Inference and Search for
Probabilistic and deterministic Graphical Models

Rina Dechter
Bren school of ICS, University of California, Irvine

Joint work with Radu Marinescu, Robert Mateescu, Kalev Kask, Irina Rish

HUJI 2012
Outline

- Graphical models: reasoning principles
- Inference
- Search; via AND/OR Search
- Lower Bounding schemes for inference
- Lower-bounding heuristics for AND/OR search
- 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.
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
Graphical Models

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

Operators:
- combination
- elimination (projection)

Tasks:
- **Belief updating**: \(\Sigma_{x \setminus y} \Pi_j P_i\)
- **MPE**: \(\max_X \Pi_j P_j\)
- **CSP**: \(\Pi_X \times \Pi_j C_j\)
- **Max-CSP**: \(\min_X \Sigma_j F_j\)

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

HUJI 2012
Graphical Models

- A graphical model \((X,D,F)\):
  - \(X = \{X_1, \ldots, X_n\}\) variables
  - \(D = \{D_1, \ldots, D_n\}\) domains
  - \(F = \{f_1, \ldots, f_r\}\) functions
    (constraints, CPTs, CNFs ...)

- Tasks:
  - Belief updating: \(\Sigma_{x-y} \Pi_j p_i\)
  - MPE: \(\max_X \Pi_j p_j\)
  - CSP: \(\Pi_{x \in X} c_j\)
  - Max-CSP: \(\min_X \Sigma_j F_j\)

The MRF

\(x = (x_1, \ldots, x_n)\) to all the variables which maximizes the sum of the factors:

\[
\text{MAP}(\theta) = \max_x \sum_{i \in V} \theta_i(x_i) + \sum_{f \in F} \theta_f(x_f).
\]  

(1.1)
Bayesian Networks (Pearl 1988)

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

\[ \text{Combination: Product} \]
\[ \text{Marginalization: sum/max} \]

\[ P(x_1...x_n) = \prod_i p(x_i \mid pa(x_i)) \]
\[ P(e) = \sum_{x \in E} \prod_i p(x_i \mid pa(x_i)) \]
\[ mpe = \max_x P(x) \]
Monitoring Intensive-Care Patients

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

Map coloring

Variables: countries (A B C etc.)

Values: colors (red green blue)

Constraints: A ≠ B, A ≠ D, D ≠ E, ...

Constraint graph

Queries: Find one solution, all solutions, counting
Propositional Satisfiability

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

Combination: “AND”
Marginalization: ?
Resolution does combine followedBy project
Mixed Networks
(Mateescu and Dechter, 2004)

Belief Network  Constraint Network

Moral mixed graph

Examples: NLP, Linkgage, Software verification, probabilistic languages

\[ P(D | B, C) \]

<table>
<thead>
<tr>
<th>B</th>
<th>C</th>
<th>D=0</th>
<th>D=1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>.2</td>
<td>.8</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>.1</td>
<td>.9</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>.3</td>
<td>.7</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>.5</td>
<td>.5</td>
</tr>
</tbody>
</table>

\[ P_M(\bar{x}) = \begin{cases} 
  P_B(\bar{x} | \bar{x} \in \rho) = \frac{P_B(\bar{x})}{P_B(\bar{x} \in \rho)}, & \text{if } \bar{x} \in \rho \\
  0, & \text{otherwise} 
\end{cases} \]

Complex cnf queries: \( P((A \lor B) \land (\neg CVD)) \)

HUJI 2012
Solution methods

- **Solving** tree is easy
- **Inference:** move to trees by clustering
  - (dynamic programming, variable elimination, junction trees)
  - Exploit structure well.
- **Search:** move to trees by conditioning.
  - Can also exploit structure well.
Tree-solving is Easy

Belief updating
(sum-prod)

CSP – consistency
(projection-join)

Dynamic Programming,
Inference

MPE (max-prod)

Trees are processed in linear time and memory
Message-passing

#CSP (sum-prod)

HUJI 2012
Clustering and Treewidth

**Inference algorithm:**
*Time:* \(\exp(\text{tree-width}+1)\)
*Space:* \(\exp(\text{separator-width})\)

\[\text{treewidth} = 4 - 1 = 3\]

\[\text{treewidth} = (\text{maximum cluster size}) - 1\]

