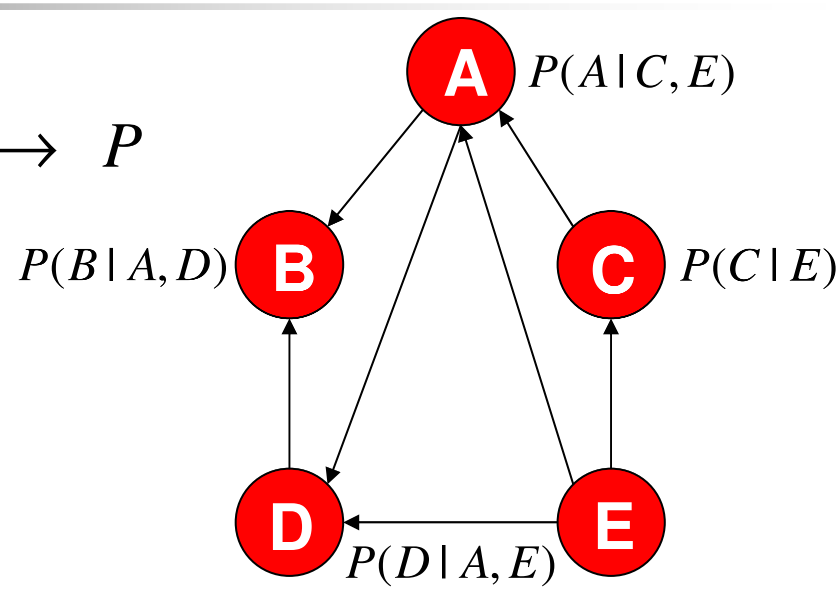
$sol\;\;(A,B,C,D,E)$ $\sim \prod_{marginal} P(A,C,D,D,E,C_1,\dots,C_7)$

A variable-elmination conversion

(eliminates new variables, and more…)



$$R \rightarrow P$$



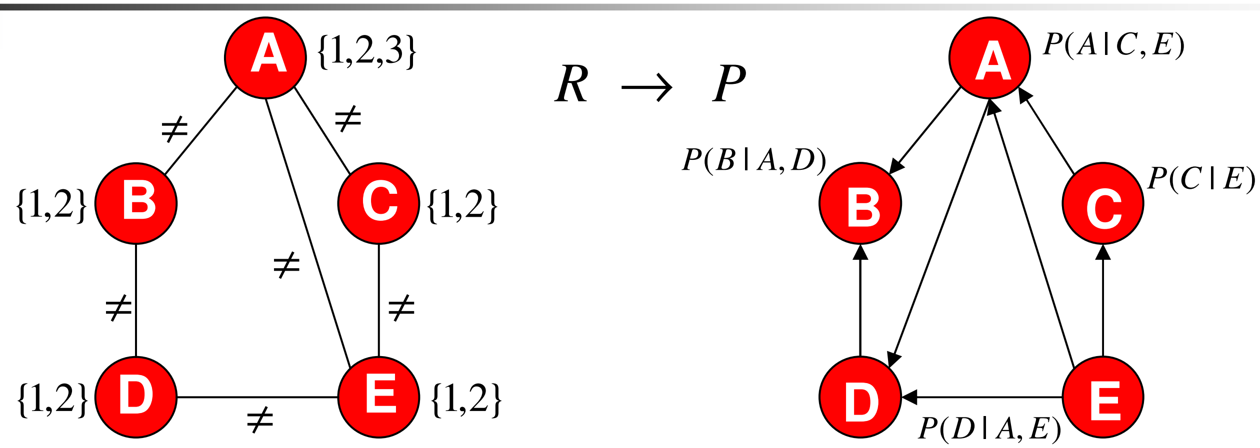
$$P(B|A,D) = \frac{R(A,B) * R(B,D)}{\sum_B R(A,B) * R(B,D)}$$

$$P(x_1,\ldots,x_n) = \frac{1}{\#\text{sol}} \qquad \text{if } (x_1,\ldots,x_n) \text{ is a solution}$$

Complexity: exp(w*)

But the network is already easy

A variable-elmination conversion (into a pure BN)



$B:\ R_{AB}\quad R_{BD} \;\rightarrow\; B: P(B \mid A,D)$

$D:\ R_{DE}\quad R_{AD}\quad R'_{AD} \;\rightarrow\; D: P(D \mid A,E)$

$A:\ R_{AC}\quad R_{AE}\quad R'_{ED} \;\rightarrow\; A: P(A \mid C,E)$

$C:\ R_{CE}\quad R'_{CE} \;\rightarrow\; C: P(C \mid E)$

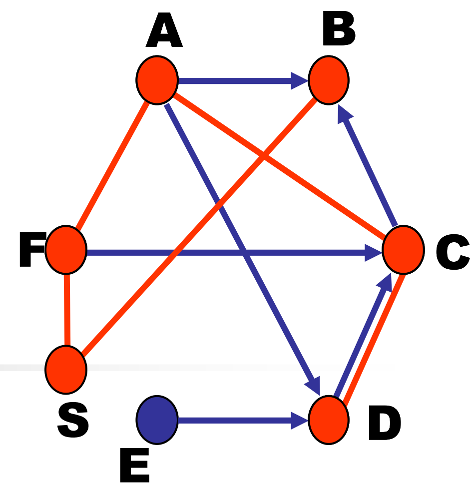
$E:\quad R'_{E} \qquad E: P(E)$

$$P(B \mid A,D) = \frac{R(A,B) * R(B,D)}{\sum_B R(A,B) * R(B,D)} \qquad R(A,D) = \sum_B R(A,B) * R(B,D)$$

What is the point?

- Understanding, cross-fertilization, hybrids
- Different intended semantics:
  - BNs models **what is** (nature, the world)
    - → consistent
  - CNs model **what is desired** by human:
    - plans, intervention, decision-making processes → often inconsistent
- Reasons for hybrids:
  - Modeling agents behavior require both (games, treatments of diseases, etc)
  - Human actions has consequences on the world.
  - Actions are constrained
  - Exploiting computational properties

Hybrid networks



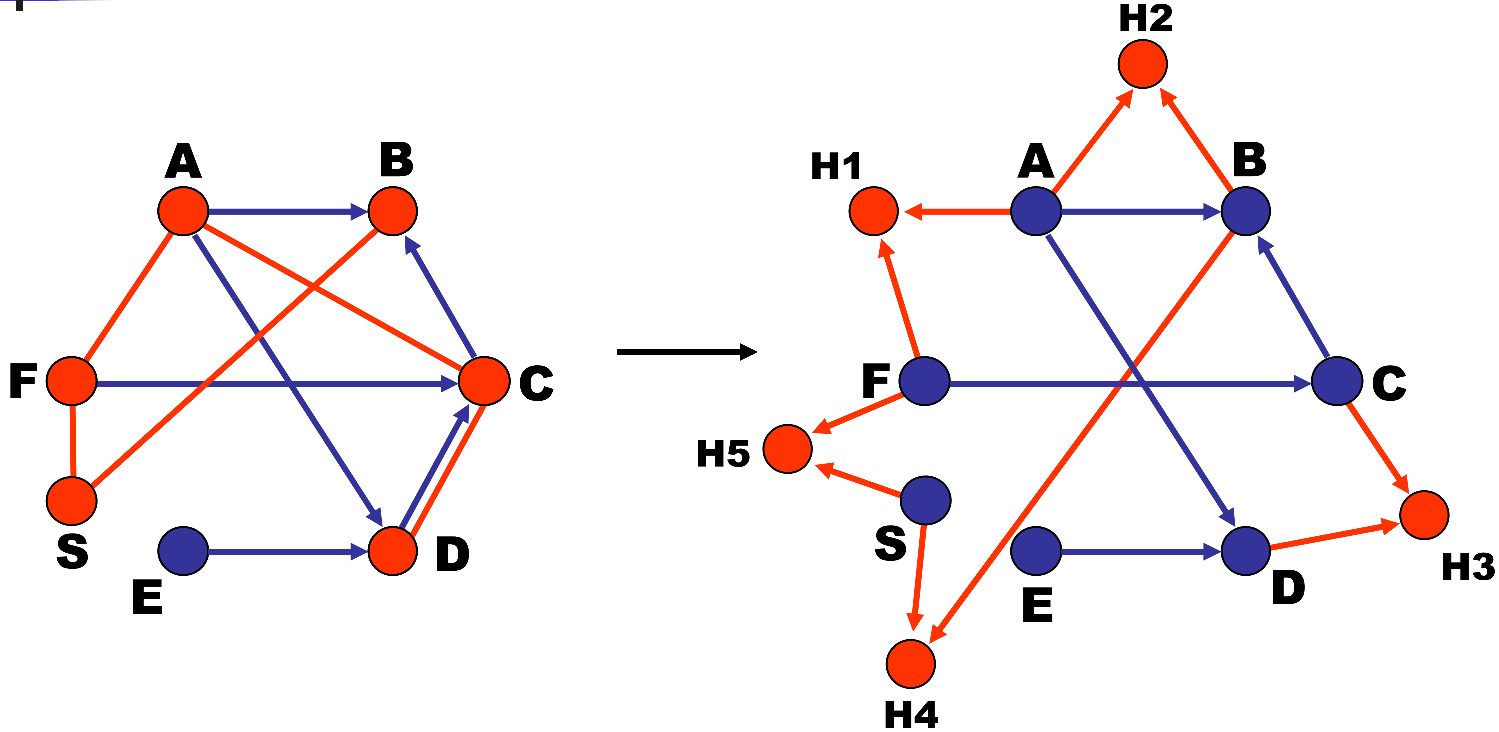
- Hybrid belief networks: <BN,CN>
  - BN= (X,D,G,P), CN= (X, D, C)
- Semantics:

$$P_h(\bar{x}) = \begin{cases} P_{BN}(\bar{x}) & if\ \bar{x}\ in\ sol(CN) \\ 0 & otherwise \end{cases}$$

$$= P_{BN}(\bar{x} \mid \bar{x} \in sol(CN))$$

- Queries: $P(x_1 \mid CN) = ?,\ P(CN\ consistent) = ?$
- **Processing: express as conditional pure Bayesian network with hidden variable (approach 1), or**
- **Variable-eliminate hidden variables to get a pure BN (approach 2)**

How to process Hybrid BN,CN network?



