From Constraint Programming to Graphical models; the role of AND/OR search

Rina Dechter
Bren school of ICS

University of California, Irvine

Collaborators: Kalev Kask
Bozhena Bidyuk
Radu Marinescu, Robert Mateescu
Vibhav Gogate

DOD SAT workshop, March 2008
From Constraint Programming to Graphical Models

**Languages/modeling**
- Eclipse, ILOG solver
- CPLex

**Graphical models:**
- Probabilistic networks
- Cost networks
- Influence diagrams
- MDPs

**Queries:**
- Likelihood computation
- Constraint Optimization

**Principles:**
- Decomposition
- Equivalence
- Pruning

**Algorithms over:**
- Constraint networks
- Queries: constraint satisfaction, Satisfiability/counting

**SAT/CSP:**
- Using the simplest model
- Focus on algorithms/data-structures
- Code perfection/code sharing

**Current focus: Mixed networks**
- Use sat as subroutine
- Apply the same principle.
Principles for SAT/CSPs

Constraint Satisfaction/counting:
- Problem decomposition: backjumping
- Subproblem equivalence:
  - Learn nogoods (clause learning)
  - Learn goods
- Pruning: constraint propagation, unit propagation

Combinatorial optimization/ Likelihood queries:
- Decomposition: (AND/OR)
- Equivalence: caching optimal conditioned solutions
- Pruning: by mini-bucket, soft arc-consistency, belief propagation, lower-bound heuristic
Overview

- Introduction to graphical models algorithms: Inference, search and hybrids.

- Exact Algorithms: AND/OR search spaces

- AND/OR search for combinatorial optimization

- Current focus:
  - AND/OR Compilation
  - Approximation by Sampling and belief propagation
Example: map coloring

Variables - countries (A, B, C, etc.)
Values - colors (red, green, blue)
Constraints: \[ A \neq B, \quad A \neq D, \quad D \neq E, \quad \text{etc.} \]

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>red</td>
<td>green</td>
</tr>
<tr>
<td>red</td>
<td>yellow</td>
</tr>
<tr>
<td>green</td>
<td>red</td>
</tr>
<tr>
<td>green</td>
<td>yellow</td>
</tr>
<tr>
<td>yellow</td>
<td>green</td>
</tr>
<tr>
<td>yellow</td>
<td>red</td>
</tr>
</tbody>
</table>

Semantics: set of all solutions
Primary task: find a solution
Propositional Satisfiability

\[ \varphi = \{ (\neg C), (A \lor B \lor C), (\neg A \lor B \lor E), (\neg B \lor C \lor D) \}. \]
Constraint Optimization

- Variables ⇒ Nodes
- Constraints ⇒ Edges
- e.g.:
  \[ f_1(x_1, x_2, x_3) \]
  \[ f_2(x_2, x_3, x_5) \]
  \[ f_3(x_1, x_4) \]
  \[ f_4(x_4, x_5) \]

\[ \min \{ \sum_{i \in Sol} \sum_{i=1}^{m'} f_i(t) \} \]
Belief Updating, Most probable tuple (MPE)

- \( P(\text{lung cancer}=\text{yes} \mid \text{smoking}=\text{no}, \text{dyspnoea}=\text{yes}) = ? \)
- **MPE** = find argmax \( P(S) \cdot P(C|S) \cdot P(B|S) \cdot P(X|C,S) \cdot P(D|C,B) = ? \)
Mixed Networks
(Mateescu and Dechter, 2004)

Belief Network

Constraint Network

Moral mixed graph

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

<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>

Complex cnf queries:
P((A or B) and (~CVD))

\[
P_M(\overline{x}) = \begin{cases} 
    P_B(\overline{x} | \bar{x} \in \rho) = \frac{P_B(\overline{x})}{P_B(\overline{x} \in \rho)}, & \text{if } \bar{x} \in \rho \\
    0, & \text{otherwise}
\end{cases}
\]
Linkage analysis:
6 people, 3 markers
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, CPT, CNFs …)

Operators:
- combination
- elimination (projection)

Primary tasks:
- **Belief updating**: \(\Sigma_{x,y} \Pi_j P_i\)
- **Combinatorial optimization**: \(\max_x \Pi_j P_j\)
- **Constraint satisfaction**: \(\Pi_{x \times j} C_j\)
- **Max expected utility**

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

- **Constraints:**
  - Scheduling, design, diagnosis, planning

- **Belief networks, Markov fields:**
  - Prediction, diagnosis, situation assessment, monitoring, learning

- **Influence diagrams, Factored MDPS:**
  - Planning and decision making under uncertainty.