Separator-width=2

HUJI 2012
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

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

“Moral” graph
Bucket elimination Algorithm BE-bel (Dechter 1996)

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

\[ \sum \prod_b \quad \text{Elimination operator} \]

bucket B: \quad \begin{align*}
P(b|a) & \quad P(d|b,a) \quad P(e|b,c) \\
\end{align*}

bucket C: \quad \begin{align*}
P(c|a) & \quad \lambda^B(a,d,c,e) \\
\end{align*}

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

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

bucket A: \quad P(a) \quad \lambda^E(a)

\[ P(e=0) \]

\[ P(a|e=0) = \frac{P(a,e=0)}{P(e=0)} \]

HUJI 2012
Inference for Optimization: Bucket Elimination

**Algorithm BE-mpe** (Dechter 1996, Bertele and Briochi, 1977)

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

\[ \max_{\mathbf{x}} \prod \]

**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:**
- \( P(a) \)
- \( h^E(a) \)

\( W^* = 4 \)

"induced width" (max clique size)

HUJI 2012
Generating the MPE-tuple

\[ a' = \arg \max_a P(a) \cdot h^E(a) \]

\[ b' = \arg \max \frac{P(b | a') \times P(d' | b, a') \times P(e' | b, c')}{P(b | a)} \]

\[ c' = \arg \max \frac{P(c | a') \times h^B(a', d', c, e')}{P(c | a)} \]

\[ d' = \arg \max \frac{h^C(a', d, e')}{h^C(a, d, e)} \]

\[ e' = 0 \]

\[ B: \quad P(b | a) \quad P(d | b, a) \quad P(e | b, c) \]

\[ C: \quad P(c | a) \quad h^B(a, d, c, e) \]

\[ D: \quad h^C(a, d, e) \]

\[ E: \quad e = 0 \quad h^D(a, e) \]

\[ A: \quad P(a) \quad h^E(a) \]

Return \((a', b', c', d', e')\)
Complexity of Elimination

\[ O(n \exp(w^*(d))) \]

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

**The effect of the ordering:**

```
<table>
<thead>
<tr>
<th>Ordering</th>
<th>Induced Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>( d_1 )</td>
<td>4</td>
</tr>
<tr>
<td>( d_2 )</td>
<td>2</td>
</tr>
</tbody>
</table>
```

"Moral" graph

HUJI 2012
Bucket Elimination

Finding max

Assignment argmax

Messages

Ordering: (A, B, C, D, E, F, G)
Tree-solving is Easy

Belief updating  
*(sum-prod)*

CSP – consistency  
*(projection-join)*

**Dynamic Programming, Inference**

MPE *(max-prod)*

#CSP *(sum-prod)*

*Trees are processed in linear time and memory*

Message-passing

HUJI 2012
Conditioning and Cycle cutset

Cycle cutset = \{A,B,C\}
Search over the Cutset (cont)

Graph Coloring problem

Time exp in cycle-cutset
Memory-linear

- Inference may require too much memory
- Condition on some of the variables

Inference may require too much memory
Condition on some of the variables
Conditioning vs. Elimination

Conditioning (search)

A

B

C

D

E

F

G

A=1

A=k

k “sparser” problems

Elimination (inference)

A

B

C

D

E

F

G

1 “denser” problem

HUJI 2012
Inference vs. Conditioning

- **By Inference (thinking)**

  Exponential in *treewidth*
  Time and memory

- **By Conditioning (guessing)**

  Exponential in *cycle-cutset*
  Time-wise, linear memory
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
The Search Space

\[ f(X) = \min_X \sum_{i=1}^{9} f_i(X) \]
The Search Space

\[ f(X) = \min_X \sum_{i=1}^{9} f_i(X) \]

are calculated based on cost components
The Search Space

\[ f(x) = \sum_{i=1}^{9} f_i(x) \]

\[
\min_{a,b,c,d,e,f} \quad f_1(a,b) + f_2(a,c) + f_3(a,f) + f_4(b,c) + f_5(b,d) + f_6(b,e) + f_7(c,d) + f_8(e,f)
\]
**An Optimal Solution**

\[ f(X) = \sum_{i=1}^{9} f_i(X) \]