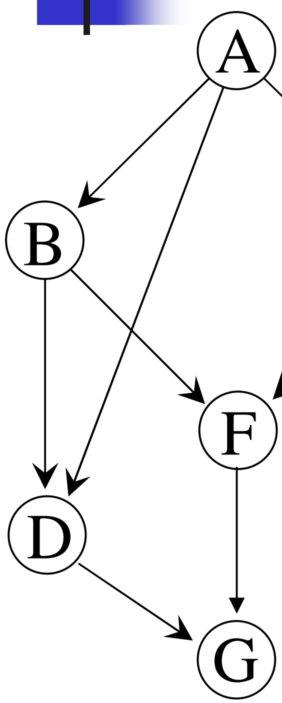
$$P_H(\bar{x}) = P_T(\bar{x}, h_1, \ldots, h_5 \mid h_1 = 1, \ldots, h_5 = 1)$$

Should we convert to pure BN ?

Exploit Constrains properties ?

Hybrid Processing Beliefs and Constraints

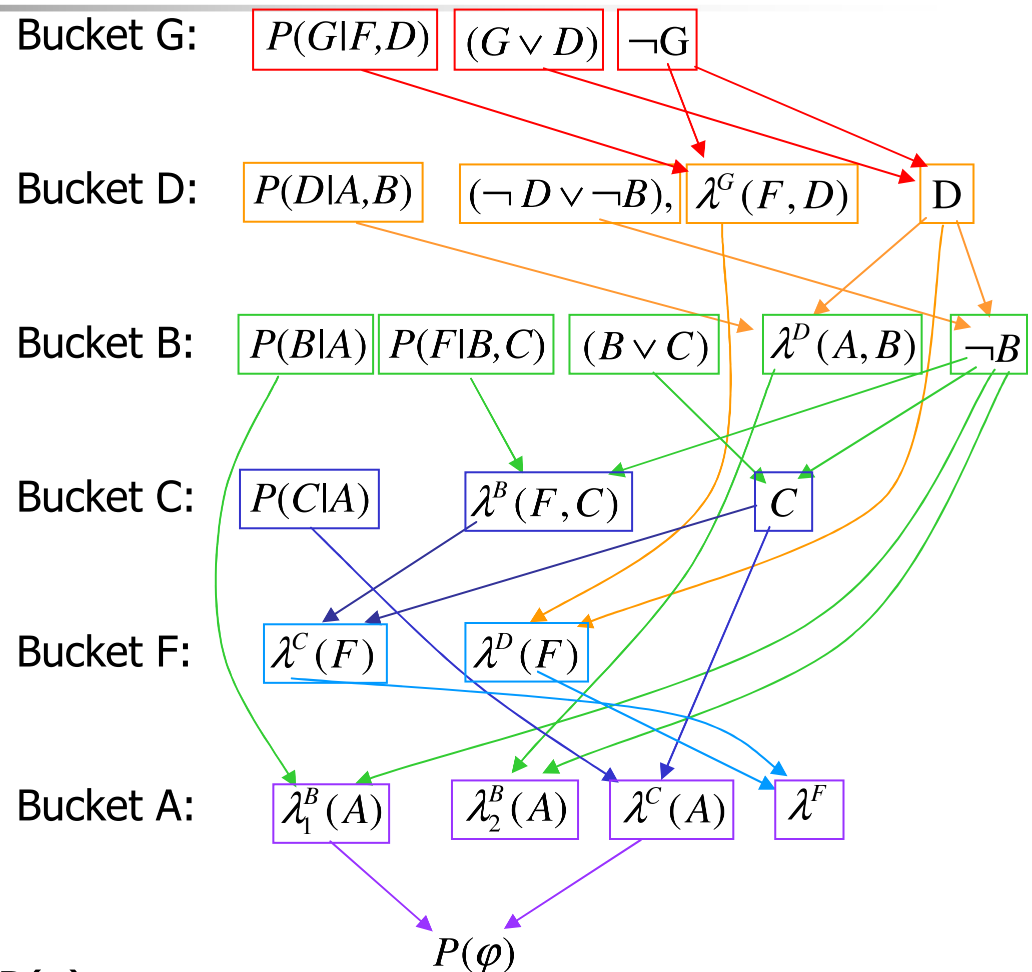
Trace of Elim-CPE: Evaluating a cnf query



$\varphi = (G \vee D) \wedge (\neg D \vee \neg B) \wedge (B \vee C)$

$P(\varphi) = ?$

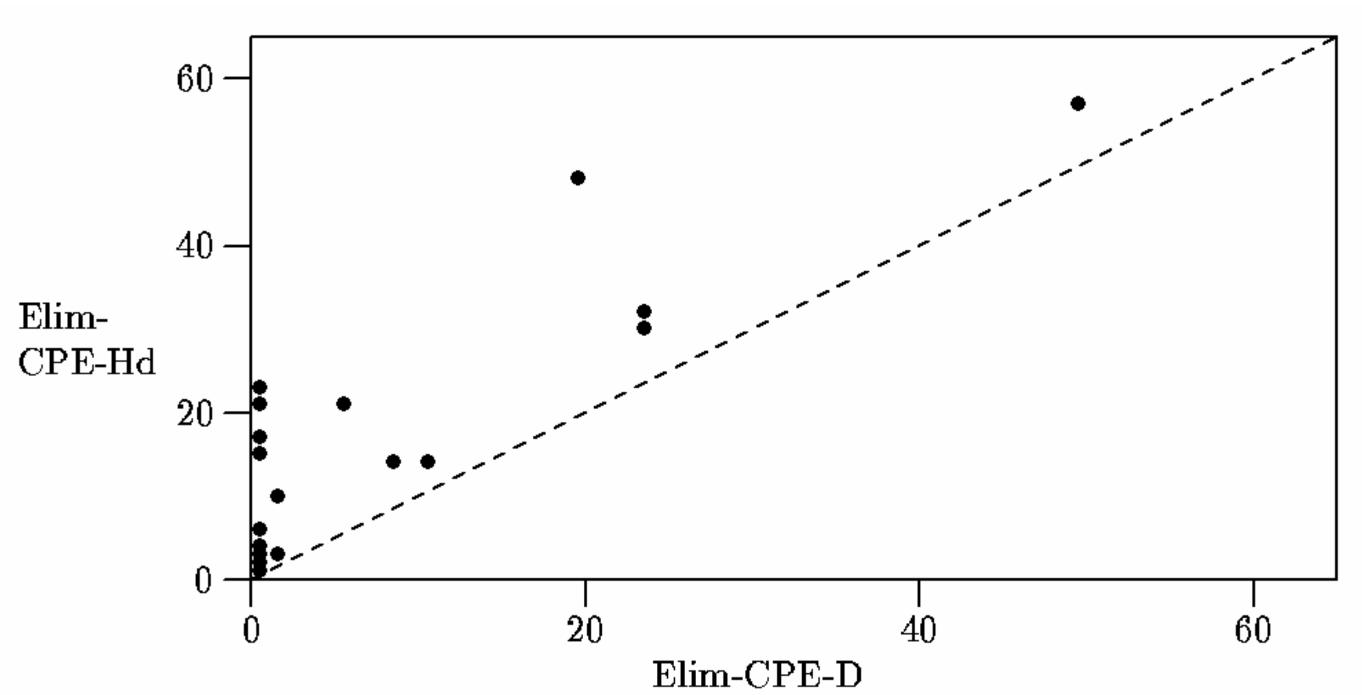
$P(x \mid \varphi) = ?$



Belief network P(g,f,d,c,b,a)
=P(g|f,d)P(f|c,b)P(d|b,a)P(b|a)P(c|a)P(a)

Elim-CPE-D on Insurance network

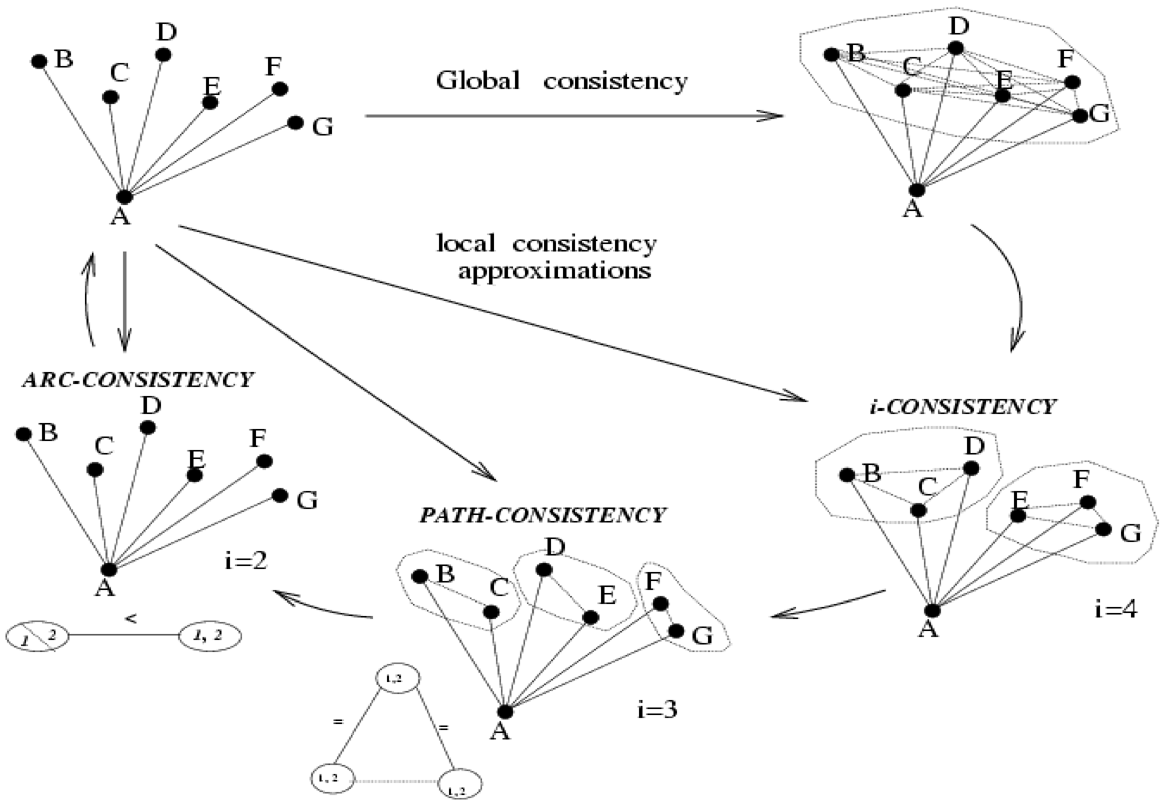
19 instances with Insurance network. 20 relations, arity 3, tightness 25 %, 5 evidence nodes.