- **Decision making agents require**
  - Constraints and probabilities to model the world.
  - Decision variable, and cost functions to model agents goals and actions.
Sample Domains for Graphical Models

- Web Pages and Link Analysis
- Linkage analysis
- Communication Networks (Cell phone Fraud Detection)
- Natural Language Processing (e.g. Information Extraction and Semantic Parsing)
- Object Recognition and Scene Analysis
- Battle-space Awareness
- Epidemiological Studies
- Citation Networks
- Geographical Information Systems
- Intelligence Analysis (Terrorist Networks)
- Financial Transactions (Money Laundering)
- Computational Biology

...
Solution Techniques

All queries are NP-hard so: exploit structure, identify tractable classes, approximate

Search: Conditioning

**Complete**
- Dfs search, Branch and bound, A*

**Incomplete**
- Simulated Annealing
- Gradient Descent sampling

**Hybrids**

**Incomplete**
- Local Consistency
- Unit Resolution mini-bucket(i)

**Complete**
- Adaptive Consistency
- Tree Clustering
- Dynamic Programming
- Resolution

Time: \(\text{exp}(n)\)
Space: linear

Time: \(\text{exp}(w*)\)
Space: \(\text{exp}(w*)\)

Trading space for time
Tree-solving is Easy

Belief updating (sum-prod)

CSP – consistency (projection-join)

MPE (max-prod)

Trees are processed in linear time and memory
Also Acyclic graphical models
Inference and Treewidth

**Inference algorithm:**

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

$\text{treewidth} = \frac{\text{maximum cluster size}}{0} - 1$

Separator-width = 2
Cluster Tree Propagation

Join-tree clustering (Spigelhalter et. al. 1988, Dechter, Pearl 1987)

Time: \( O(\exp(w+1)) \)
Space: \( O(\exp(sep)) \)

For each cluster \( P(X|e) \) is computed
Conditioning and Cycle cutset

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

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

Graph Coloring problem
Time vs Space for w-cutset

- Random Graphs (50 nodes, 200 edges, average degree 8, $w^* \approx 23$)

$W$-cutset time $O(\exp(w+\text{cutset-size}))$

Space $O(\exp(w))$

(Dechter and El-Fatah, 2000)
(Larrosa and Dechter, 2001)
(Rish and Dechter 2000)
Approximation

- Since inference, search and hybrids are too expensive when graph is dense; (high treewidth) then:

- **Bounding inference:**
  - mini-bucket and mini-clustering
  - Belief propagation

- **Bounding search:**
  - Sampling

- Goal: an anytime scheme
Inference vs Search

**Inference**
- decomposition
- equivalence

Exp(w*) time/space

**Search**
- Pruning → 
Exp(n) time
O(n) space
Overview

- Introduction to graphical models algorithms: Inference, search and hybrids.

- AND/OR search spaces
  - Decomposition in AND/OR trees
  - Equivalence in AND/OR Graphs

- AND/OR search for combinatorial optimization

- Current focus:
  - AND/OR Compilation
  - Approximation by Sampling and belief propagation
AND/OR Search Space

Constraint network

DFS tree
AND/OR vs. OR

AND/OR size: exp(4), OR size exp(6)
AND/OR vs. OR
with Constraints

No-goods
(A=1, B=1)
(B=0, C=0)

AND/OR

OR

AND

OR

AND

OR

AND

AND/OR

OR
AND/OR vs. OR
with Constraints

No-goods
(A=1, B=1)
(B=0, C=0)
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
DFS algorithm (#CSP example)