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

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 (Freuder & Quinn85)
The AND/OR Search Tree

A solution subtree is

(A=0, B=1, C=0, D=0, E=1, F=1)
Weighted AND/OR Search Tree

\[ f(x) = \sum_{i=1}^{9} f_i(x) \]

Node Value
(bottom-up evaluation)

OR – minimization
AND – summation
AND/OR vs. OR Spaces

54 nodes

126 nodes
Pseudo-Trees
(Freuder 85, Bayardo 95, Bodlaender and Gilbert, 91)

\[ m \leq w \times \log n \]

(a) Graph
(b) DFS tree
   depth=3
(c) pseudo-tree
   depth=2
(d) Chain
   depth=6

HUJI 2012
AND/OR Tree DFS Algorithm (Belief Updating)

**Evidence: E=0**

```
<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>E=0</th>
<th>E=1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>.4</td>
<td>.6</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>.5</td>
<td>.5</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>.7</td>
<td>.3</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>.2</td>
<td>.8</td>
<td></td>
</tr>
</tbody>
</table>
```

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

**Evidence: D=1**

```
<table>
<thead>
<tr>
<th></th>
<th>B</th>
<th>C</th>
<th>D=0</th>
<th>D=1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>.2</td>
<td>.8</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>.1</td>
<td>.9</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>.3</td>
<td>.7</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>.5</td>
<td>.5</td>
<td></td>
</tr>
</tbody>
</table>
```

**OR node:** Marginalization operator (summation)

**AND node:** Combination operator (product)

**Value of node = updated belief for sub-problem below**
## Complexity of AND/OR Tree Search

<table>
<thead>
<tr>
<th></th>
<th>AND/OR tree</th>
<th>OR tree</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Space</strong></td>
<td>$O(n)$</td>
<td>$O(n)$</td>
</tr>
<tr>
<td><strong>Time</strong></td>
<td>$O(n , k^m)$</td>
<td>$O(k^n)$</td>
</tr>
<tr>
<td></td>
<td>$O(n , k^{w*} , \log n)$</td>
<td></td>
</tr>
</tbody>
</table>
| (Freuder & Quinn85), (Collin, Dechter & Katz91), (Bayardo & Miranker95), (Darwiche01)

$k =$ domain size  
m = depth of pseudo-tree  
n = number of variables  
w* = treewidth
From Search Trees to Search Graphs

- Any two nodes that root identical sub-trees or sub-graphs can be merged
From Search Trees to Search Graphs

- Any two nodes that root identical sub-trees or sub-graphs can be merged.
From AND/OR Tree
An AND/OR Graph
Context-based Caching

- Caching is possible when context is the same

- context = parent-separator set in induced pseudo-graph
  = current variable + parents connected to subtree below

\[
\text{context}(B) = \{A, B\} \\
\text{context}(c) = \{A, B, C\} \\
\text{context}(D) = \{D\} \\
\text{context}(F) = \{F\}
\]
AND/OR Tree DFS Algorithm (Belief Updating)

**Value of node:** Updated belief for sub-problem below

**AND node:** Combination operator (product)

**OR node:** Marginalization operator (summation)

**Evidence:** D=1

**Evidence:** E=0

**Result:** $P(D=1, E=0)$
AND/OR Graph DFS Algorithm (Belief Updating)

\[
P(E \mid A, B) \quad P(B \mid A) \quad P(C \mid A) \quad P(A)
\]

\[
\begin{array}{ccc}
A & B & E=0 \ E=1 \\
0 0 & .4 & .6 \\
0 1 & .5 & .5 \\
1 0 & .7 & .3 \\
1 1 & .2 & .8 \\
\end{array}
\]

\[
\begin{array}{ccc}
A & B=0 \ B=1 \\
0 0 & .4 & .6 \\
0 1 & .7 & .3 \\
1 0 & .2 & .8 \\
1 1 & .1 & .9 \\
\end{array}
\]

\[
\begin{array}{ccc}
A & C=0 \ C=1 \\
0 0 & .2 & .8 \\
0 1 & .7 & .3 \\
1 0 & .6 & .4 \\
1 1 & .1 & .9 \\
\end{array}
\]

\[
\begin{array}{c}
P(A) \\
0 & .6 \\
1 & .4 \\
\end{array}
\]