Overview

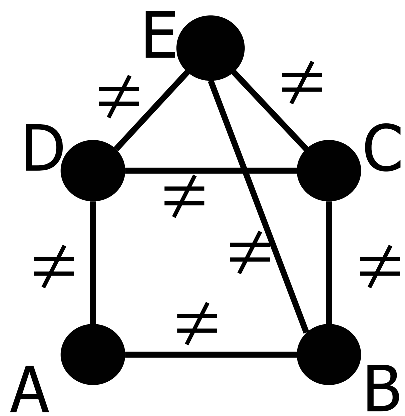
1. Preliminaries
2. Observing constraint vs probabilistic networks.
3. **Importing constraint propagation ideas into probabilistic inference**
4. Hybrid processing of constraints and probabilities
5. Random sampling of constraint solutions
6. Conclusions

From Global to Local Consistency



Propagation Impossible unless semi-ring idempotent operator (Bistareli, Rossi, Montanari, 1997)

Directional i-consistency



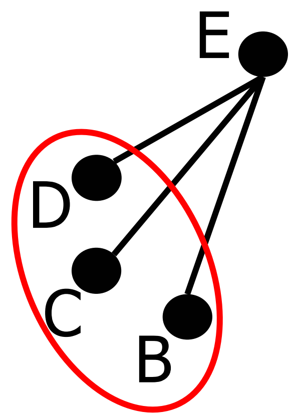
E : E $\neq$ D, E $\neq$ C, E $\neq$ B

D : D $\neq$ C, D $\neq$ A

C : C $\neq$ B

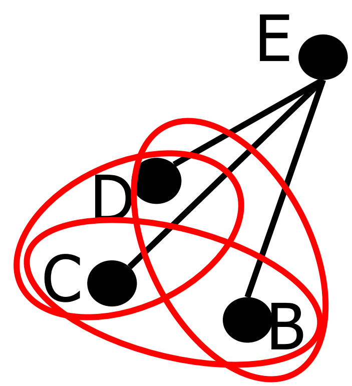
B : A $\neq$ B

A :



Adaptive

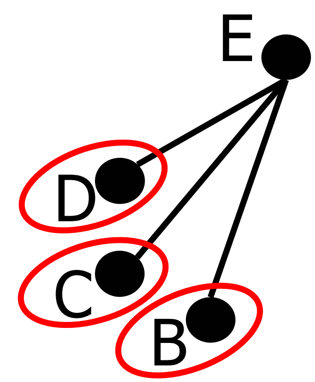
R_{DCB}



d-path

R_{DC} , R_{DB}

R_{CB}



d-arc

R_D

R_C

R_D

The idea of Mini-bucket MPE task (Dechter and Rish 1997)

Split a bucket into mini-buckets =>bound complexity

$$\mathbf{bucket\ (X)} = \{ \mathbf{h}_1, \ldots, \mathbf{h}_r, \mathbf{h}_{r+1}, \ldots, \mathbf{h}_n \}$$

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

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

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

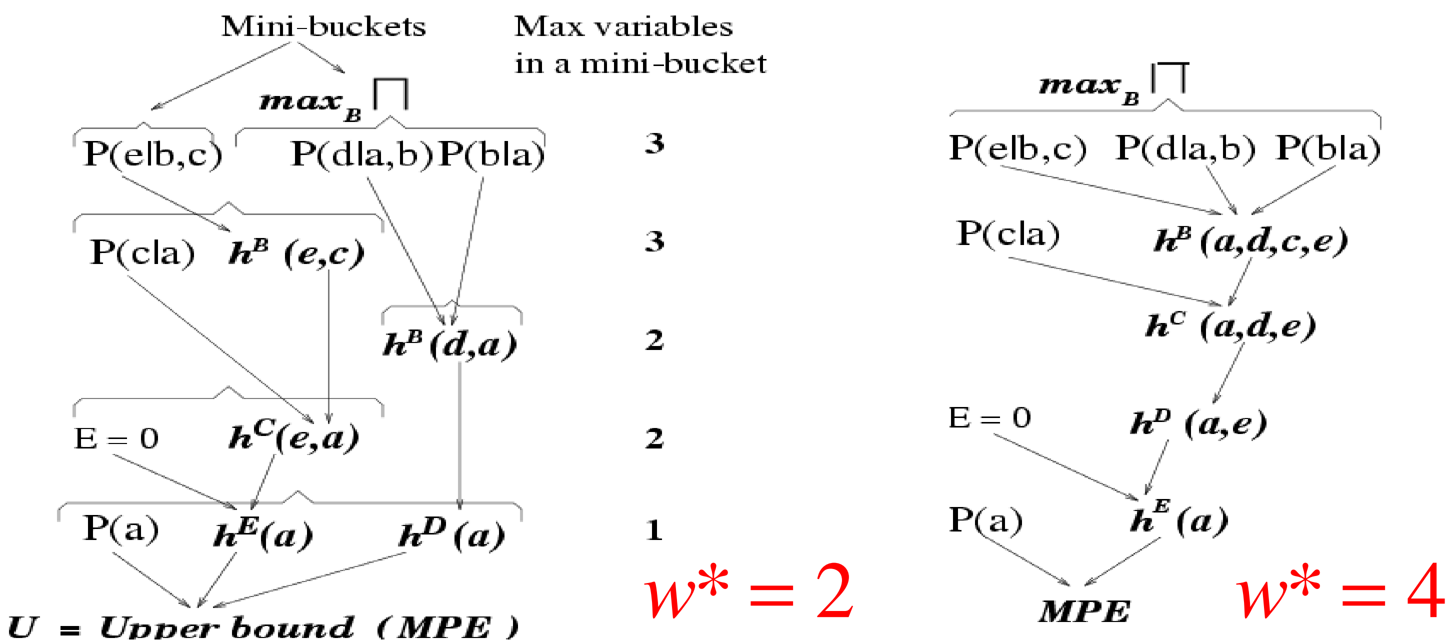
$$\mathbf{h^X \leq g^X}$$

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

Mini-bucket-mpe(i)

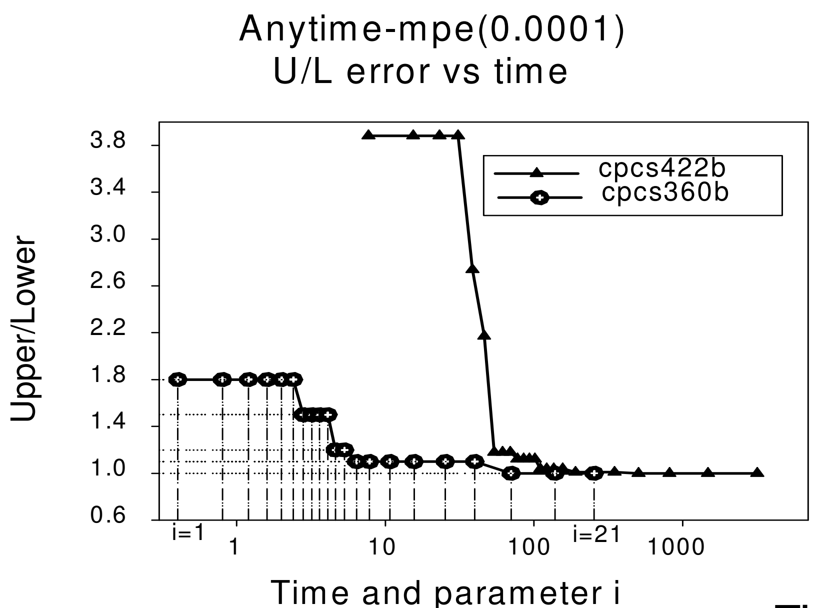
- Input: i – max number of variables allowed in a mini-bucket
- Output: [lower bound (P of a sub-optimal solution), upper bound]

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



CPCS networks – medical diagnosis (noisy-OR model)

Test case: no evidence

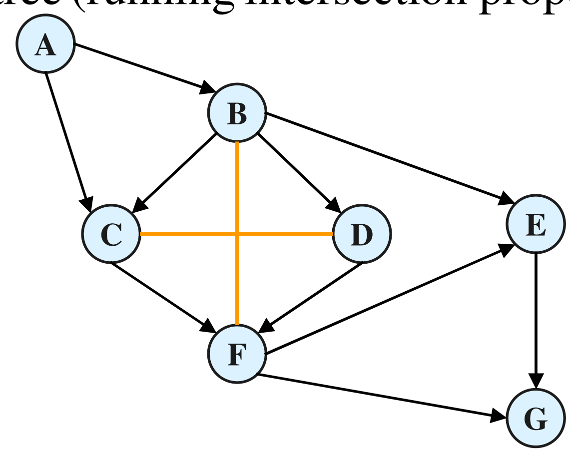


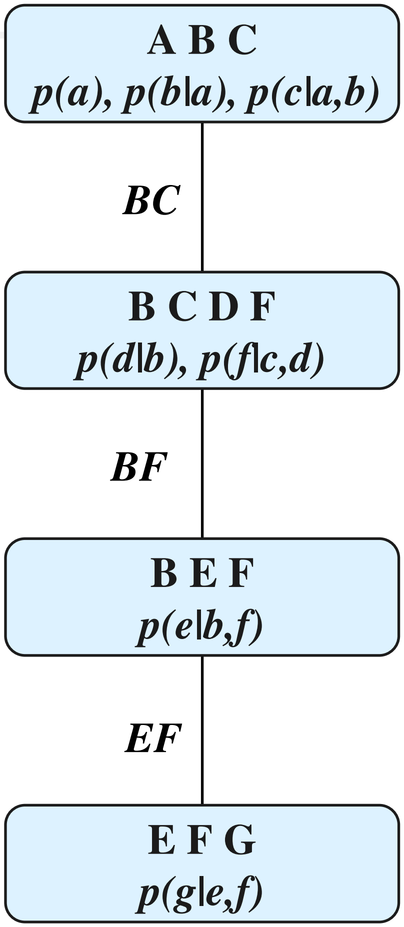
Algorithm	Time (sec) cpcs360	cpcs422
elim-mpe	115.8	1697.6
anytime-mpe(ε), $\varepsilon = 10^{-4}$	70.3	505.2
anytime-mpe(ε), $\varepsilon = 10^{-1}$	70.3	110.5

Tree decompositions

A *tree decomposition* for a belief network $BN = < X, D, G, P >$ is a triple $< T, \chi, \psi >$, where $T = (V, E)$ is a tree and χ and ψ are labeling functions, associating with each vertex $v \in V$ two sets, $\chi(v) \subseteq X$ and $\psi(v) \subseteq P$ satisfying:

1. For each function $p_i \in P$ there is exactly one vertex such that $p_i \in \psi(v)$ and $scope(p_i) \subseteq \chi(v)$
2. For each variable $X_i \in X$ the set $\{v \in V | X_i \in \chi(v)\}$ forms a connected subtree (running intersection property)





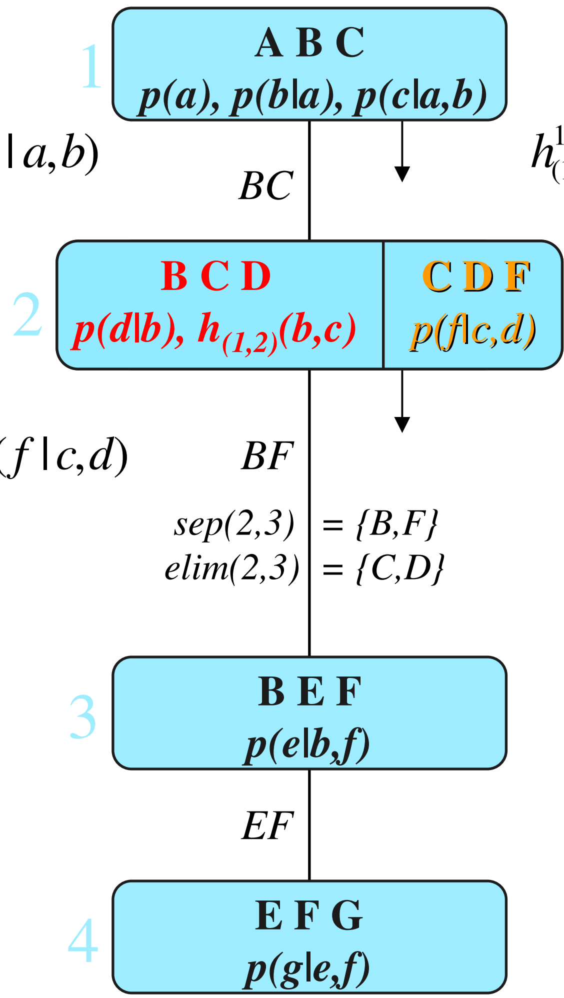
Mini-Clustering (MC) vs CTE

Cluster Tree Elimination

Mini-Clustering, i=3

$$h_{(1,2)}(b,c) = \sum_{a} p(a) \cdot p(b \mid a) \cdot p(c \mid a,b)$$

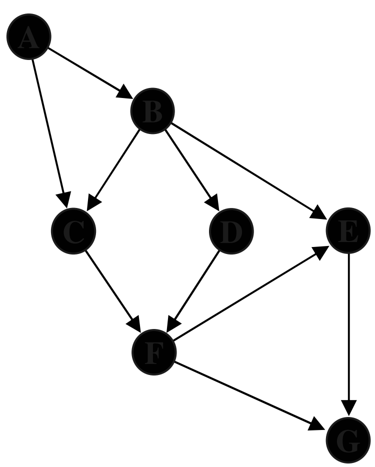
$$h^{1}_{(1,2)}(b,c) = \sum_{a} p(a) \cdot p(b \mid a) \cdot p(c \mid a,b)$$



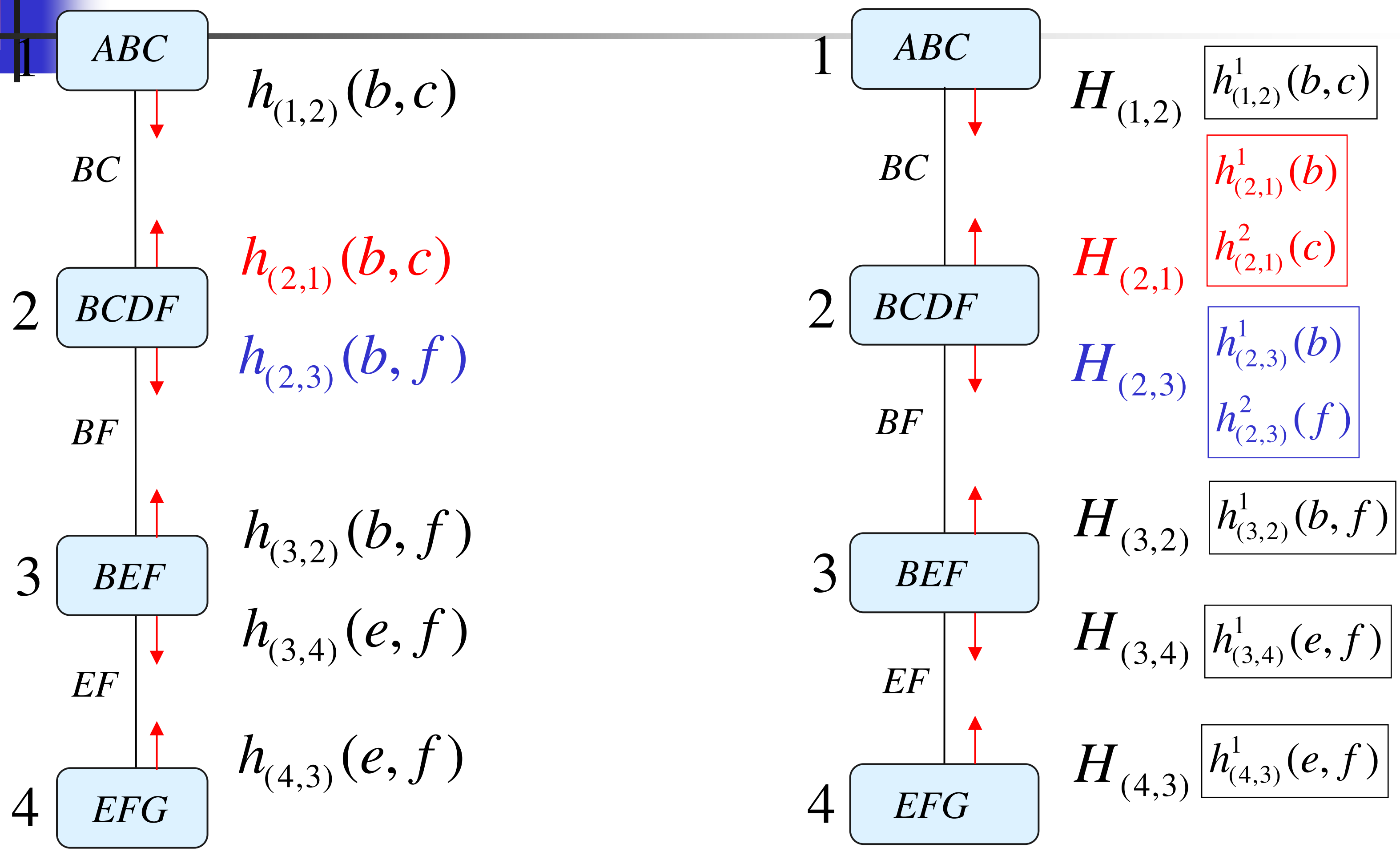
$$h_{(2,3)}(b,f) = \sum_{c,d} p(d \mid b) \cdot h^{1}_{(1,2)}(b,c) \cdot p(f \mid c,d)$$

$$h^{1}_{(2,3)}(b) = \sum_{c,d} p(d \mid b) \cdot h^{1}_{(1,2)}(b,c)$$

$$h^{2}_{(2,3)}(f) = \sum_{c,d} p(f \mid c,d)$$



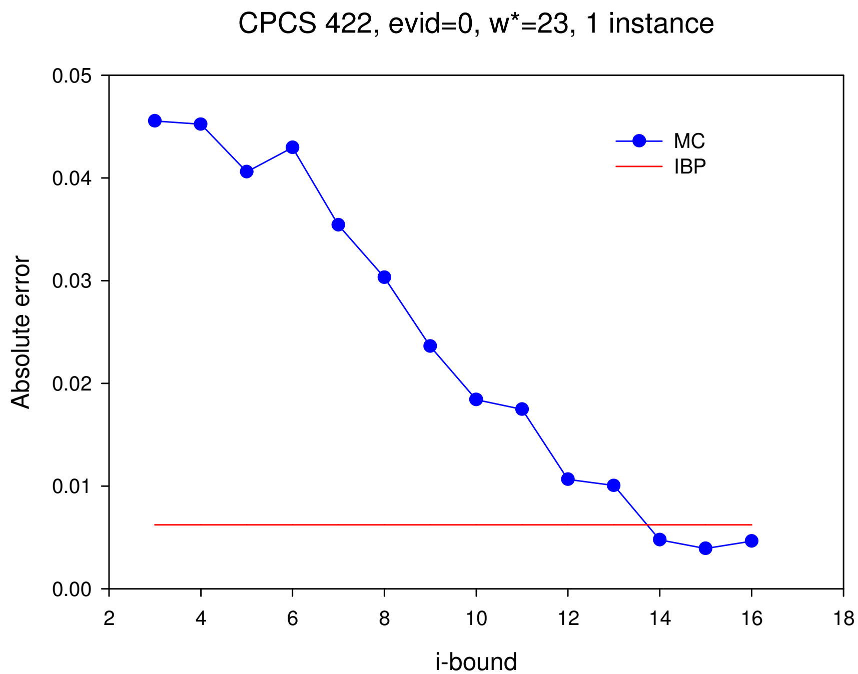
Cluster Tree Elimination vs. Mini-Clustering