Value of node = number of solutions below it
AND/OR tree search (belief updating)

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

Weighted AND/OR
Has weights on arcs

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

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

OR node: Marginalization operator (summation)
AND node: Combination operator (product)
Value of node = updated belief for subproblem below
AND/OR Tree Search for COP

AND node = Combination operator (summation)

OR node = Marginalization operator (minimization)

Goal: \( \min_X \sum_{i=1}^{9} f_i(X) \)
# Complexity of AND/OR Tree Search

SAT: Backjumping will do  
Counting: special care needed

<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>
</tbody>
</table>
| **Time**       | $O(n \, k^m)$  
                 | $O(n \, k^{w^* \log n})$  
                 | $O(k^n)$  |

[Freuder & Quinn85], [Collin, Dechter & Katz91],  
[Bayardo & Miranker95], [Darwiche01]

$k = \text{domain size}$  
$m = \text{depth of pseudo-tree}$  
$n = \text{number of variables}$  
$w^* = \text{treewidth}$

**Tasks:** Consistency, Counting,  
Optimization, Belief updating  
Max-expected utility, partition function
Overview

- Introduction to graphical models algorithms: Inference, search and hybrids.

- AND/OR search spaces
  - Decomposition in AND/OR trees
  - Equivalence AND/OR Graphs

- AND/OR search for combinatorial optimization

- Current focus:
  - AND/OR Compilation
  - Approximation by Sampling and belief propagation
Any two nodes that root identical subtrees (subgraphs) can be merged.
From AND/OR Tree
An AND/OR Graph
Context-based Caching

- context = current variable + ancestors connected to subtree below

- Caching is possible when context is the same
Context-based Caching

context(A) = \{A\}
context(B) = \{B, A\}
context(C) = \{C, B\}
context(D) = \{D\}
context(E) = \{E, A\}
context(F) = \{F\}

Cache Table (C)

Primal graph

Space: \(O(\exp(2))\)
Example (graph search)

Goal: \( \min_X \sum_{i=1}^9 f_i(X) \)
AND/OR Tree DFS Algorithm
( Belief Updating )

Evidence: E = 0

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

\[
\begin{array}{cccc}
A & B & E = 0 & E = 1 \\
0 & 0 & 0.4 & 0.6 \\
0 & 1 & 0.5 & 0.5 \\
1 & 0 & 0.7 & 0.3 \\
1 & 1 & 0.2 & 0.8 \\
\end{array}
\]

\[
\begin{array}{cccc}
A & B = 0 & B = 1 \\
0 & 0.4 & 0.6 \\
1 & 0.1 & 0.9 \\
\end{array}
\quad
\begin{array}{cccc}
A & C = 0 & C = 1 \\
0 & 0.2 & 0.8 \\
1 & 0.7 & 0.3 \\
\end{array}
\quad
\begin{array}{c}
P(A) \\
0 & 0.6 \\
1 & 0.4 \\
\end{array}
\]

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

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

\[
\begin{array}{cccc}
B & C & D = 0 & D = 1 \\
0 & 0 & 0.2 & 0.8 \\
0 & 1 & 0.1 & 0.9 \\
1 & 0 & 0.3 & 0.7 \\
1 & 1 & 0.5 & 0.5 \\
\end{array}
\]

Evidence: D = 1

OR node: Marginalization operator (summation)
AND node: Combination operator (product)
Value of node = updated belief for sub-problem below

Context

[ ]

[A]

[AB]

[BC]

[AC]

[ABC]
AND/OR Graph DFS Algorithm
(Belief Updating)

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

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

Evidence: \( E=0 \)

\[
\begin{array}{c|c|c}
A & B & E=0 \mid E=1 \\
0 & 0 & .2 \mid .8 \\
& 1 & .7 \mid .3 \\
1 & 0 & .3 \mid .7 \\
1 & 1 & .5 \mid .5 \\
\end{array}
\]

Context

\[
P(D=1, E=0)
\]

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

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

Evidence: \( D=1 \)
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
How Big Is the Context?

