Anytime Best+Depth-First Search for Bounding Marginal MAP

Radu Marinescu  
IBM Research - Ireland

Junkyu Lee, Alex Ihler and Rina Dechter  
University of California, Irvine

AAAI 2017 Technical Session: RU: Reasoning under Uncertainty  
Feb. 8th. 2017 10:00 am – 11:00 am Oral Presentation Paper 2066
Motivation and Contribution

• Marginal MAP Inference
  – Probabilistic inference query
    • Optimal partial configuration after marginalizing hidden/latent variables in a probability distribution
  – Complexity: \( \text{NP}^{\text{pp}} \) complete
  – Often it is the right task on various applications
    • Probabilistic conformant planning [Lee, Marinescu, Dechter, 2015]
    • Natural language processing task [Bird, Klein, Loper, 2009]
    • Image completion task [Xue, Li, Ermon, Gomes, Selman, 2016]

• Contributions
  – Anytime hybrid (best+depth-first) search for MMAP
  – Improvement of anytime performance for finding upper and lower bounds
    • Upper-bound: estimate of optimal solution from a partial solution
    • Lower-bound: sub-optimal solution
Outline

● Background
  - Graphical model
  - AND/OR search space & WMB heuristic
  - Previous MMAP search algorithms

● Best+Depth-First search for MMAP
  - LAOBF (Best-First AND/OR Search with Depth-First Lookaheads)
  - AAOBF (Alternating Best-First and Depth-First AND/OR search)
  - LnDFS (Learning Depth-First AND/OR search)

● Experiments

● Conclusion
Background – graphical model

- **Graphical model** \( \mathcal{M} = \langle X, D, F \rangle \)
  - variables \( X = \{ X_1, \ldots, X_n \} \)
  - domains \( D = \{ D_1, \ldots, D_n \} \)
  - functions \( F = \{ \psi_1, \ldots, \psi_r \} \)

\[
P(x) = \frac{1}{Z} \prod_{\alpha \in F} \psi_\alpha(x_\alpha)
\]

- **Marginal Map task**
  \( x^*_M = \text{argmax}_{x_M} \sum_{x_S} \prod_\alpha \psi_\alpha(x_\alpha) \)
  - \( x = x_M \cup x_S \)
  - Max and sum not commute

- **Primal graph**
  - nodes are variables
  - two nodes are connected if they appear in the same function

\[ \begin{align*}
X_M &= \{ A, B, C, D \} \\
X_S &= \{ E, F, G, H \}
\end{align*} \]
Background – AND/OR search space

- Bucket elimination [Dechter, 1999]
  
- AND/OR search graph [Mateescu, Dechter, 2007]

- Pseudo tree [Freuder, Quinn, 1985]
Background - WMB heuristics

- **Mini-bucket elimination** [Dechter, Rish 2001]
  - “i-bound”, limit on the number of variables in a single mini-bucket

- **Weighted Mini-bucket** [Liu, Ihler, 2012]
  - Holder’s inequality
    \[
    \sum_r \left( \prod_{i=1}^w \psi_{kr} \right) \leq \prod_r \left( \sum_x f(x)^{1/w} \right) w = \sum_r w_r
    \]

- **WMB Moment Matching** [Liu, Ihler, 2011]
  - MAP variables
    \[
    \psi_{kr} = \psi_{kr} \left( \frac{\mu}{\mu_r} \right)
    \]
    \[
    \mu_r = \max \psi_{kr}; \quad \mu = \left( \prod_r \mu_r \right)^{1/R}
    \]
  - SUM variables
    \[
    \psi_{kr} = \psi_{kr} \left( \frac{\mu}{\mu_r} \right)^{w_{kr}}
    \]
    \[
    \mu_r = \sum_{Y_r} \psi_{kr}^{1/w_{kr}}; \quad \mu = \prod_r (\mu_r)^{w_{kr}}
    \]