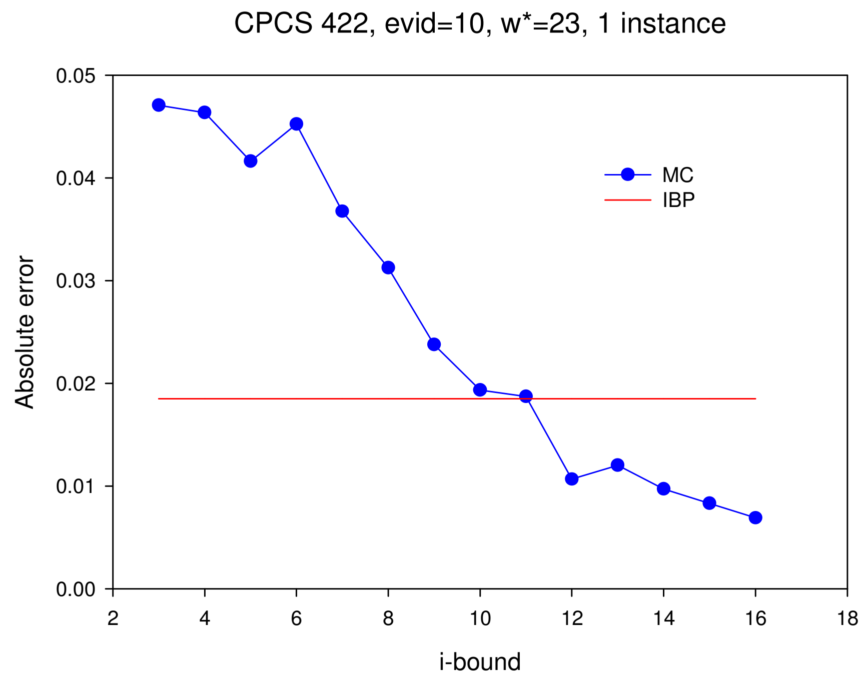


Properties of MC(z)

- MC(z) computes a bound on the joint probability P(X,e) of each variable and each of its values.
- Time & space complexity: $O(n \times hw^* \times exp(z))$
- Lower, Upper bounds and Mean approximations
- Approximation improves with z but takes more time

Performance on CPCS422 - Absolute error

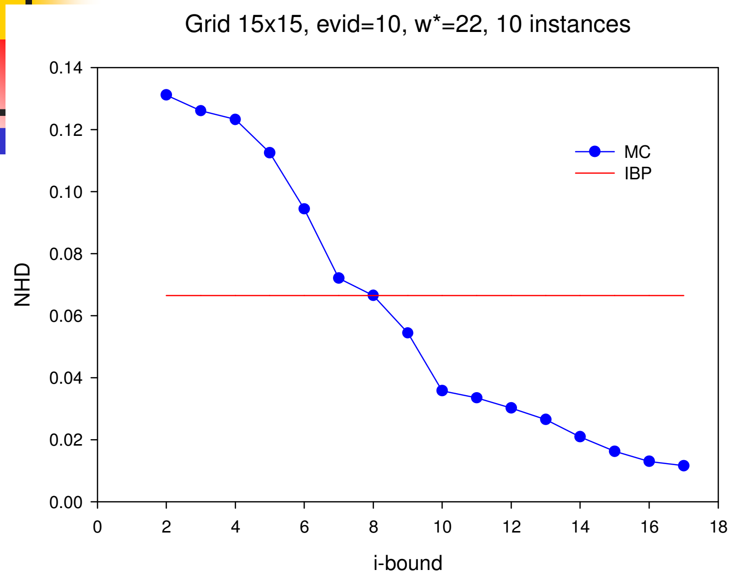


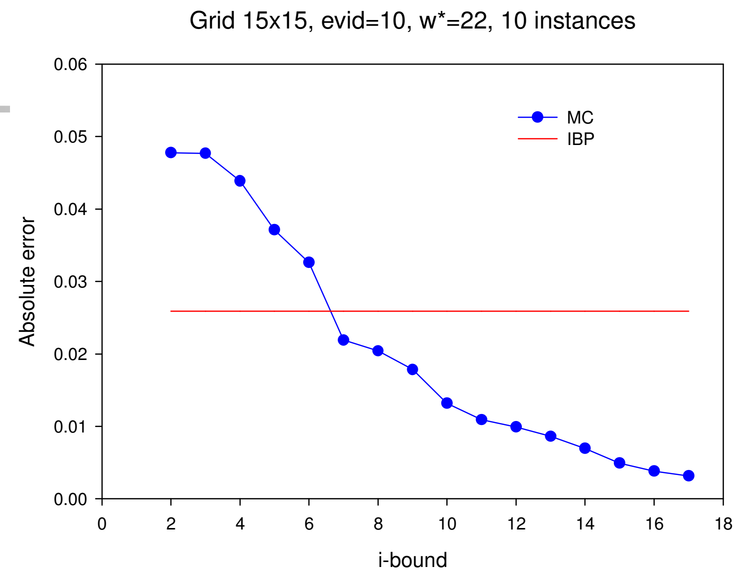


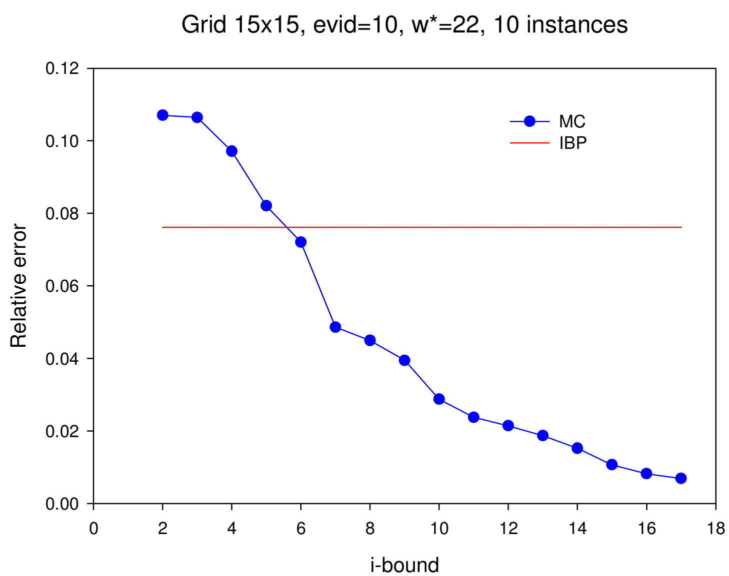
evidence=0

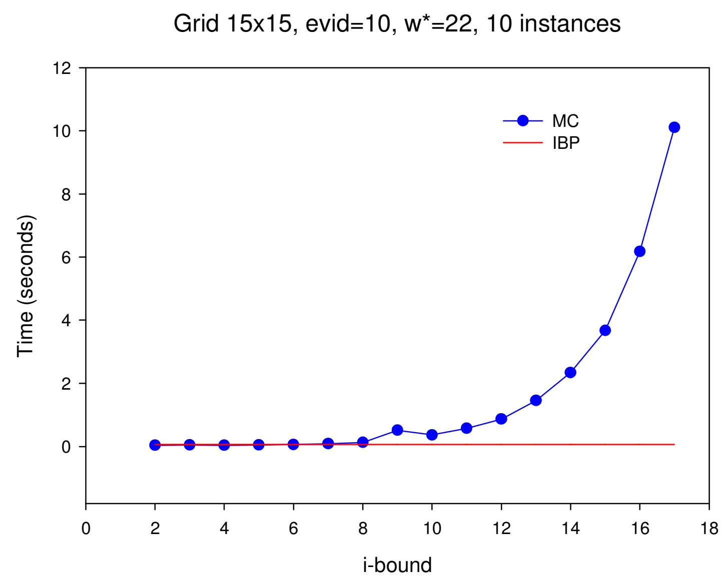
evidence=10

Grid 15x15 - 10 evidence







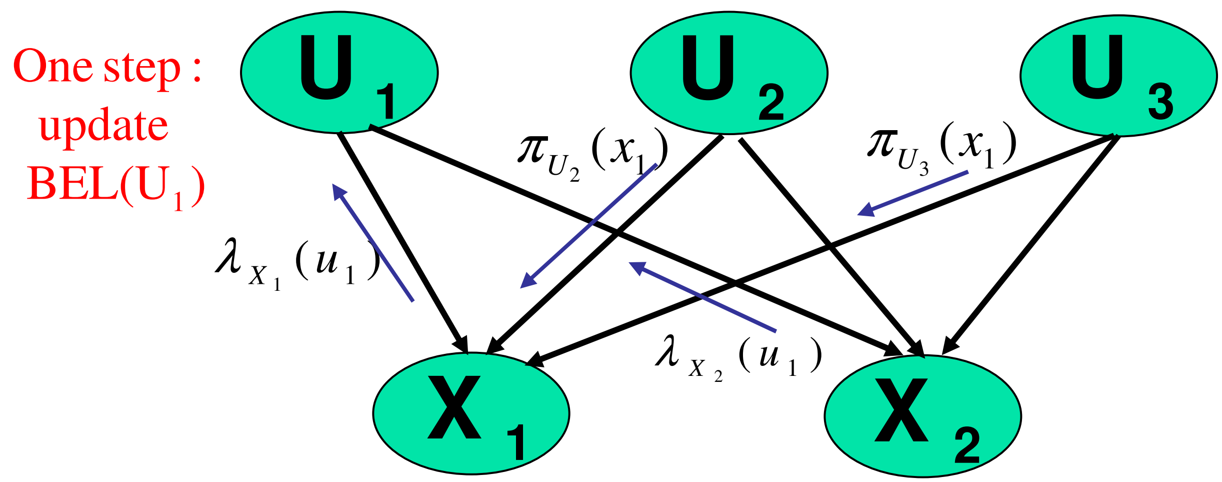


Overview

- Preliminaries
- Observing the commonalities and differences: constraint networks vs probabilistic networks.
- Importing constraint propagation ideas into probabilistic inference:
  - Mini-bucket/ mini-clustering
  - **Iterative join-graph propagation vs join-graph based propagation**
- Hybrid processing of constraints and probabilities
- Random sampling of constraint networks solutions

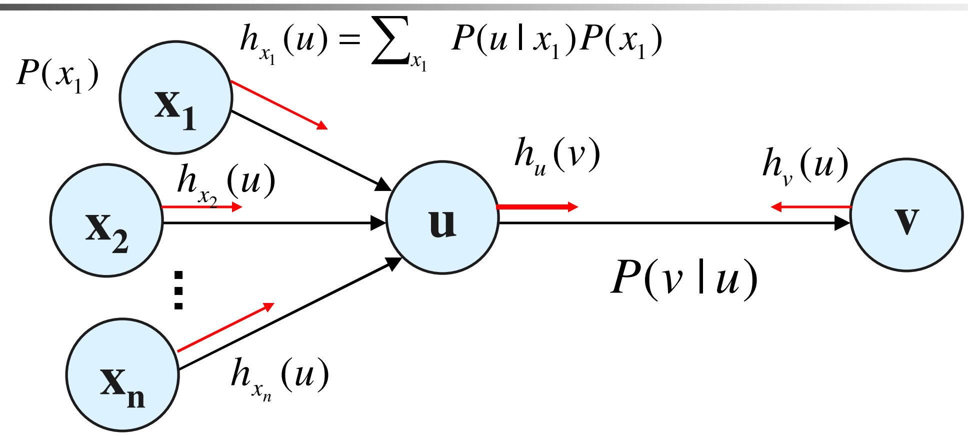
Iterative Belief Proapagation

- Belief propagation is exact for poly-trees
- IBP - applying BP iteratively to cyclic networks



- No guarantees for convergence
- Works well for many coding networks, but why? and when?