Theorem: The maximum context size for a pseudo tree is equal to the treewidth of the graph along the pseudo tree.
## 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><strong>Space</strong></td>
<td>$O(n , k^{w^*})$</td>
<td>$O(n , k^{pw^*})$</td>
</tr>
<tr>
<td><strong>Time</strong></td>
<td>$O(n , k^{w^*})$</td>
<td>$O(n , k^{pw^*})$</td>
</tr>
</tbody>
</table>

$k = \text{domain size}$  
$n = \text{number of variables}$  
$w^* = \text{treewidth}$  
$pw^* = \text{pathwidth}$

$w^* \leq pw^* \leq w^* \log n$

**Tasks:** Consistency, Counting, Optimization, Belief updating  
Max-expected utility, partition function
Treewidth vs. Pathwidth

**TREE**

treewidth = 3

= (max cluster size) - 1

**CHAIN**

pathwidth = 4

= (max cluster size) - 1
AND/OR Context Minimal Graph

Variable Elimination

(C K H A B E J L N O D P M F G)
Searching AND/OR Graphs

- **AO(j)**: searches depth-first, cache i-context
  - $j$ = the max size of a cache table (i.e. number of variables in a context)

Space: $O(n)$  
Time: $O(\exp(w* \log n))$

- For SAT: formula caching?  
  Clause learning

Space: $O(\exp(j))$  
Time: $O(\exp(m_j + j))$

Space: $O(\exp(w*))$  
Time: $O(\exp(w*))$
The Effect of Constraint Propagation

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

CONSTRAINTS ONLY

FORWARD CHECKING

MAINTAINING ARC CONSISTENCY
Overview

- Introduction to graphical models algorithms: Inference, search and hybrids.

- AND/OR search spaces
  - AND/OR trees
  - AND/OR Graphs

- AND/OR search for combinatorial optimization
  - The mini-bucket heuristic
  - AO depth-first and best-first Branch and Bound
  - Empirical evaluation

- Current focus:
  - AND/OR Compilation
  - Approximation by Sampling and belief propagation
AND/OR Branch-and-Bound (AOBB)  
(Marinescu & Dechter, IJCAI’05)

Maintain

\[ ub = \text{best solution found so far} \]

\[ \text{Prune subtree below } n \text{ if } lb(n) \geq ub \]

\[ lb(n) = g(n) + h(n) \]

\[ g(n) \]

\[ h(n) \]

estimates the optimal cost below n

OR Branch-and-Bound
Mini-bucket Heuristics for BB search
(Kask and dechter AIJ, 2001, Kask, Dechter and Marinescu UAI 2003)

\[ f(a,e,D) = P(a) \cdot h^B(D,a) \cdot h^C(e,a) \]
AND/OR Branch-and-Bound (contd.)

\[ h(n) \geq ub(n) \]
AND/OR Branch and Bound for Constraint Optimization

- Search AND/OR Context-minimal graph
  - exploit decomposition and equivalence

- Prune irrelevance via mini-bucket heuristics, and constraint propagation

- Depth-first (AOBB) and best-first (AOBF)

- Dynamic variable orderings

- Applied to MPE and weighted CSPs

- Applied to Integer Programming
### Genetic Linkage Analysis

(Marinescu & Dechter, AAAI’07; Marinescu & Dechter, UAI’07)

