HEURISTIC SEARCH FOR M BEST SOLUTIONS WITH APPLICATIONS TO GRAPHICAL MODELS

Rina Dechter and Natalia Flerova
Summary:

- Optimization problem:
  - Finding the m best solutions

- Previous works on the m-best tasks:
  - Compute the m-best solutions by successively computing the best solution, each time using a slightly different reformulation of the original problem.
    - Lawler, 1972; Nilsson, 1998; Yanover and Weiss, 2004
  - Compute the m best solutions in a single pass of algorithm, using message passing/propagation
    - Seroussi and Golmard, 1994; Elliot, 2007; Rollon, Flerova, Dechter, 2011
Summary:

- **Our contribution:**
  - We develop search algorithms to compute the m-best task avoiding the repeated computation inherent in Lawler’s scheme and reducing explored state space compared to message-passing schemes.
  - In particular, we extend:
    - Best First search (specifically, A*)
    - Depth First Branch and Bound search
  - We focus on Graphical Models and apply our algorithm to the AND/OR search spaces, yielding for the m-best task:
    - AND/OR Best First search
    - AND/OR Branch and Bound
1-best vs. m-best optimization

Variables: $X = \{X_1, X_2, X_3, \ldots, X_n\}$

Finite domain values: $D = \{D_1, D_2, \ldots, D_n\}$

Objective function:

$A$ is a totally ordered set ($<$)

**1-best optimization**

$F(t)$ such that

$\forall t' \notin t. \quad F(t) \geq F(t')$

**m-best optimization**

$\{F(t_1), \ldots, F(t_m)\}$ such that

$\forall t' \notin t_j. \quad 1 \leq j \leq m. \quad F(t_j) \geq F(t')$

Example: Finding $K$ shortest paths in a graph
Background: Best First search

- **Objective function:** $f(n)$ (under-)estimates the best cost solution path passing through $n$.

$$f(n) = g(n) + h(n)$$

- **A***: $f(n) = g(n) + h(n)$

  - cost of the best path from start to $n$
  - heuristic evaluation function, underestimating $h^*(n)$, optimal cost from $n$ to goal

![Diagram](image-url)
Algorithm m-A*

- Explore nodes in Best First manner
- Continue search after finding the 1\textsuperscript{st} solution
- If the node is discovered more than once – keep at most $m$ copies both in OPEN and CLOSED
- If heuristic is not consistent – new path to the node might be better than the old one. If we already know $m$ paths to node $n$, discard the worse one.
m-A* example

\[ h(A) = 5, \quad h(B) = 4, \quad h(C) = 3, \quad h(D) = 2, \quad h(E) = 1, \quad h(F) = 1, \quad h(G) = 0 \]

\[ f(A) = 5, \quad f(B) = 8, \quad f(C) = 5, \quad f(D) = 5, \quad f(E) = 5, \quad f(F) = 8, \quad f(G) = 6 \]
**m-A* example**

- **m=2**
- **h(A)=5**
- **h(B)=4**
- **h(C)=3**
- **h(D)=2**
- **h(E)=1**
- **h(F)=1**
- **h(G)=0**

Diagram:

- Node A with edges to B and C
- Node B with edges to A and D
- Node C with edges to A and D
- Node D with edges to B, C, E, and F
- Node E with edges to D and F
- Node F with edges to D and E
- Node G with no connections

Costs and Estimates:

- **f(A)=5**
- **f(B)=8**
- **f(C)=5**
- **f(D)=5**
- **f(E)=5**
- **f(F)=8**
- **f(G)=6**

Soft 2011 - Perugia
m-A* example

\[
\begin{align*}
m & = 2 \\
h(A) & = 5 \\
h(B) & = 4 \\
h(C) & = 3 \\
h(D) & = 2 \\
h(E) & = 1 \\
h(F) & = 1 \\
h(G) & = 0 \\
f(A) & = 5 \\
f(B) & = 8 \\
f(C) & = 5 \\
f(D) & = 8 \\
f(E) & = 5 \\
f(F) & = 11 \\
f(G) & = 6 \\
m & = 2
\end{align*}
\]
m-A* example

\[
\begin{align*}
&h(A)=5 \\
&h(B)=4 \\
&h(C)=3 \\
&h(D)=2 \\
&h(E)=1 \\
&h(F)=1 \\
&h(G)=0
\end{align*}
\]

\[
\begin{align*}
m=2
\end{align*}
\]
m-A* example

\[ h(A) = 5 \quad \text{m}=2 \quad h(C) = 3 \]

\[ h(B) = 4 \quad h(D) = 2 \quad h(F) = 1 \]

\[ h(E) = 1 \quad h(G) = 0 \]

\[ f(A) = 5 \quad f(B) = 8 \quad f(C) = 5 \]

\[ f(D) = 8 \quad f(D_1) = 5 \]