Belief Propagation



Compute the message :

$$h_u(v) = \alpha \sum_u P(v \mid u) \bullet h_{x1}(u) \bullet h_{x2}(u) \bullet, \ldots, \bullet h_{x_n}(u)$$

Exchanging by relational operators : join, project

$$h_u(v) = \Downarrow_v [R(u,v) \otimes h_{x_1}(u) \otimes h_{x2}(u) \otimes, \ldots, \otimes h_{x_n}(u)]$$

Performs arc − consistency (relational, generalized)

IBP vs arc-consistency

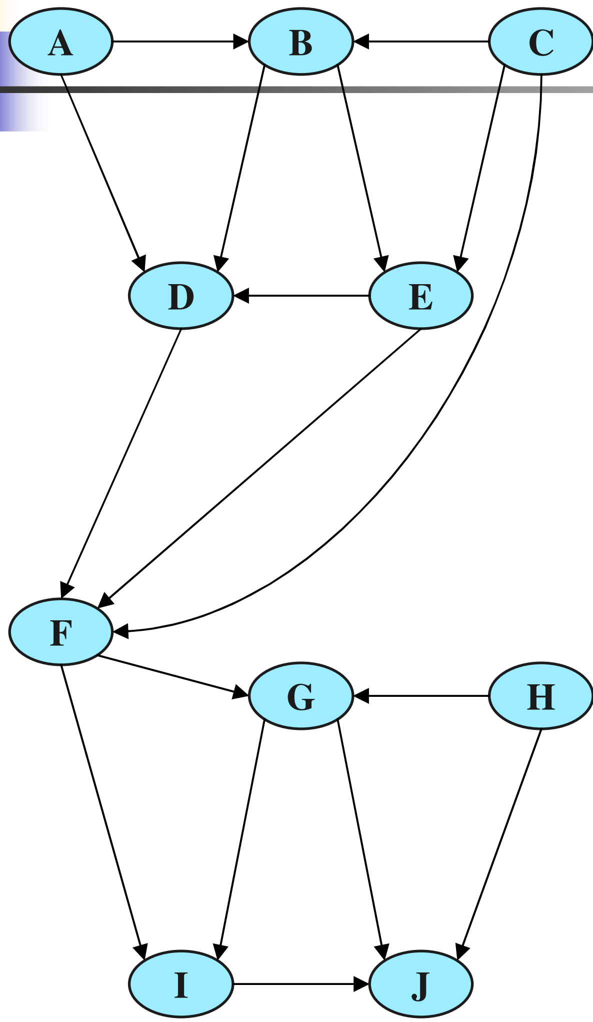
- IBP corresponds to arc-consistency
- For flattened network, IBP = arc-consistency,
- Arc-consistency converges
- IBP's zero belief is correct.
- Questions:
  - Can tractable classes for arc-consistency shed light on IBP's performance?
  - Can this correspondence inspire improvements to IBP?

IJGP - The basic idea

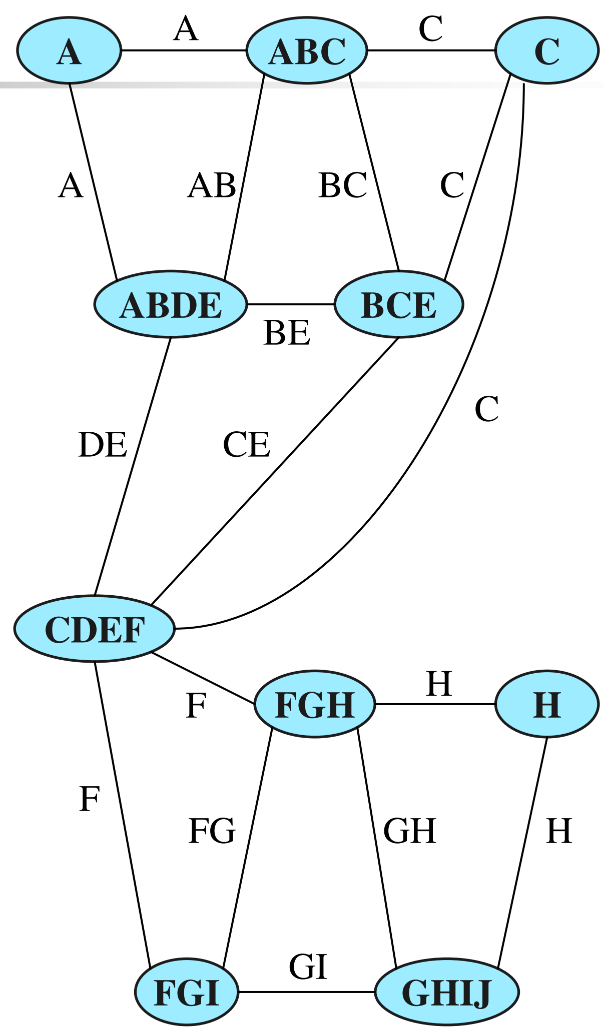
- Can we improve IBP convergence? Accuracy?
- Can we have anytime behavior?

- Idea: Apply join-tree propagation to any join-graph
- Join-graphs that avoid redundant cycles are best (avoid over-counting)
- Result: use *minimal arc-labeled* join-graphs