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

Evidence: \( E=0 \)

Cache table for \( D \)

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

\[
\begin{array}{ccc}
B & C & D=0 \ D=1 \\
0 0 & .2 & .8 \\
0 1 & .1 & .9 \\
1 0 & .3 & .7 \\
1 1 & .5 & .5 \\
\end{array}
\]

Evidence: \( D=1 \)

HUJI 2012
The Effect of Constraint Propagation

**Domains are** {1,2,3,4}

**CONSTRAINTS ONLY**

**FORWARD CHECKING**

**MAINTAINING ARC CONSISTENCY**

HUJI 2012
All Four Search Spaces

Full OR search tree
126 nodes

Full AND/OR search tree
54 AND nodes

Context minimal OR search graph
28 nodes

Context minimal AND/OR search graph
18 AND nodes

Any query is best computed Over the c-minimal AO space
Complexity of AND/OR Graph Search

<table>
<thead>
<tr>
<th></th>
<th>AND/OR graph</th>
<th>OR graph</th>
</tr>
</thead>
<tbody>
<tr>
<td>Space</td>
<td>$O(n \ k^{w*})$</td>
<td>$O(n \ k^{pw*})$</td>
</tr>
<tr>
<td>Time</td>
<td>$O(n \ k^{w*})$</td>
<td>$O(n \ k^{pw*})$</td>
</tr>
</tbody>
</table>

$k$ = domain size  
$n$ = number of variables  
$w*$ = treewidth  
$pw*$ = pathwidth
THEOREM (Mateescu and Dechter, 2005):

For positive models and given a pseudo tree, AND/OR search and Bucket-elimination have the performance.
The impact of the pseudo-tree

$W=4, h=8$

Min-Fill (Kjaerulff90)

Hypergraph Partitioning (h-Metis)

What is a good pseudo-tree?
How to find a good one?

$W=5, h=6$

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

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

HUJI 2012
Outline

- Graphical models: the primary reasoning principles
- Inference
- AND/OR Search Trees and Graphs
- Lower Bounding schemes
- AND/OR Branch-and-Bound Search
- Experiments
Primary bounding schemes

- Goal: bound $\min_x \sum_i f_i (x)$ or $\max_x \prod_i f_i (x)$

- Two primary relaxation ideas:
  - Node duplication control:
  - Reparameterization schemes:
    - Soft arc-consistency (Bistareli, 2000, Sciex 2000)
Mini-Bucket: duplicating/Splitting a Node

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

Before Splitting:
Network $N$

After Splitting:
Network $N'$
Semantics of dual decomposition: Splitting each functions + reparameterization

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

Before Splitting:
Network $N$

![Diagram of network $N$]
Reparameterization: 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 for each node:**  
Network $N'$
Reparameterization: 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
Deriving upper-lower bounds

- Mini-Bucket Elimination for graphical models
- Cost-shifting schemes
- Hybrids of cost-shifting and mini-bucket
- The impact of these bounds as heuristics for search
Mini-bucket approximation: MPE task
(Dechter and Rish, 1997, 2003)

Split a bucket into mini-buckets \(\Rightarrow\) bound complexity

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

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

\[
\{ h_1, ..., h_r \} \quad \{ h_{r+1}, ..., h_n \}
\]

\[
g^X = (\max_X \prod_{i=1}^{r} h_i) \cdot (\max_X \prod_{i=r+1}^{n} h_i)
\]

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

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

\[ \max_B \Pi \]

Bucket B

\[ P(E|B,C) \]

Bucket C

\[ P(C|A) \]

Bucket D

\[ h^B(A,D) \]

Bucket E

\[ E = 0 \]

Bucket A

\[ P(A) \]

\[ \max_B \Pi \]

\[ P(B|A) \]

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

Node duplication, renaming

\[ h^B(C,E) \]

\[ h^B(C,E) \]

\[ h^C(A,E) \]

\[ h^E(A) \]

\[ h^D(A) \]

\[ W=2 \]