\[ f(E) = 6 \quad f(E_1) = 5 \]

\[ f(F) = 11 \quad f(F_1) = 8 \]

\[ f(G) = 9 \quad f(G_1) = 6 \]

\[ f(G_2) = 9 \quad f(G_3) = 9 \]
m-A* example

\[ h(A) = 5 \]
\[ h(B) = 4 \]
\[ h(C) = 3 \]
\[ h(D) = 2 \]
\[ h(E) = 1 \]
\[ h(F) = 1 \]
\[ h(G) = 0 \]

\[ f(A) = 5 \]
\[ f(B) = 8 \]
\[ f(C) = 5 \]
\[ f(D) = 8 \]
\[ f(E) = 5 \]
\[ f(F) = 11 \]
\[ f(G) = 6 \]

\[ f(G_1) = 9 \]
\[ f(G_2) = 9 \]
\[ f(G_3) = 9 \]

\[ m = 2 \]
Properties m-A*

1. **Soundness and completeness**: m-A* terminates with the m-best solutions generated in order of their costs.

2. **Optimal efficiency**: any node that is *surely* expanded by m-A* must be expanded by any other sound and complete algorithm.

3. **Optimal efficiency for consistent heuristics**: when the heuristic function is consistent, m-A* expands each node at most \( m \) times.

4. **Dominance**: Given two heuristic functions \( h_1 \) and \( h_2 \), s.t. for every \( n \), \( h_1(n) < h_2(n) \), \( m-A^*1 \) will expand every node surely expanded by \( m-A^*2 \).
Properties m-A*

- m-A* with a consistent heuristic:
  - any node $n$ will be expanded at most $m$ times
  - the set $\{n \mid f(n) < C_m^*\}$ will surely be expanded
  - Some nodes with $\{n \mid f(n) = C_m^*\}$ are also expanded, depending on the tie breaking rule

![Diagram of search space]
Properties m-A*

- Impact of $m$ on the search space size

The difference in the search spaces explored by 1-A* and $m$-A*
Background: Depth First Branch and Bound

- Maintains the cost of the best solution found so far $U$ as an upper bound on the optimal solution.

If $f(n) \leq U$, $n$ is pruned.

$$h(A) = 5, \quad h(B) = 4, \quad h(C) = 3, \quad h(D) = 2, \quad h(E) = 1, \quad h(F) = 1, \quad h(G) = 0$$

$$u = 7, \quad f(F) = 10 + 1 = 11$$

Soft 2011 - Perugia
m-BB algorithm

- Algorithm maintains the list of m best out of the solutions found so far
- \( U_m \) – the cost of \( m^{th} \) best of them
- Prune node \( n \) if \( f(n) < U_m \)
- Given a consistent heuristic
  - m-B&B will expand all nodes such that \( E_m^* = \{ n \mid f(n) < C_m^* \} \)
  - Initially, and until m-B&B encounters the true \( m^{th} \)-best solution, it will expand also nodes \( F_m^* = \{ n \mid f(n) > C_m^* \} \) Subsequently, like m-A*, it will only expand nodes for which \( f(n) \leq C_m^* \)
m-BB algorithm

\[ m = 2 \]

\[ h(A) = 5 \]
\[ h(B) = 4 \]
\[ h(C) = 3 \]
\[ h(D) = 2 \]
\[ h(E) = 1 \]
\[ h(F) = 1 \]
\[ h(G) = 0 \]

\[ U_1 = 7; U_2 = \infty \]
m-BB algorithm

\[ m=2 \]

\[
\begin{align*}
&h(A)=5 \\
&h(B)=4 \\
&h(C)=3 \\
&h(D)=2 \\
&h(E)=1 \\
&h(F)=1 \\
&h(G)=0
\end{align*}
\]

\[ U_1 = 7; U_2 = \infty \]
m-BB algorithm

$m=2$

$U_1 = 7; U_2 = 12$
AND/OR search spaces

- Optimization problems are known to be effectively solved by searching a **weighted AND/OR search graph** or weighted AND/OR search tree.

![Graphs and Trees](image_url)
m-A* for graphical models (m-AOBF)

- The size of the **AND/OR search tree** is bound by $O(N \cdot k^h)$
- The size of the **context-minimal AND/OR search graph** is bound by $O(N \cdot k^{w*+1})$
- Applying m-A* to searching **weighted context-minimal AND/OR search graphs** yields m-AOBF.
- Each node can be expanded at most $m$ times
- The search space explored by m-AOBF is bounded by $O(N \cdot m \cdot k^{w*})$
m-BB for graphical models (m-AOBB)

- At each AND node we combine the m best solutions to the subproblems rooted in its children and pick the best m out of the combination results – time overhead of per AND node $O(\text{deg} \cdot m \cdot \log m)$