IJGP - Example

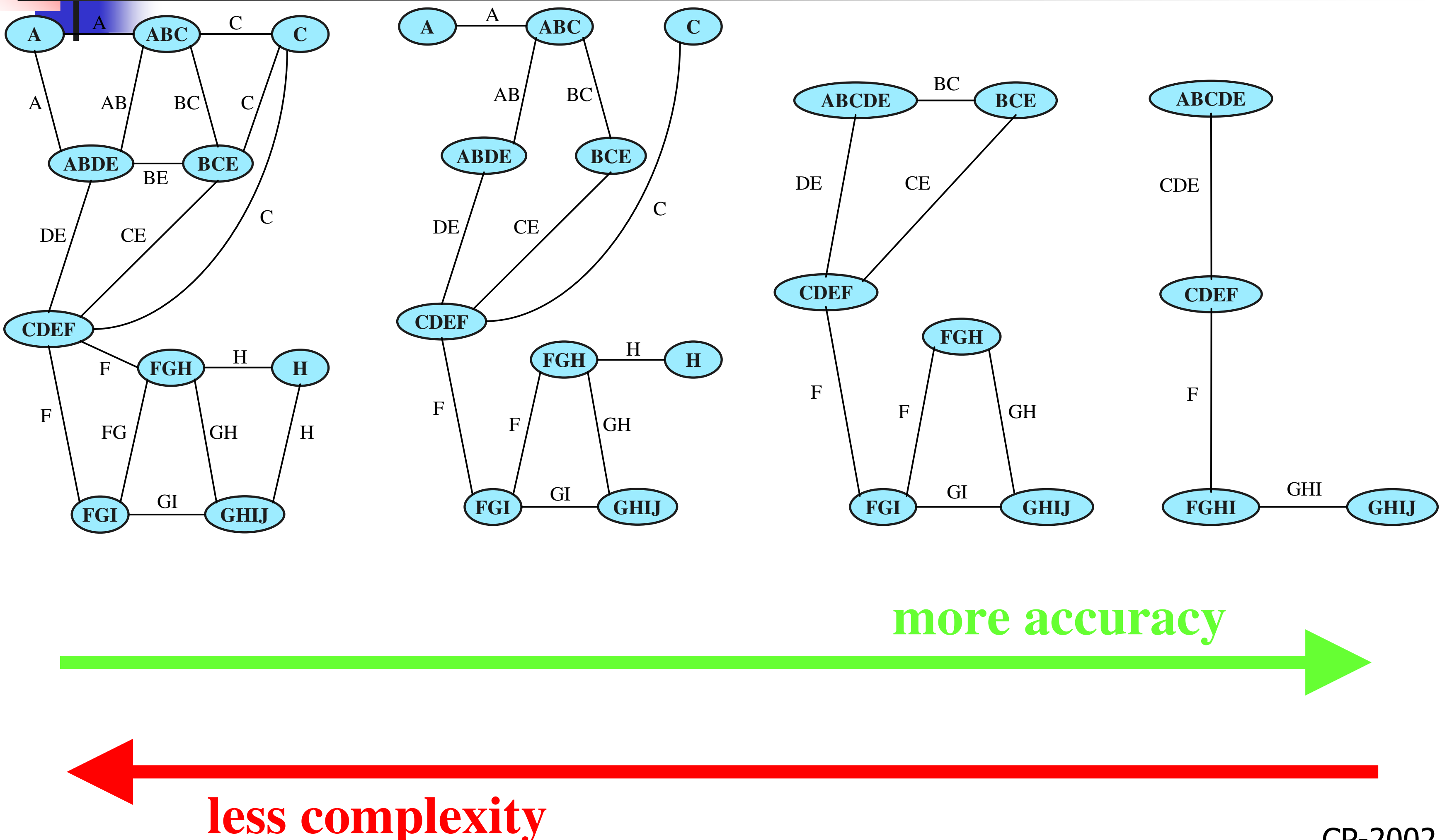


a) Belief network

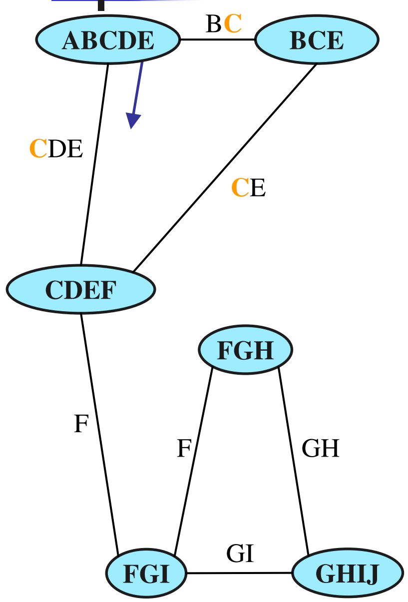


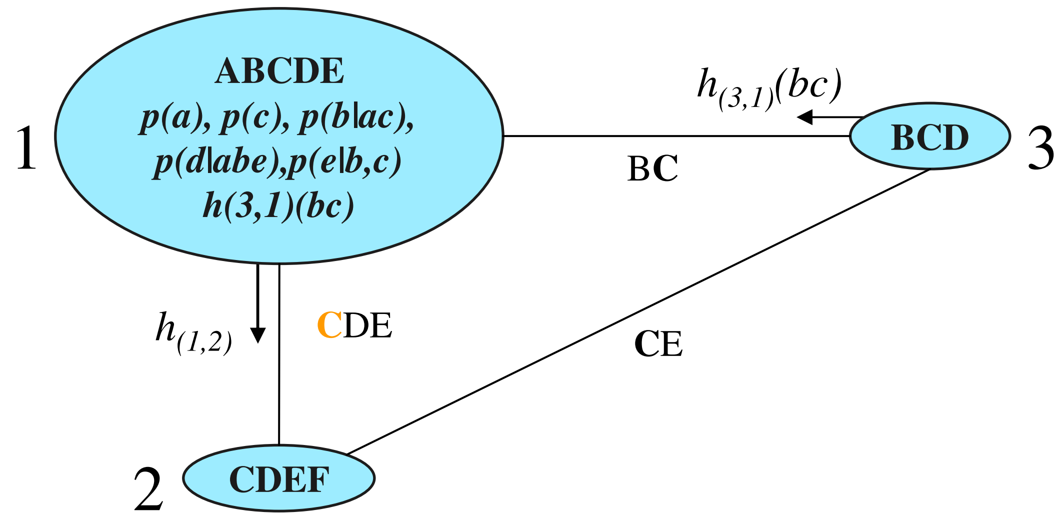
a) The graph IBP works on

Join-graphs


A
ABC
C
ABDE
BCE
CDEF
FGH
H
FGI
GHIJ
ABCDE
FGHI
more accuracy
less complexity

Message propagation





Minimal arc-labeled:

sep(1,2)={D,E}

elim(1,2)={A,B,C}

Non-minimal arc-labeled:

sep(1,2)={C,D,E}

elim(1,2)={A,B}

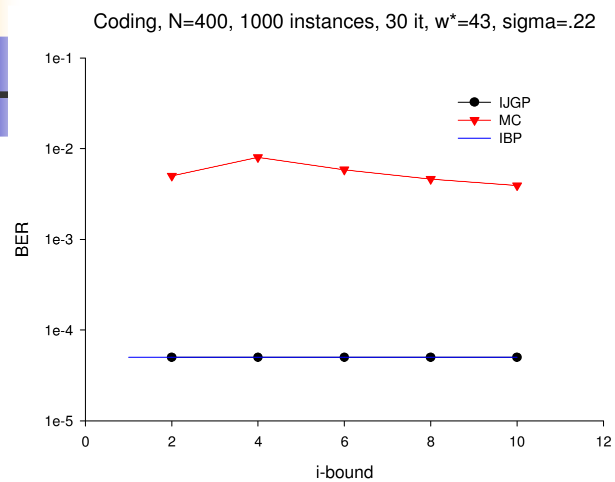
$$h_{(1,2)}(de) = \sum_{a,b,c} p(a)p(c)p(b \mid ac)p(d \mid abe)p(e \mid bc)h_{(3,1)}(bc)$$

$$h_{(1,2)}(cde) = \sum_{a,b} p(a)p(c)p(b \mid ac)p(d \mid abe)p(e \mid bc)h_{(3,1)}(bc)$$

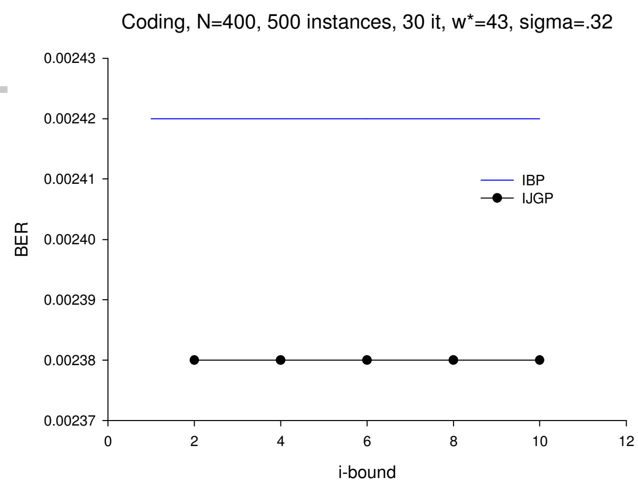
IJGP properties

- IJGP(*i*) applies BP to min arc-labeled join-graph, whose cluster size is bounded by i.
- On join-trees IJGP finds exact beliefs
- Complexity of one iteration:
  - time: $O(deg \bullet (n+N) \bullet k^{i+1})$
  - space: $O(N \bullet k^{\theta})$
- Still,
  - no guaranteed convergence
  - no bound on accuracy