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

Assignments
argmax

Ordering: (A, B, C, D, E, F, G)
Static Mini-Bucket Heuristics

Node duplication $\rightarrow$ lower bound

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

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>elim-mpe</td>
<td></td>
<td></td>
</tr>
<tr>
<td>anytime-mpe(ε=10^{-4})</td>
<td>70.3</td>
<td>505.2</td>
</tr>
<tr>
<td>anytime-mpe(ε=10^{-1})</td>
<td>70.3</td>
<td>110.5</td>
</tr>
</tbody>
</table>
Mini-Clustering (for sum-product)

Split a cluster into mini-clusters => bound complexity

\[ \{h_1,\ldots,h_r, h_{r+1},\ldots,h_n\} \]

\[ \sum \prod_{i=1}^{n} h_i \leq \left( \sum \prod_{i=1}^{r} h_i \right) \cdot \left( \sum \prod_{i=r+1}^{n} h_i \right) \]

Exponential complexity decrease

\[ O(e^n) \to O(e^{\text{var}(r)}) + O(e^{\text{var}(n-r)}) \]
Grid 15x15 - 10 evidence
(Mateescu, Kask and Dechter, 2002)
Finding optimal i-partition;
(Rollon and Dechter 2010)

- Given an i-bound, and a distance \( d() \), choose partition \( Q^* \) s.t.
  \[
  Q^* = \arg \min_Q \{ d(g^Q, g) \}
  \]
  where \( Q \) is an i-partition

- We considered as distance measure \( d() \):
  - Log relative error (RE)
  - Maximum log relative error (MRE)
  - KL divergence (KL)
  - Absolute error (AE)
Optimizing the partitioning
(Rollon and Dechter 2010)

Scope-based Partitioning scheme (SCP) minimizes the number of mini-buckets as respecting the $i$ bound $i$ (Rish, Kask 2000)

- Log relative error:

$$RE(f, h) = \sum_i (\log(f(t)) - \log(h(t)))$$

- Max log relative error:

$$MRE(f, h) = \max_t \{\log(f(t)) - \log(h(t))\}$$

Partitioning lattice of bucket $\{f_1, f_2, f_3, f_4\}$.

Use greedy heuristic derived from a distance function to decide which functions go into a single mini-bucket.
Experiments: Coding networks

- Performance of the partition heuristics as a function of the $i$-bound.
Reparameterization Schemes:
How to find best cost-shifting bound

\[ MAP = \min_x \sum\{ f \downarrow i \in F \} \uparrow \sum \downarrow f \downarrow i \ (x,y) \]

Task: given a fixed partition \( Q = \{ \{ f \downarrow 1 \}, \ldots, \{ f \} \downarrow r \} \)
find \( f \)'s that minimize distance over classes of cost-shifting.

\[ distance \downarrow f \uparrow \ cost-shifts \uparrow \ \{ \text{Map}, \sum \{ f' \downarrow i \in F \} \uparrow \min f' \downarrow i \ (x,y) \} \]

Soft arc-consistency uses heuristic idea.

Optimal schemes: Optimal soft arc-consistency (OSAC), and dual-decompositions (MPLP).
Reparameterization using Linear relaxation-based schemes (MPLP class) \( \text{Globerson and Jaakkola, Nips 2008} \)

**Find:**

\[
    x = (x_1, \ldots, x_n) \text{ to all the variables which maximizes the sum of the factors:}
    \]

\[
    \text{MAP}(\theta) = \max_x \sum_{i \in V} \theta_i(x_i) + \sum_{f \in F} \theta_f(x_f). \quad (1.1)
    \]

**Best upper bound by Equivalence preserving transformations:**

\[
    \min_{\delta} L(\delta), \quad (1.2)
    \]

\[
    L(\delta) = \sum_{i \in V} \max_{x_i} \left( \theta_i(x_i) + \sum_{f : i \in f} \delta_{fi}(x_i) \right) + \sum_{f \in F} \max_{x_f} \left( \theta_f(x_f) - \sum_{i \in f} \delta_{fi}(x_i) \right).
    \]

\[
    \delta_{fi}(x_i) \text{ Is the cost shifted from } f \text{ to value } x_i \text{ of } X_i.
    \]

