Constraints and Probabilistic networks: a look at the interface

Rina Dechter
Information and Computer Science
University of California, Irvine

Collaboration with: Kalev Kask, Robert Mateescu, David Larkin
Probabilistic vs Deterministic networks

1. Understanding
2. Algorithm Cross Fertilization
3. Hybrids: combine? Subsume?

Semantic? Algorithms?

BN

CN

CP-2002
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): \( A \neq B, \ A \neq D, \ D \neq E, \ etc. \)

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 E: \( E \neq D, \ E \neq C \)
Bucket D: \( D \neq A \)
Bucket C: \( C \neq B \)
Bucket B: \( B \neq A \)
Bucket A: contradiction

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

CP-2002
Probabilistic Networks

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

Conditional Independencies \(\rightarrow\) Efficient Representation

\[
\begin{array}{c|cc}
C & B & D=0 & D=1 \\
0 & 0 & 0.1 & 0.9 \\
0 & 1 & 0.7 & 0.3 \\
1 & 0 & 0.8 & 0.2 \\
1 & 1 & 0.9 & 0.1 \\
\end{array}
\]
What are they good for?

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

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

Variable Elimination

\( h^B(a,d,c,e) \)
Bucket elimination

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

\[
\sum \prod_b
\]

Elimination operator

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

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

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

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

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

\( P(a|e=0) \)

“induced width” (max clique size)

\( W^* = 4 \)
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

<table>
<thead>
<tr>
<th></th>
<th>Search</th>
<th>Variable Elimination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Worst-case time</td>
<td>$O(\exp(n))$</td>
<td>$O(n \exp(w^*))$</td>
</tr>
<tr>
<td></td>
<td>$O(\exp(cutset))$</td>
<td>$w^* \leq n$</td>
</tr>
<tr>
<td></td>
<td>$O(\exp(dfs - depth))$</td>
<td></td>
</tr>
<tr>
<td>Average time</td>
<td>Better than worst-case</td>
<td>Same as worst-case</td>
</tr>
<tr>
<td>Space</td>
<td>$O(n)$</td>
<td>$O(n \exp(w^*))$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$w^* \leq n$</td>
</tr>
<tr>
<td>Output</td>
<td>One solution</td>
<td>Knowledge compilation</td>
</tr>
</tbody>
</table>

*Note:* Knowledge compilation one solution.
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

**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

![Diagram](image)

represents \( sol \ (A, B, C, D, E, F) \)
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
- $P(H \mid X,Y,Z) = 1$ iff $Z = X+Y$
Modeling CN as BN

Is the network consistent? 
P(C_1, ..., C_6) > 0)?

Find a solution.

\textit{sol} (A, B, C, D, E)

\[ \prod_{\text{marginal}} P(A, C, D, D, E, C_1, ..., C_7) \]
A variable-elimination conversion
(eliminates new variables, and more...)

\[ P(B | A, D) = \frac{R(A, B) \times R(B, D)}{\sum_B R(A, B) \times R(B, D)} \]

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

Complexity: \( \exp(w^*) \)
But the network is already easy
A variable-elimination conversion (into a pure BN)

\[ R \rightarrow P \]

\[
P(B \mid A, D) = \frac{R(A, B) \times R(B, D)}{\sum_B R(A, B) \times R(B, D)}
\]

\[
R(A, D) = \sum_B R(A, B) \times R(B, D)
\]

\[
P(A \mid C, E) = |
\]

\[
P(B \mid A, D)
\]

\[
P(B \mid A, D)
\]

\[
P(C \mid E)
\]

\[
P(D \mid A, E)
\]

\[
P(E)
\]
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:**
  \[
  P_h(\bar{x}) = \begin{cases} 
  P_{BN}(\bar{x}) & \text{if } \bar{x} \text{ in sol}(CN) \\
  0 & \text{otherwise} 
  \end{cases}
  \]
  
  \[
  = P_{BN}(\bar{x} \mid \bar{x} \in \text{sol}(CN))
  \]

- **Queries:**
  \[
  P(x_1 \mid \text{CN}) = ?, P(\text{CN consistent}) = ?
  \]

- **Processing:** express as conditional pure Bayesian network with hidden variable (approach 1), or

- **Variable-eliminate hidden variables to get a pure BN** (approach 2)

---

CP-2002
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

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 ≠ D, E ≠ C, E ≠ B
D: D ≠ C, D ≠ A
C: C ≠ B
B: A ≠ B
A:
The idea of Mini-bucket MPE task (Dechter and Rish 1997)

Split a bucket into mini-buckets => bound complexity

\[
\text{bucket}(X) = \{ h_1, \ldots, h_r, h_{r+1}, \ldots, h_n \}
\]