<table>
<thead>
<tr>
<th>ped (w*, h)</th>
<th>Samlam</th>
<th>Superlink</th>
<th>BB-C+SMB(i)</th>
<th>AOB-C+SMB(i)</th>
<th>AOF-C+SMB(i)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>time nodes</td>
<td>time nodes</td>
<td>time nodes</td>
</tr>
<tr>
<td>ped1 (15, 61)</td>
<td>5.44</td>
<td>54.73</td>
<td>1.14 7,997</td>
<td>0.39 4,576</td>
<td>0.26 1,177</td>
</tr>
<tr>
<td>ped38 (17, 59)</td>
<td>out</td>
<td>28.36</td>
<td>- -</td>
<td>2046.95 11,868,672</td>
<td>216.94 583,401</td>
</tr>
<tr>
<td>ped50 (18, 58)</td>
<td>out</td>
<td>-</td>
<td>- -</td>
<td>66.66 403,234</td>
<td>12.75 25,507</td>
</tr>
<tr>
<td>ped18 (21, 119)</td>
<td>157.05</td>
<td>139.06</td>
<td>- -</td>
<td>23.83 118,869</td>
<td>19.85 53,961</td>
</tr>
<tr>
<td>ped25 (29, 53)</td>
<td>out</td>
<td>-</td>
<td>- -</td>
<td>2041.64 6,117,320</td>
<td>out</td>
</tr>
<tr>
<td>ped39 (23, 94)</td>
<td>out</td>
<td>322.14</td>
<td>- -</td>
<td>61.20 313,496</td>
<td>41.69 79,356</td>
</tr>
<tr>
<td>uwlp50-400 (w*, h)</td>
<td>CPLEX</td>
<td>AOBB+PVO</td>
<td>AOBF+PVO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------------</td>
<td>--------</td>
<td>----------</td>
<td>----------</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>time</td>
<td>nodes</td>
<td>time</td>
<td>nodes</td>
<td>time</td>
</tr>
<tr>
<td>uwlp-1 (50, 123)</td>
<td>10.76</td>
<td>12</td>
<td>106.63</td>
<td>29</td>
<td>81.63</td>
</tr>
<tr>
<td>uwlp-4 (50, 123)</td>
<td>6.52</td>
<td>6</td>
<td>55.10</td>
<td>10</td>
<td>51.85</td>
</tr>
<tr>
<td>uwlp-5 (50, 123)</td>
<td>30.55</td>
<td>58</td>
<td>247.03</td>
<td>50</td>
<td>131.58</td>
</tr>
<tr>
<td>uwlp-6 (50, 123)</td>
<td>3.59</td>
<td>0</td>
<td>32.31</td>
<td>1</td>
<td>32.65</td>
</tr>
<tr>
<td>uwlp-8 (50, 123)</td>
<td>3.40</td>
<td>0</td>
<td>96.66</td>
<td>21</td>
<td>60.27</td>
</tr>
<tr>
<td>uwlp-9 (50, 123)</td>
<td>9.02</td>
<td>6</td>
<td>97.00</td>
<td>9</td>
<td>78.05</td>
</tr>
</tbody>
</table>

Uncapacitated Warehouse Location Problems with 50 stores and 400 locations
### MAX-SAT Instances

**(Marinescu & Dechter, CPAIOR’07)**

<table>
<thead>
<tr>
<th>pret</th>
<th>(w*, h)</th>
<th>CPLEX</th>
<th>AOB-B-C</th>
<th>AOBF-C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>time</td>
<td>nodes</td>
<td>time</td>
</tr>
<tr>
<td>pret60-40</td>
<td>(6, 13)</td>
<td>676.94</td>
<td>3,926,422</td>
<td>7.38</td>
</tr>
<tr>
<td>pret60-60</td>
<td>(6, 13)</td>
<td>535.05</td>
<td>2,963,435</td>
<td>7.30</td>
</tr>
<tr>
<td>pret60-75</td>
<td>(6, 13)</td>
<td>402.53</td>
<td>2,005,738</td>
<td>6.34</td>
</tr>
<tr>
<td>pret150-40</td>
<td>(6, 15)</td>
<td>out</td>
<td>75.19</td>
<td>5,625</td>
</tr>
<tr>
<td>pret150-60</td>
<td>(6, 15)</td>
<td>out</td>
<td>78.25</td>
<td>5,813</td>
</tr>
<tr>
<td>pret150-75</td>
<td>(6, 15)</td>
<td>out</td>
<td>84.97</td>
<td>6,144</td>
</tr>
</tbody>
</table>