There are several variations of scheme computing the optimizing shifts based on partial gradient descent, which differ by what is being kept constant. The 1.2 task is the Dual of a linear relaxation of the original problem.
Mini-Bucket with moment-matching
(Ihler, Flerova, Dechter, Otten, 2011)

• **MBE**: non-iterative message-passing schemes
• **iterative schemes using re-parametrization**
• **MPLP** [Globerson, Jakkola, Sontag et al. 2008],
• **Max-sum diffusion** [Kovalevsky et al. 1975]
• **Soft arc-consistency** [Schiex 2000, Bistarelli et al. 2000]
Mini-Bucket with moment-matching
(Ihler, Flerova, Dechter, Otten, 2011)

\[ C_1(x_2, x_3) = \max_{x_1} f_1(x_1, x_2) \cdot f_3(x_1, x_3) \]

\[ C_1(x_2, x_3) = \max_{x_1} f_1(x_1, x_2) \cdot f_3(x_1, x_3) \cdot \frac{g(x_1)}{g(x_1)} \]

\[ C(x_2, x_3) \leq \hat{C}_1(x_2, x_3) = \max_{x_1'} f_1(x_1', x_2) g(x_1') \cdot \max_{x_1''} f_3(x_1'', x_3) / g(x_1'') \]

\[ \max_{x_2} f_1(x_1, x_2) g(x_1) = \max_{x_3} f_3(x_1, x_3) / g(x_1) \]

\[ g(x_1) = \sqrt{\frac{\max_{x_3} f_3(x_1, x_3)}{\max_{x_2} f_1(x_1, x_2)}} \]

\[ \hat{C}_1(x_2, x_3) = \max_{x_1} f_1(x_1, x_2) \sqrt{\frac{\max_{x_3} f_3(x_1, x_3)}{\max_{x_2} f_1(x_1, x_2)}} \cdot \max_{x_1} f_3(x_1, x_3) \sqrt{\frac{\max_{x_2} f_1(x_1, x_2)}{\max_{x_3} f_3(x_1, x_3)}} \]
Outline

- Graphical models: reasoning principles
- Inference
- Search; via AND/OR Search
- Lower Bounding schemes for inference
- Lower-bounding heuristics for AND/OR search
- Experiments
AND/OR Branch-and-Bound

\[ \text{ub}(n) \geq h(n) \]
Bucket Elimination

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

Messages
Finding max

Assignment
argmax

\[ h^*(a, b, c) = h^D(a, b, c) + h^E(b, c) \]

Ordering: \((A, B, C, D, E, F, G)\)
Static Mini-Bucket Heuristics

Node duplication $\rightarrow$ 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)
Outline

- Graphical models: the primary reasoning principles
- Inference
- AND/OR Search Trees and Graphs
- Lower Bounding heuristics for search
- AND/OR Branch-and-Bound Search
- Experiments
## Grid Networks (BN)

(Sang et al.05)

<table>
<thead>
<tr>
<th>Samlam</th>
<th>MBE(i)</th>
<th>MBE(i)</th>
<th>MBE(i)</th>
<th>MBE(i)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BB-C+SMB(i)</td>
<td>BB-C+SMB(i)</td>
<td>BB-C+SMB(i)</td>
<td>BB-C+SMB(i)</td>
</tr>
<tr>
<td></td>
<td>AOBB+ SMB(i)</td>
<td>AOBB+C +SMB(i)</td>
<td>AOBF+C+SMB(i)</td>
<td>AOBF-C+SMB(i)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>i=12</td>
<td>i=14</td>
<td>i=16</td>
<td>i=18</td>
</tr>
<tr>
<td></td>
<td>time</td>
<td>nodes</td>
<td>time</td>
<td>nodes</td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>90-24-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(33, 111)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(576, 20)</td>
<td>0.28</td>
<td>0.64</td>
<td>1.69</td>
<td>4.60</td>
</tr>
<tr>
<td>out</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>-</td>
<td>1273.09</td>
<td>9,047,518</td>
</tr>
<tr>
<td>out</td>
<td>-</td>
<td>-</td>
<td>21.94</td>
<td>75,637</td>
</tr>
<tr>
<td>90-34-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(45, 153)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(1154, 80)</td>
<td>0.63</td>
<td>1.25</td>
<td>3.72</td>
<td>11.66</td>
</tr>
<tr>
<td>out</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>out</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>90-38-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(47, 163)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(1444, 120)</td>
<td>0.78</td>
<td>1.67</td>
<td>4.20</td>
<td>12.36</td>
</tr>
<tr>
<td>out</td>
<td>2032.33</td>
<td>6,835,745</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>969.02</td>
<td>2,623,971</td>
<td>1753.10</td>
<td>3,794,053</td>
</tr>
<tr>
<td></td>
<td>101.69</td>
<td>174,786</td>
<td>103.80</td>
<td>146,237</td>
</tr>
</tbody>
</table>