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

\[
\begin{align*}
\{ h_1, \ldots, h_r \} & \quad \text{and} \quad \{ h_{r+1}, \ldots, h_n \} \\
g^X &= (\max_X \prod_{i=1}^r h_i) \cdot (\max_X \prod_{i=r+1}^n h_i)
\end{align*}
\]

\[
h^X \leq g^X
\]

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**

\[
\begin{align*}
\text{Mini-buckets} & \quad \text{Max variables in a mini-bucket} \\
\max_B & \\
P(\text{elb,c}) & P(\text{bla}) \quad 3 \\
P(\text{cla}) & \quad \mathcal{H}^B(e,c) \quad 3 \\
 & \quad \mathcal{H}^B(d,a) \quad 2 \\
E = 0 & \quad \mathcal{H}^C(e,a) \quad 2 \\
P(a) & \quad \mathcal{H}^E(a) \quad 1 \\
U = \text{Upper bound (MPE)} & \\
\end{align*}
\]

\[w^* = 2\]  \( \quad \text{MPE} \quad w^* = 4\]
CPCS networks – medical diagnosis (noisy-OR model)

Test case: no evidence

Anytime-mpe(0.0001)
U/L error vs time

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>cpcs360</th>
<th>cpcs422</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>elim-mpe</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>anytime-mpe(ε), ε = 10^{-4}</strong></td>
<td>70.3</td>
<td>505.2</td>
</tr>
<tr>
<td><strong>anytime-mpe(ε), ε = 10^{-1}</strong></td>
<td>70.3</td>
<td>110.5</td>
</tr>
</tbody>
</table>

CP-2002
A tree decomposition for a belief network $BN = \langle X, D, G, P \rangle$ is a triple $< T, \chi, \psi >$, where $T = (V, E)$ is a tree and $\chi$ and $\psi$ 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 $\text{scope}(p_i) \subseteq \chi(v)$
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

\[ h_{(1,2)}(b,c) = \sum_a p(a) \cdot p(b \mid a) \cdot p(c \mid a,b) \]

\[ h_{(1,2)}^1(b,c) = \sum_a p(a) \cdot p(b \mid a) \cdot p(c \mid a,b) \]

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

\[ h_{(2,3)}^1(b) = \sum_{c,d} p(d \mid b) \cdot h_{(1,2)}^1(b,c) \]

\[ h_{(2,3)}^2(f) = \sum_{c,d} p(f \mid c,d) \]

Cluster Tree Elimination Mini-Clustering, i=3

\[ p(a), p(b \mid a), p(c \mid a,b) \]

\[ p(e \mid b,f) \]

\[ p(f \mid c,d) \]

\[ p(g \mid e,f) \]

CP-2002
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

CPCS 422, evid=0, w*=23, 1 instance

CPCS 422, evid=10, w*=23, 1 instance

evidence=0  
evidence=10

CP-2002
Grid 15x15 - 10 evidence

Grid 15x15, evid=10, w*=22, 10 instances

NHD

0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00

MC
IBP

i-bound

0 2 4 6 8 10 12 14 16 18

Absolute error

0.06
0.05
0.04
0.03
0.02
0.01
0.00

MC
IBP

i-bound

0 2 4 6 8 10 12 14 16 18

Relative error

0.12
0.10
0.08
0.06
0.04
0.02
0.00

MC
IBP

i-bound

0 2 4 6 8 10 12 14 16 18

Time (seconds)

12
10
8
6
4
2
0

MC
IBP

i-bound

0 2 4 6 8 10 12 14 16 18
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 Propagation

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

One step: update $\text{BEL}(U_1)$

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

CP-2002
Belief Propagation

\[ P(x_i) \]

\[ h_{x_i}(u) = \sum_{x_i} P(u \mid x_i)P(x_i) \]

\[ x_1 \]

\[ h_{x_2}(u) \]

\[ u \]

\[ h_u(v) \]

\[ v \]

\[ P(v \mid u) \]

Compute the message:

\[ h_u(v) = \alpha \sum_{u} P(v \mid u) \bullet h_{x_1}(u) \bullet h_{x_2}(u) \bullet ..., \bullet h_{x_n}(u) \]

Exchanging by relational operators: join, project

\[ h_u(v) = \downarrow_v [ R(u, v) \otimes h_{x_1}(u) \otimes h_{x_2}(u) \otimes ..., \otimes h_{x_n}(u) ] \]

Performs arc – consistency (relational, generalized)\textsubscript{CP-2002}
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
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}

\[
h_{(1,2)}(de) = \sum_{a,b,c} p(a) p(c) p(b | ac) p(d | a| b) p(e | bc) h_{(3,1)}(bc)
\]

\[
h_{(1,2)}(cde) = \sum_{a,b} p(a) p(c) p(b | ac) p(d | a| b) p(e | bc) h_{(3,1)}(bc)
\]
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: $O(\text{deg} \cdot (n+N) \cdot k^{i+1})$
  - space: $O(N \cdot k^0)$
- Still,
  - no guaranteed convergence
  - no bound on accuracy
Coding networks - BER