*pret* MAX-SAT instances solved as 0-1 ILPs
### Genetic Linkage Analysis

*(Marinescu & Dechter, AAAI’07; Marinescu & Dechter, UAI’07)*

<table>
<thead>
<tr>
<th>Pedigree</th>
<th>Samlamo</th>
<th>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>
<th>MBE(i) BB-C+SMB(i)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>v. 2.3.2</td>
<td>v. 1.6</td>
<td>i=6</td>
<td>i=8</td>
<td>i=10</td>
<td>i=12</td>
<td>i=14</td>
</tr>
<tr>
<td><strong>ped1</strong></td>
<td>0.05</td>
<td>0.05</td>
<td>0.11</td>
<td>0.31</td>
<td>0.97</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(15, 61)</td>
<td>24.30</td>
<td>13.17</td>
<td>1.58</td>
<td>0.73</td>
<td>1.31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(299, 5)</td>
<td>4.19</td>
<td>2.17</td>
<td>0.39</td>
<td>1.84</td>
<td>1.36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.30</td>
<td>7.314</td>
<td>2.17</td>
<td>0.26</td>
<td>0.65</td>
<td>1.54</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ped38</strong></td>
<td>0.12</td>
<td>0.45</td>
<td>5.38</td>
<td>60.97</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(17, 59)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(582, 5)</td>
<td>128.36</td>
<td>8120.58</td>
<td>3040.60</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5946.44</td>
<td>534,828,046</td>
<td>1554.65</td>
<td>272.69</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>out</td>
<td>134.41</td>
<td>348,723</td>
<td>103.17</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1534.65</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ped50</strong></td>
<td>0.11</td>
<td>0.74</td>
<td>5.38</td>
<td>37.19</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(18, 58)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(479, 5)</td>
<td>4140.29</td>
<td>28,201,843</td>
<td>476.77</td>
<td>104.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1478.75</td>
<td>15,729,294</td>
<td>66.66</td>
<td>52.11</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>78.53</td>
<td>204,886</td>
<td>36.03</td>
<td>38.52</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5,766</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Genetic Linkage Analysis

*(Marinescu & Dechter, AAAI’07; Marinescu & Dechter, UAI’07)*

<table>
<thead>
<tr>
<th>pedigree</th>
<th>Samlam</th>
<th>Superlink</th>
<th>MBE(i)</th>
<th>MBE(i)</th>
<th>MBE(i)</th>
<th>MBE(i)</th>
<th>MBE(i)</th>
<th>MBE(i)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>v. 2.3.2</td>
<td>v. 1.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ped30</td>
<td>out</td>
<td>13095.83</td>
<td>0.31</td>
<td>0.81</td>
<td>2.66</td>
<td>8.41</td>
<td>24.88</td>
<td></td>
</tr>
<tr>
<td>(25, 51)</td>
<td></td>
<td>(w*, h)</td>
<td>5563.22</td>
<td>1397.14</td>
<td>1811.34</td>
<td>550.57</td>
<td>82.25</td>
<td></td>
</tr>
<tr>
<td>(1016, 5)</td>
<td></td>
<td>(n, d)</td>
<td>63,068,960</td>
<td>15,336,772</td>
<td>20,275,620</td>
<td>5,535,261</td>
<td>588,558</td>
<td></td>
</tr>
<tr>
<td>ped33</td>
<td>out</td>
<td>2335.28</td>
<td>0.41</td>
<td>1.08</td>
<td>5.28</td>
<td>14.45</td>
<td>51.24</td>
<td></td>
</tr>
<tr>
<td>(26, 48)</td>
<td></td>
<td>32,444,818</td>
<td>806.12</td>
<td>370.41</td>
<td>26.31</td>
<td>67.92</td>
<td>320,279</td>
<td></td>
</tr>
<tr>
<td>(581, 5)</td>
<td></td>
<td>11,403,812</td>
<td>4,032,864</td>
<td>229,856</td>
<td>219,047</td>
<td>83,360</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ped39</td>
<td>out</td>
<td>322.14</td>
<td>0.52</td>
<td>2.32</td>
<td>8.41</td>
<td>33.15</td>
<td>81.27</td>
<td></td>
</tr>
<tr>
<td>(23, 94)</td>
<td></td>
<td>4041.56</td>
<td>4041.56</td>
<td>386.13</td>
<td>386.13</td>
<td>141.23</td>
<td>407,280</td>
<td></td>
</tr>
<tr>
<td>(1272, 5)</td>
<td></td>
<td>52,804,044</td>
<td>2,171,470</td>
<td>313,496</td>
<td>313,496</td>
<td>93.19</td>
<td>83,714</td>
<td></td>
</tr>
<tr>
<td>ped42</td>
<td>out</td>
<td>561.31</td>
<td>4.20</td>
<td>31.33</td>
<td>96.28</td>
<td>68.52</td>
<td>14,479</td>
<td></td>
</tr>
<tr>
<td>(25, 76)</td>
<td></td>
<td>2364.67</td>
<td>22,595,247</td>
<td>79,356</td>
<td>79,356</td>
<td>87.63</td>
<td>14,479</td>
<td></td>
</tr>
<tr>
<td>(448, 5)</td>
<td></td>
<td>22,595,247</td>
<td>37,705</td>
<td>37,705</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Additional Metrics