- **Mini-bucket upper bound**
  \[
  \sum_{E} \left[ \psi(A, E) \psi(C, E) \right] \leq \left[ \sum_{E} \psi(A, E) \right] \left[ \sum_{E} \psi(C, E) \right]
  \]
Previous MMAP search algorithms

- **Park, Darwiche (2003)**
  - Depth-First BnB
  - Join-tree upper bound (relaxed variable ordering)
  - depth-first search
  - dynamic heuristic

- **Marinescu, Dechter, Ihler (2014)**
  - Depth-First BnB
  - AND/OR Search
  - WMB Heuristic
  - compact AND/OR search space
  - more accurate WMB heuristics

- **Lee, Marinescu, Dechter, Ihler (2016)**
  - Weighted Best-First
  - Anytime Depth-First AND/OR
  - WMB heuristic
  - anytime solutions
  - infrequent solution updates
  - still memory intensive

- **Yuan, Hansen (2009)**
  - Depth-First BnB
  - Incremental
  - Join-tree upper bound
  - static heuristic

- **Marinescu, Dechter, Ihler (2015)**
  - Best-First/Recursive BF
  - AND/OR Search
  - WMB heuristic
  - BF avoids solving summation problems
  - very memory intensive
  - no anytime, return optimal solution or no solution

- **Marinescu, Lee, Ihler, Dechter (2017)**
  - Best+Depth-First
  - high quality upper/lower bounds
  - more frequent solution updates
  - memory efficiency
Outline

- Background
  - Graphical model
  - AND/OR search space & WMB heuristic

- Best+Depth-First search for MMAP
  - LAOBF (Best-First AND/OR Search with Depth-First Lookaheads)
  - AAOF (Alternating Best-First and Depth-First AND/OR search)
  - LnDFS (Learning Depth-First AND/OR search)

- Experiments

- Conclusion
Best+Depth-First Search

- Better guidance for depth-first dives using improved heuristics
- Frequent solution updates
- Cutoff frontier of best-first search using improved lower bounds
- Learn accurate heuristics by depth-first lookahead

When Global UB = Global LB, Optimal Solution Discovered
Notations – solution tree

MAX

OR
AND
OR
AND
OR
AND
OR
AND

partial solution tree  tip of partial solution tree  solution tree
Notations – basic operations

- \( q(n) \) : upper bound at \( n \)
- \( q(\text{root}) \) : global upper bound

- \( l(n) \) : lower bound at \( n \)
- \( l(\text{root}) \) : global lower bound

- \( T_b \) : best partial solution tree (partial solution tree where OR nodes direct the child \( m \) with best \( q(m) \))

Expand(\( n \))
- backup \( q \) and \( l \) values

Update(\( n \))
- re-direct best partial solution tree
LAOBF (best-first AND/OR search with depth-first lookaheads)

- Best-first selection
  - $T_b$
  - $Select(T_b)$

- Depth-first lookahead
  - depth-first dive at the tip of $T_b$
  - compute global lower bound
  - cache summation subproblems

- Best-first expansion & update
  - Select a tip node $n$
  - Expand and Update $n$

- Update($n$)
- Expand($n$)

Cutoff parameter: control depth-first lookahead (at every $\theta$ number of node expansions.)
AAOBF (alternating best-first with depth-first AND/OR search)

Depth-first greedy expansion $T_l$
Best-first re-direct $T_b$

Depth-first selection
Best-first selection

- Expand($n$) and Update($n$)
  $n \in T_l$
- depth-first greedy search
  $\max w(n, m)q(m)$

- redirect $T_l$ from explicated search graph from the root with updated $q$ and $l$
  $\min \frac{l(n)}{w(n, m)q(m)}$
- select $T_b$
  Expand and Update a tip node
LnDFS (learning depth-first AND/OR search)

Best-first selection

\[ T_b \]

Select\((T_b)\)

Depth-first expansion

\[ \text{Expand}(T_b) \]