Coding, N=400, 1000 instances, 30 it, w*=43, sigma=.22

sigma=.22

Coding, N=400, 500 instances, 30 it, w*=43, sigma=.32

sigma=.32

Coding, N=400, 500 instances, 30 it, w*=43, sigma=.51

sigma=.51

Coding, N=400, 500 instances, 30 it, w*=43, sigma=.65

sigma=.65

CP-2002
CPCS 422 – KL distance

CPCS 422, evid=0, w*=23, 1 instance

KL distance

i-bound

evidence=0

CPCS 422, evid=30, w*=23, 1 instance

KL distance

i-bound

evidence=30

CP-2002
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 \( \rho = \text{sol}(R) \):
  \[
  \forall t \in \rho, \quad P(t) = \frac{1}{|\rho|}
  \]

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>0.5</td>
<td></td>
</tr>
</tbody>
</table>

Brute-force: generate and list all solutions
Modeling CN as BN on the same variables (Approach 2):

- Find a BN over same variables s.t.

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

Conversion solved problem
Random Sampling of Belief Networks

- **Forward sampling**
  - Sample all variables with priors
  - For each $X_i$ such that all variables in $\text{pa}(X_i)$ have been sampled, pick a value for $X_i$ randomly according to $P(X_i \mid \text{pa}(X_i))$

- **Gibbs sampling**
  - Randomly assign $X_i$
  - Repeat for all $j$:
    - Pick $X_j$ randomly according to its Markov neighborhood
Empirical Results II

<table>
<thead>
<tr>
<th>$T$</th>
<th>$KL_{u_i}$ abs-e</th>
<th>$KL_{u_i}$ rel-e</th>
<th>$KL_{d_i}$ abs-e</th>
<th>$KL_{d_i}$ rel-e</th>
<th>$KL_{i}$ abs-e</th>
<th>$KL_{i}$ rel-e</th>
<th>$KL_{ii}$ abs-e</th>
<th>$KL_{ii}$ rel-e</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>0.398</td>
<td>0.223</td>
<td>0.184</td>
<td>0.144</td>
<td>0.086</td>
<td>0.091</td>
<td>0.063</td>
<td>0.020</td>
</tr>
<tr>
<td></td>
<td>0.106</td>
<td>0.095</td>
<td>0.081</td>
<td>0.058</td>
<td>0.058</td>
<td>0.045</td>
<td>0.026</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.56</td>
<td>1.13</td>
<td>0.86</td>
<td>0.65</td>
<td>0.64</td>
<td>0.48</td>
<td>0.21</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0.557</td>
<td>0.255</td>
<td>0.323</td>
<td>0.303</td>
<td>0.132</td>
<td>0.109</td>
<td>0.082</td>
<td>0.085</td>
</tr>
<tr>
<td></td>
<td>0.110</td>
<td>0.125</td>
<td>0.112</td>
<td>0.074</td>
<td>0.064</td>
<td>0.053</td>
<td>0.045</td>
<td></td>
</tr>
<tr>
<td></td>
<td>37</td>
<td>28</td>
<td>23</td>
<td>5.16</td>
<td>1.76</td>
<td>0.99</td>
<td>0.61</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.819</td>
<td>0.643</td>
<td>0.480</td>
<td>0.460</td>
<td>0.340</td>
<td>0.295</td>
<td>0.401</td>
<td>0.228</td>
</tr>
<tr>
<td></td>
<td>0.164</td>
<td>0.124</td>
<td>0.123</td>
<td>0.108</td>
<td>0.105</td>
<td>0.098</td>
<td>0.064</td>
<td></td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>7.51</td>
<td>9.41</td>
<td>5.41</td>
<td>4.31</td>
<td>2.69</td>
<td>0.81</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1.237</td>
<td>0.825</td>
<td>0.803</td>
<td>1.063</td>
<td>0.880</td>
<td>0.249</td>
<td>0.276</td>
<td>0.193</td>
</tr>
<tr>
<td></td>
<td>0.203</td>
<td>0.184</td>
<td>0.209</td>
<td>0.166</td>
<td>0.088</td>
<td>0.098</td>
<td>0.068</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.33</td>
<td>1.65</td>
<td>2.71</td>
<td>1.15</td>
<td>0.88</td>
<td>1.24</td>
<td>0.33</td>
<td></td>
</tr>
</tbody>
</table>

$N = 40, K = 5, C = 90, \theta = 10.8, 20$ instances.
**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