*Min-fill pseudo tree. Time limit 1 hour.*
## Genetic Linkage Analysis

### Pedigree Information

<table>
<thead>
<tr>
<th>Pedigree</th>
<th>Samlam Superlink</th>
<th>MBE(i) BB-C+SMB(i)</th>
<th>MBE(i) BB-C+SMB(i)</th>
<th>MBE(i) BB-C+SMB(i)</th>
<th>MBE(i) BB-C+SMB(i)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>time</td>
<td>nodes</td>
<td>time</td>
<td>nodes</td>
</tr>
<tr>
<td>ped30</td>
<td></td>
<td>0.42</td>
<td>10212.70</td>
<td>885.82</td>
<td>82,552,957</td>
</tr>
<tr>
<td></td>
<td></td>
<td>13095.83</td>
<td>93,233,570</td>
<td>out</td>
<td>out</td>
</tr>
<tr>
<td>ped33</td>
<td></td>
<td>0.58</td>
<td>34,229,495</td>
<td>737.96</td>
<td>9,114,411</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2804.61</td>
<td>11,349,475</td>
<td>307.39</td>
<td>1,504,020</td>
</tr>
<tr>
<td>ped42</td>
<td></td>
<td>4.20</td>
<td>1426.99</td>
<td>140.61</td>
<td>407,387</td>
</tr>
<tr>
<td></td>
<td></td>
<td>561.31</td>
<td>out</td>
<td>out</td>
<td>2364.67</td>
</tr>
</tbody>
</table>

### Min-fill pseudo tree. Time limit 3 hours.
MBE-MM vs. MPLP (pedigrees)

MBE, MBE-MM z-bound = 10, time cutoff = 3600 sec
MPLP on original factors, cutoff = 1500 iterations

HUJI 2012
Runtime (sec) (pedigrees) by AOBB with MBE, MBE-MM or MPLP as a heuristic generator.

<table>
<thead>
<tr>
<th>Instances</th>
<th>AOBB-MBE(z)</th>
<th>AOBB-MBE-MM(z)</th>
<th>AOBB-MPELPLP(z)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(n,k,w,h)</td>
<td>z-bound=4</td>
<td>z-bound=5</td>
<td>z-bound=8</td>
</tr>
<tr>
<td>pedigree1 298.4,15.48</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>pedigree7 867.4,32.90</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>pedigree9 935,7,27,100</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>pedigree20 387.5,22,60</td>
<td>4460</td>
<td>167</td>
<td>137</td>
</tr>
<tr>
<td>pedigree23 309.5,25,51</td>
<td>45</td>
<td>22</td>
<td>13</td>
</tr>
<tr>
<td>pedigree25 993.5,25,69</td>
<td>13321</td>
<td>36</td>
<td>4</td>
</tr>
<tr>
<td>pedigree30 1015.5,21,108</td>
<td>13198</td>
<td>1690</td>
<td>246</td>
</tr>
<tr>
<td>pedigree33 581.4,28,98</td>
<td>24142</td>
<td>201</td>
<td>177</td>
</tr>
<tr>
<td>pedigree37 725.5,21,56</td>
<td>298</td>
<td>33</td>
<td>13</td>
</tr>
<tr>
<td>pedigree39 953.5,21,76</td>
<td>30724</td>
<td>2871</td>
<td>732</td>
</tr>
<tr>
<td>pedigree50 478.6,17,47</td>
<td>25440</td>
<td>39</td>
<td>16</td>
</tr>
</tbody>
</table>

Time cutoff 24h
Memory limit 2Gb
Runtime (sec) expanded (grids) by AOBB with MBE, MBE-MM or MPLP as a heuristic generator.