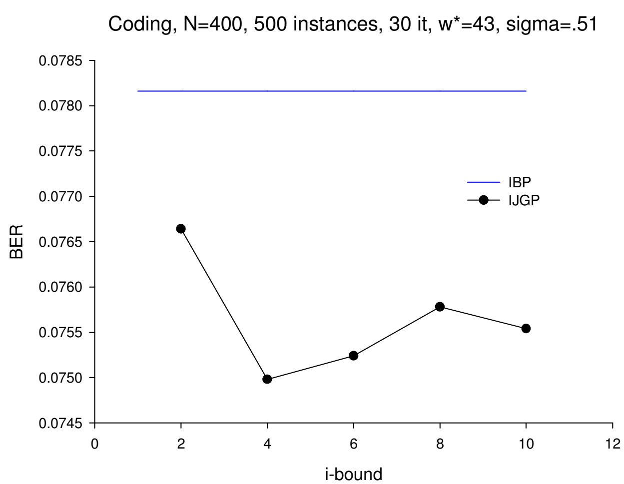
Coding networks - BER



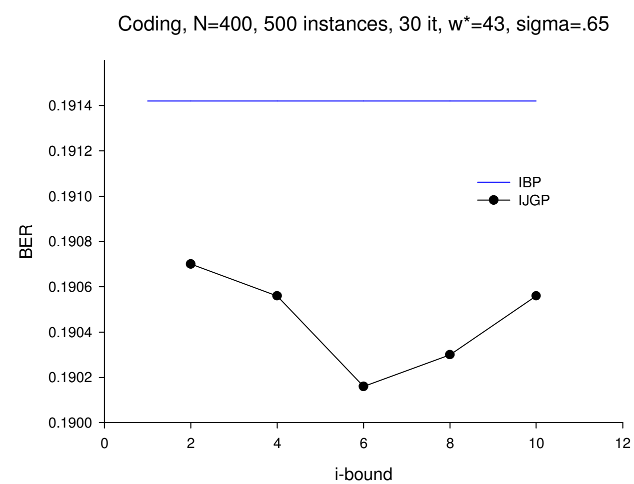
sigma=.22



sigma=.32

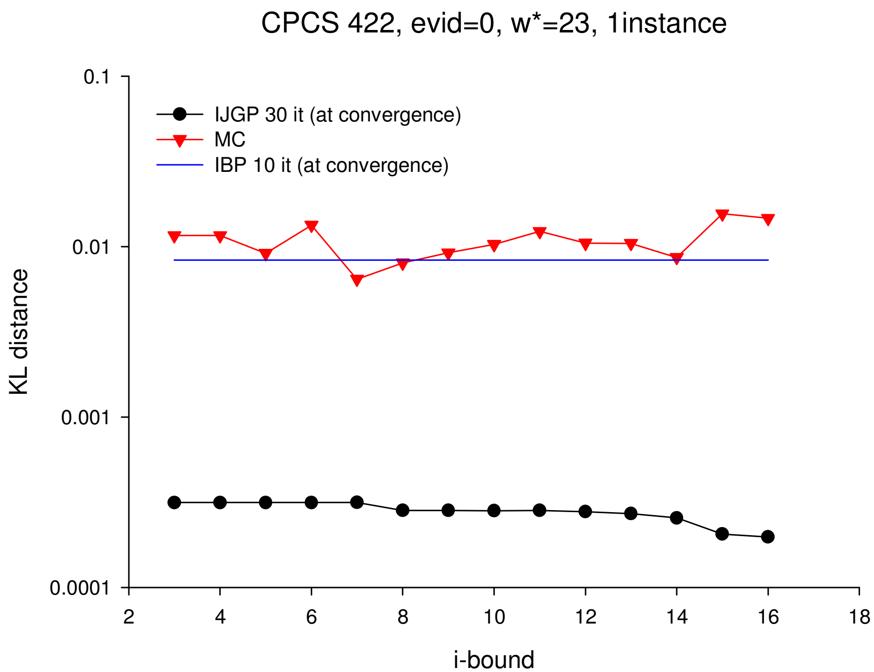


sigma=.51

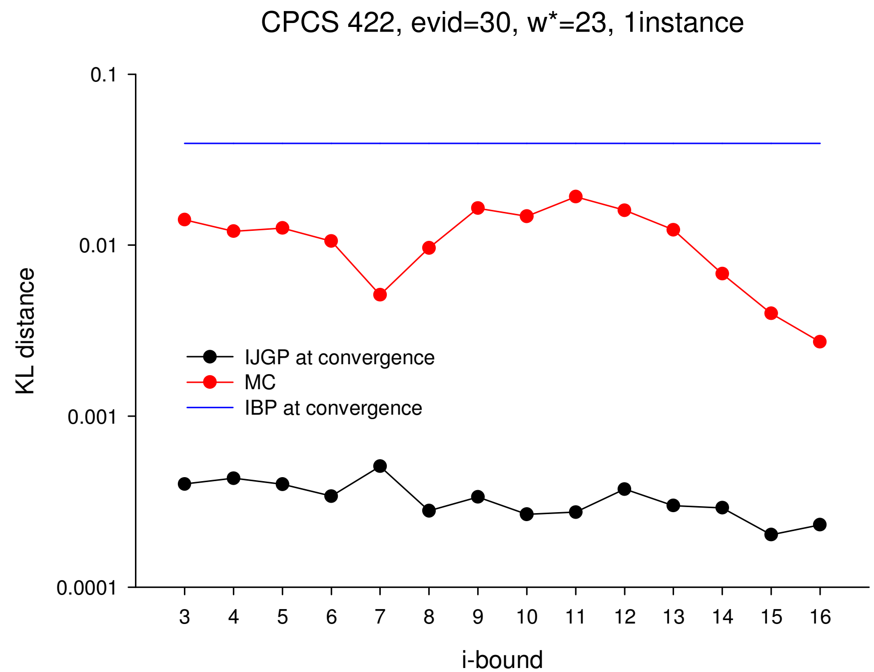


sigma=.65

CPCS 422 – KL distance



evidence=0



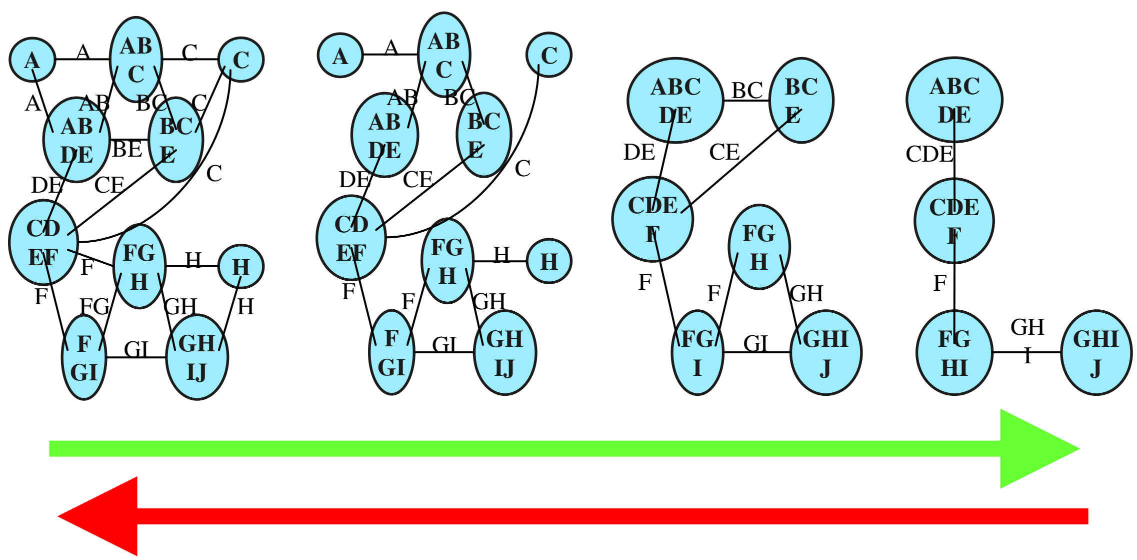
evidence=30

Conclusion

- IJGP borrows the iterative feature from IBP and the anytime virtues of bounded inference from MC
- Empirical evaluation showed the potential of IJGP, which improves with iteration and most of the time with i-bound, and scales up to large networks
- IJGP is almost always superior, often by a high margin, to IBP and MC
- Based on all our experiments, we think that IJGP provides a practical breakthrough to the task of belief updating

Back to Constraints

- IJGP suggests a new variant of constraint propagation (iterative join-graph consistency) which:
  - is guaranteed to converge
  - Guarantee to improve with i-bounds
- Implies IJGP is sound for zero beliefs

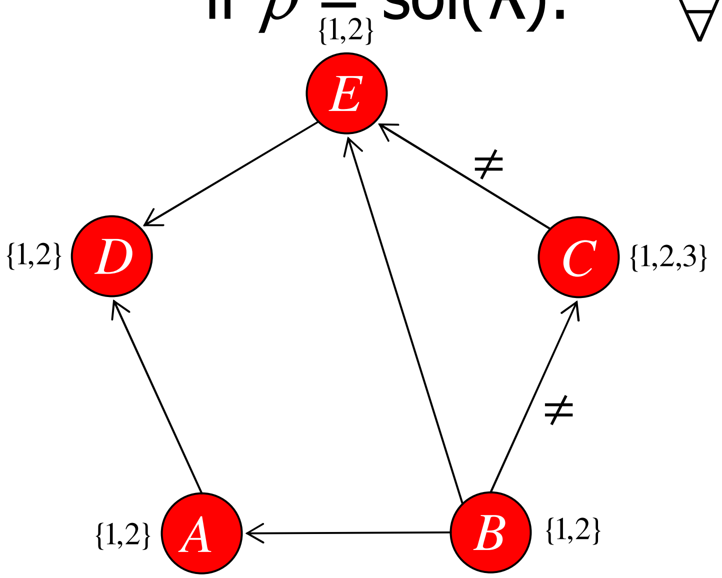


Overview

- Preliminaries
- Observing the commonalities and differences: constraint networks vs probabilistic networks.
- Importing constraint propagation ideas into probabilistic inference:Mini-bucket/ mini-clustering
- Iterative join-graph propagation
- Hybrid processing of constraints and probabilities
- **Random sampling of constraint networks solutions**
- Conclusions

Generate Random Solutions

- Motivation: generating tests for hardware verification
- Given a CSP, $R = (X,D,C)$, generate solutions for R *s.t.* if $\rho = \text{sol}(R)$: $\forall t \in \rho, P(t) = \frac{1}{|\rho|}$



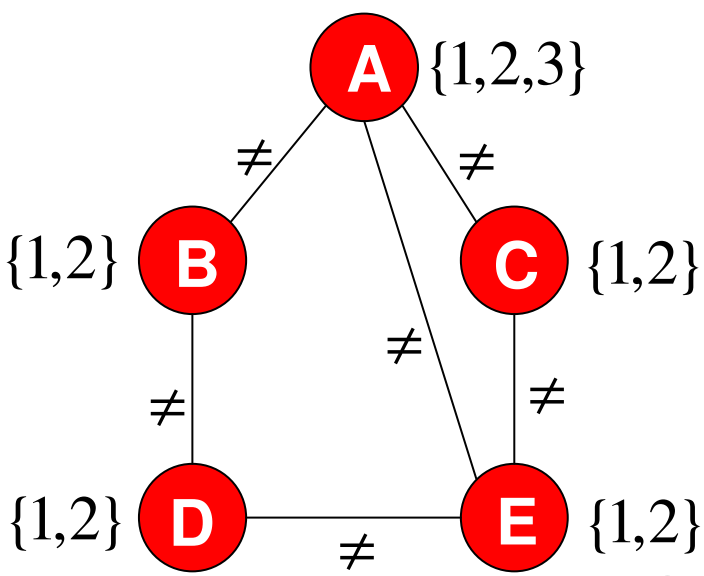
A	B	C	D	E	P
1	2	3	2	1	0.5
2	1	3	1	2	0.5

Brute-force: generate and list all solutions

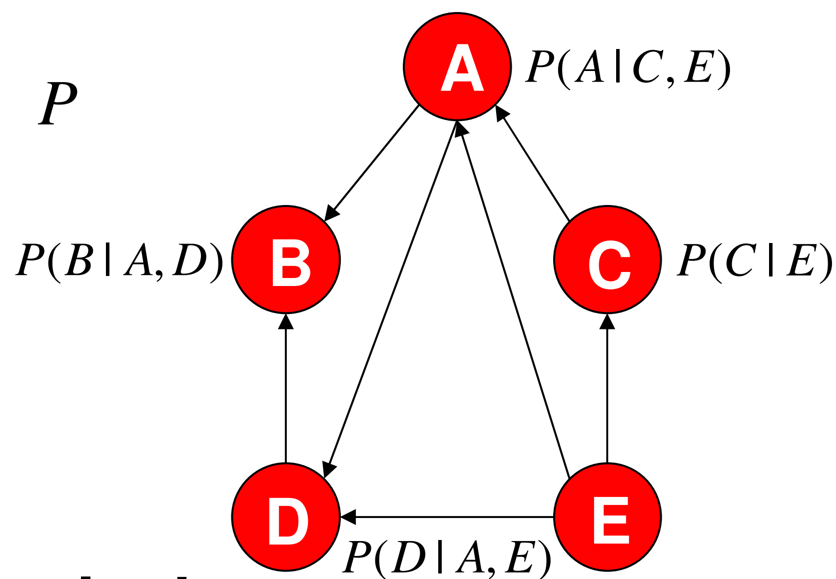
Modeling C*N* as B*N* on the same variables (Approach 2):

- Find a B*N* over same variables s.t.

$$P(x_1,\ldots,x_n) = \frac{1}{\#\text{sol}} \qquad \textit{if } (x_1,\ldots,x_n) \text{ is a solution}$$



$$R \rightarrow P$$



Conversion solved problem

Random Sampling of Belief Networks

- **Forward sampling**
  - Sample all variables with priors
  - For each X_i such that all variables in $pa(X_i)$ have been sampled, pick a value for X_i randomly according to $P(X_i \mid pa(X_i))$
- **Gibbs sampling**
  - Randomly assign X_i
  - Repeat for all j:
    - Pick X_j randomly according to its Markov neighborhood

Empirical Results II

$N = 40, K = 5, C = 90. w^* = 10.8.$ 20 instances.								
		i=4	i=5	i=6	i=7	i=8	i=9	i=10
time		0.05	0.09	0.33	1.3	5.2	20	86
T	KL_u	KL_i abs-e rel-e	KL_i abs-e rel-e	KL_i abs-e rel-e	KL_i abs-e rel-e	KL_i abs-e rel-e	KL_i abs-e rel-e	KL_i abs-e rel-e
8	0.398	0.223 0.106 1.56	0.184 0.095 1.13	0.144 0.081 0.86	0.086 0.058 0.65	0.091 0.058 0.64	0.063 0.045 0.48	0.020 0.026 0.21
9	0.557	0.255 0.110 37	0.323 0.125 28	0.303 0.112 23	0.132 0.074 5.16	0.109 0.064 1.76	0.082 0.053 0.99	0.085 0.045 0.61
10	0.819	0.643 0.164 28	0.480 0.124 7.51	0.460 0.123 9.41	0.340 0.108 5.41	0.295 0.105 4.31	0.401 0.098 2.69	0.228 0.064 0.81
11	1.237	0.825 0.203 1.33	0.803 0.184 1.65	1.063 0.209 2.71	0.880 0.166 1.15	0.249 0.088 0.88	0.276 0.098 1.24	0.193 0.068 0.33

Idea

- **Theorem:** Given CN, BN(CN) gives a BN whose distribution is uniform(CN) with complexity exponential in the induced width of the ordered CN
- **Use forward-sampling or Gibbs sampling to sample the solutions.**
- If it is too expensive… approximate the conversion (directional I-consistency, mini-buckets) and then sample.
- Knuth sampling is not correct even if the network is backtrack-free.

CONCLUSION

- BN vs CN: common computational aspect
- Difference in semantics and level of basic knowledge of the world
- Cross-fertilization is worthwhile:
  - Importing bounded inference CN -> BN
  - Importing sampling BN -> CN
- Semantics for hybrids should be developed