- **ped30**
  - Out: 13095.83
  - Time: 0.31
  - Nodes: 2563.22
  - Out: 63,068,960
  - Out: 550.57
  - Out: 82.25

- **ped33**
  - Out: 2335.28
  - Time: 0.41
  - Nodes: 806.12
  - Out: 11,403,812
  - Out: 67.92
  - Out: 320,279

- **ped39**
  - Out: 322.14
  - Time: 0.52
  - Nodes: 4041.56
  - Out: 52,804,044
  - Out: 141.23
  - Out: 407,280

- **ped42**
  - Out: 561.31
  - Time: 4.20
  - Nodes: 2364.67
  - Out: 22,595,247
  - Out: 87.63
  - Out: 14,479
UAI’06 Results

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

- Introduction to graphical models algorithms: Inference, search and hybrids.

- AND/OR search spaces
  - AND/OR trees
  - AND/OR Graphs

- AND/OR search for combinatorial optimization
  - The mini-bucket heuristic
  - AO depth-first and best-first Branch and Bound
  - Empirical evaluation

- Current focus:
  - AND/OR Compilation
  - Approximation by Sampling and belief propagation
AOBDD vs. OBDD

AOBDD
18 nonterminals
47 arcs

OBDD
27 nonterminals
54 arcs
Recent work

- **Radu Marinescu**: Constraint optimization
  - AND/OR branch and bound for integer programming (CPAIOR 2006)
  - AO* for constraint optimization

- **Robert Mateescu (Phd 2007)**: Time-Space tradeoff schemes
  - AND/OR for mixed networks (UAI 2004)
  - AND/OR for counting (CP 2004)
  - AND/OR cutset decomposition (IJCAI 2005)
  - Bucket-elimination vs AND/OR search (UAI 2005, IJCAI 2007)
  - AND/OR compilations schemes (AOBDDs) (CP2006)
  - AND/OR compilation for weighted models and optimization (UAI, 2007, CP 2007)

- **Vibhav Gogate**: Sampling schemes for mixed networks
  - (UAI2005, IJCAI05, CP2006)
  - SampleSearch scheme, for inference and lowerbounding (AISTAT 2007, UAI 2007, AAAI 2007)

- **Boznea Bidyuk (Phd, 2006)**: w-cutset sampling, w-cutset bounding
Software

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

are available at:

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

- AND/OR search spaces are a unifying framework for search or compilation applicable to any graphical models.

- With caching AND/OR is similar to inference (context-minimal graphs)

- AND/OR time and space bounds are equal to state of the art algorithms

- Empirical results
  - AND/OR search spaces are always more effective than traditional OR spaces
  - AND/OR allows a flexible tradeoff between space and time

- Graphical models should always use AND/OR search with embedded inference.

- Current work: Hybrid of inference and search: Heuristic generation and Branch and Bound, AO cycle-cutset