Best-first update

\[ UB(T_b) \leq \text{global lower bound} \]

\[ \text{Update}(T_b) \]

Select\((T_b)\)

\[ T_b \]

\[ \text{tip}(T_b) == \text{empty} \]

\[ \text{new global lower bound!} \]

\[ \text{Update}(T_b) \]

Select\((T_b)\)

\[ T_b \]

\[ \text{Expand}(T_b) \] Keep expanding tips nodes of \( T_b \)

\[ \text{Update}(T_b) \] Update values from tip nodes of \( T_b \)
Outline

● Background
  – Graphical model
  – AND/OR search space & WMB heuristic

● Best+Depth-First search for MMAP
  – LAOBF (Best-First AND/OR Search with Depth-First Lookaheads)
  – AAOBF (Alternating Best-First and Depth-First AND/OR search)
  – LnDFS (Learning Depth-First AND/OR search)

● Experiments

● Conclusion
Experiments

• Anytime Algorithms
  – Presented Best+Depth-First Search
    • LAOBF \( \theta = 1000 \)
    • AAOBF
    • LnDFS
  – State-of-the-art
    • Weighted Recursive Best-First AND/OR Search \([\text{Lee, Marinescu, Ihler, Dechter, 2016}]\) with Overestimation \( w_{i+1} = \sqrt{w_i} \quad w_0 = 64 \)
    • Breadth Rotate AND/OR Branch and Bound \([\text{Lee, Marinescu, Ihler, Dechter, 2016}]\)
    • Anytime Factor Set Elimination \([\text{Maua, Campos, 2012}]\)

• Memory
  – total 24 GB
  – WMB-MM(i) i-bound: 20 or the largest within 4 GB
  – caching for AND/OR search graph max 4 GB
### Experiment

- **Benchmark**
  - derived from UAI inference competitions for MPE query
  - randomly choose 50% of the variables as MAP variables
  - generate 4 random MMAP instances
  - Grid, Pedigree, Promedas domain

- **Problem instance parameters**

<table>
<thead>
<tr>
<th>Domain (#. instances)</th>
<th>$N_{min}$</th>
<th>$N_{ave}$</th>
<th>$N_{max}$</th>
<th>$F_{min}$</th>
<th>$F_{ave}$</th>
<th>$F_{max}$</th>
<th>$K_{min}$</th>
<th>$K_{max}$</th>
<th>$S_{min}$</th>
<th>$S_{max}$</th>
<th>$W_{min}$</th>
<th>$W_{ave}$</th>
<th>$W_{max}$</th>
<th>$H_{min}$</th>
<th>$H_{ave}$</th>
<th>$H_{max}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grid (128)</td>
<td>144,649,2500</td>
<td>144,649,2500</td>
<td>2,2</td>
<td>3,3</td>
<td>25,163,814</td>
<td>42,189,834</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pedigree (88)</td>
<td>334,917,1289</td>
<td>334,917,1289</td>
<td>3,7</td>
<td>4,5</td>
<td>35,127,289</td>
<td>63,152,312</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Promedas (100)</td>
<td>381,1064,1997</td>
<td>385,1077,2024</td>
<td>2,2</td>
<td>3,3</td>
<td>11,137,552</td>
<td>33,171,577</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- N: number of variables, F: number of functions, K: domain size, S: scope size
- W: constrained induced width, H: constrained pseudo tree height
Experiment – individual instances

- Anytime search status for individual instances

- search: LAOBF (lab), AAOBF (aab), LnDFS (ldt), BRAOBB (bra)
- heuristic: WMB-MM (20)
- memory: 24 GB

Other algorithms couldn’t find any solution due to memory out
Experiment - average solution quality

- Average solution quality: $Ave\left[\frac{\text{best solution found}}{\text{optimal solution}}\right]$
  - anytime quality of lower bound normalized by optimal solution
  - when optimal solution is not available, used best-known solution