<table>
<thead>
<tr>
<th>Instances (n.k.w.h)</th>
<th>AOBB-MBE(z)</th>
<th>AOBB-MBE-MM(z)</th>
<th>AOBB-MPLP(z)</th>
<th>AOBB-MBE(z)</th>
<th>AOBB-MBE-MM(z)</th>
<th>AOBB-MPLP(z)</th>
<th>AOBB-MBE(z)</th>
<th>AOBB-MBE-MM(z)</th>
<th>AOBB-MPLP(z)</th>
</tr>
</thead>
<tbody>
<tr>
<td>50-16-5</td>
<td>14047</td>
<td>6759</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>97</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>256,2,21,79</td>
<td>11918</td>
<td>257</td>
<td>209</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>50-20-5</td>
<td>—</td>
<td>—</td>
<td></td>
<td>3589</td>
<td></td>
<td></td>
<td>11</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>400,2,27,97</td>
<td>—</td>
<td>28985</td>
<td></td>
<td>11</td>
<td></td>
<td></td>
<td>26</td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>75-16-5</td>
<td>2457</td>
<td>245</td>
<td></td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>256,2,21,73</td>
<td>511</td>
<td>32</td>
<td>37</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>75-20-5</td>
<td>—</td>
<td>25258</td>
<td>1912</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>400,2,27,99</td>
<td>—</td>
<td>47557 1</td>
<td></td>
<td>6</td>
<td></td>
<td></td>
<td>13</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>90-20-5</td>
<td>7281</td>
<td>1199</td>
<td></td>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>400,2,27,99</td>
<td>5575</td>
<td>585</td>
<td>389</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>90-21-5</td>
<td>7722</td>
<td>1585</td>
<td></td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>441,2,28,106</td>
<td>9064</td>
<td>861</td>
<td>593</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>90-22-5</td>
<td>27283</td>
<td>2327</td>
<td></td>
<td>50</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>484,2,30,109</td>
<td>17130</td>
<td>1172</td>
<td>604</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>90-26-5</td>
<td>—</td>
<td>36469</td>
<td>386</td>
<td>21</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>676,2,36,136</td>
<td>70798</td>
<td>7077</td>
<td>4000</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
</tbody>
</table>

Time cutoff 24 h, memory limit 2 Gb
UAI 2010 evaluation, 2008, 2006

**Toulbar2: INRA**

**Summary:** Toulbar2 is an open source exact anytime Weighted CSP solver using Branch and Bound and soft local consistency

**Team members:** S. de Givry, D. Allouche, A. Favier, T. Schiex


**Contact person:** Thomas Schiex, Thomas.Schiex@toulouse.inra.fr

**Detailed description**

---

**Daoopt: UCI Irvine**

**Summary:** "daoopt" and "daoopt.anytime" are based on AND/OR branch and bound graph search, with mini bucket heuristics and LDS (Limited Discrepancy Search) initialization.

**Team members:** Lars Otten, Rina Dechter

**Additional Contributor:** Radu Marinescu

**Contact person:** Lars Otten, lotten@ics.uci.edu

**Detailed description**

---

**Web-site:** [http://graphmod.ics.uci.edu](http://graphmod.ics.uci.edu)

---

We are first in Pascal 2012, so far…

Please join

3rd in all 3 categories

After Toolbar, Joris
Software

- AND/OR search algorithms
- Bucket-tree elimination
- Generalized belief propagation
- Samplesearch sampling

are available at:

- http://graphmod.ics.uci.edu/group/Software
Thank you!

Irina Rish

Radu Marinescu For publication see:
http://www.ics.uci.edu/~dechter/publications.html

Robert Mateescu We are first in Pascal challenge 2012 (Globerson, Elidan), so far…Please join

Vibhav Gogate

Emma Rollon Natalia Flerova

Lars Otten HUJI 2012