- Result
  - How the quality of solution improves over time
  - LAOBF, AAOBF, LnDFS
    - improved upon WRBFAOO on 3 domains
  - BRAOBB
    - best on promedas domain, second worst on pedigree domain
  - AFSE: worst performance on 3 domains
Experiment - average gap quality

- **Average gap quality**
  \[
  \text{Ave} \left[ \frac{\text{upper bound} - \text{lower bound}}{\text{upper bound}} \right]
  \]
  - anytime gap (difference between upper and lower bound) normalized by upper bound (If no lower bound available, gap = 1)

- **Result**
  - How the gap between lower/upper bound decreases over time (gap=0 optimal)
  - LAOBF, AAOBF, LnDFS
    - All similar improvements over time, especially at shorter time bounds
    - AAOBF was overall best
  - AFSE: worst performance on 3 domains
# Experiment – memory robustness

- **Memory robustness**

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$M_{out}$</th>
<th>$M_{\infty}$</th>
<th>$\Delta G$</th>
<th>$T_M$</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAOBF</td>
<td>22%</td>
<td>0%</td>
<td>7%</td>
<td>1208s</td>
</tr>
<tr>
<td>LAOBF</td>
<td>48%</td>
<td>0%</td>
<td>32%</td>
<td>1345s</td>
</tr>
<tr>
<td>WRBFAOO</td>
<td>41%</td>
<td>11%</td>
<td>42%</td>
<td>645s</td>
</tr>
<tr>
<td>LnDFS</td>
<td>0%</td>
<td>0%</td>
<td>-%</td>
<td>-s</td>
</tr>
<tr>
<td>AFSE</td>
<td>77%</td>
<td>76%</td>
<td>99.9%</td>
<td>64s</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$M_{out}$</th>
<th>$M_{\infty}$</th>
<th>$\Delta G$</th>
<th>$T_M$</th>
</tr>
</thead>
<tbody>
<tr>
<td>pedigree</td>
<td>13%</td>
<td>0%</td>
<td>6%</td>
<td>1596s</td>
</tr>
<tr>
<td></td>
<td>59%</td>
<td>5%</td>
<td>17%</td>
<td>1418s</td>
</tr>
<tr>
<td></td>
<td>53%</td>
<td>10%</td>
<td>38%</td>
<td>707s</td>
</tr>
<tr>
<td></td>
<td>1%</td>
<td>0%</td>
<td>27%</td>
<td>942s</td>
</tr>
<tr>
<td>promedas</td>
<td>0%</td>
<td>0%</td>
<td>-%</td>
<td>-s</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$M_{out}$</th>
<th>$M_{\infty}$</th>
<th>$\Delta G$</th>
<th>$T_M$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>62%</td>
<td>0%</td>
<td>77%</td>
<td>1012s</td>
</tr>
<tr>
<td></td>
<td>28%</td>
<td>28%</td>
<td>100%</td>
<td>749s</td>
</tr>
<tr>
<td></td>
<td>74%</td>
<td>74%</td>
<td>100%</td>
<td>58s</td>
</tr>
</tbody>
</table>

- How search algorithm effectively utilized the memory and improves gap within the memory limit
- $M_{out}$ $\%$ of instances terminated by memory limit
- $M_{\infty}$ $\%$ of instances terminated by memory limit and no solution found at all
- $\Delta G$ average gap computed from out of memory instances only
- $T_M$ average search time computed from out of memory instances

**Result**

- LnDFS is the most memory robust algorithm
- AAOBF (LAOBF) improved memory robustness compared to WRBFAOO
- AFSE is the worst among 5 algorithms
Conclusion

- Anytime Best+Depth-First search algorithms improved upon the state-of-the-art algorithms
  - higher quality anytime solutions
  - tighter anytime upper bounds
  - more effective use of memory

- Future work
  - New anytime search + approximate summation inference
    - variational bounds with search
    - probabilistic bounds from sampling