**AND/OR tree**
- Larger search space $O(N \cdot k^h)$
- Overhead due to AND nodes
- **Total run-time complexity** $O(m \cdot N \cdot k^h \cdot \text{deg} \cdot \log m)$

**AND/OR graph**
- Smaller search space $O(N \cdot k^{w*+1})$
- Overhead due to AND nodes and caching:
  - $O(\text{deg} \cdot m \cdot \log m)$ per AND node
  - $O(m \cdot \log m)$ per each subproblem cached (i.e. each OR node)
- **Total run-time complexity:** $O(m \cdot N \cdot k^{w*+1} \cdot \text{deg} \cdot \log m)$
Algorithm BE-Greedy-m-BF

- Generate exact heuristic by running Bucket Elimination along the order
- Run $m$-A* along the ordering, the algorithm will expand only nodes that lay on the $m$ best paths
- The total number of nodes expanded is $O(mN)$, where $N$ bounds the solution length
Empirical evaluation

Runtime (sec) as a function of number of solutions m for pedigree instances

- pedigree37
- pedigree38
- pedigree39

Soft 2011 - Perugia
Empirical evaluation

Runtime (sec) as a function of number of solutions m for grid instances

- grid 50-12-5
- grid 50-14-5
- grid 50-15-5
- grid 75-16-5
- grid 75-18-5

Soft 2011 - Perugia
Empirical evaluation

![Graph: Runtime (sec) as a function of number of solutions m for mastermind instances]
Empirical Evaluation

<table>
<thead>
<tr>
<th>Instance</th>
<th>N</th>
<th>k</th>
<th>w*</th>
<th>h</th>
<th>m=1</th>
<th>m=10</th>
<th>m=50</th>
<th>m=100</th>
</tr>
</thead>
<tbody>
<tr>
<td>pedigree1</td>
<td>298</td>
<td>4</td>
<td>15</td>
<td>59</td>
<td>1</td>
<td>32</td>
<td>464</td>
<td>1751</td>
</tr>
<tr>
<td>pedigree37</td>
<td>726</td>
<td>5</td>
<td>20</td>
<td>72</td>
<td>41</td>
<td>240</td>
<td>1537</td>
<td>4825</td>
</tr>
<tr>
<td>pedigree38</td>
<td>581</td>
<td>5</td>
<td>16</td>
<td>52</td>
<td>2700</td>
<td>8890</td>
<td>24894</td>
<td>t/o</td>
</tr>
<tr>
<td>pedigree39</td>
<td>953</td>
<td>5</td>
<td>20</td>
<td>77</td>
<td>2594</td>
<td>7907</td>
<td>28037</td>
<td>t/o</td>
</tr>
<tr>
<td>pedigree50</td>
<td>478</td>
<td>6</td>
<td>16</td>
<td>54</td>
<td>685</td>
<td>2835</td>
<td>21903</td>
<td>t/o</td>
</tr>
<tr>
<td>grid50-12-5</td>
<td>143</td>
<td>2</td>
<td>15</td>
<td>48</td>
<td>1</td>
<td>2</td>
<td>13</td>
<td>35</td>
</tr>
<tr>
<td>grid50-14-5</td>
<td>195</td>
<td>2</td>
<td>18</td>
<td>64</td>
<td>87</td>
<td>599</td>
<td>1995</td>
<td>3739</td>
</tr>
<tr>
<td>grid50-15-5</td>
<td>224</td>
<td>2</td>
<td>19</td>
<td>76</td>
<td>180</td>
<td>1382</td>
<td>5011</td>
<td>9591</td>
</tr>
<tr>
<td>grid75-16-5</td>
<td>256</td>
<td>2</td>
<td>21</td>
<td>73</td>
<td>539</td>
<td>3327</td>
<td>10917</td>
<td>20572</td>
</tr>
<tr>
<td>grid75-18-5</td>
<td>324</td>
<td>2</td>
<td>24</td>
<td>85</td>
<td>1147</td>
<td>4533</td>
<td>14078</td>
<td>23948</td>
</tr>
<tr>
<td>mm-03-0000</td>
<td>1220</td>
<td>2</td>
<td>18</td>
<td>43</td>
<td>43</td>
<td>580</td>
<td>4845</td>
<td>13381</td>
</tr>
<tr>
<td>mm-03-0001</td>
<td>1193</td>
<td>2</td>
<td>15</td>
<td>38</td>
<td>1</td>
<td>2</td>
<td>8</td>
<td>23</td>
</tr>
<tr>
<td>mm-03-0002</td>
<td>1193</td>
<td>2</td>
<td>15</td>
<td>39</td>
<td>1</td>
<td>3</td>
<td>17</td>
<td>48</td>
</tr>
<tr>
<td>mm-03-0004</td>
<td>1193</td>
<td>2</td>
<td>15</td>
<td>38</td>
<td>0</td>
<td>3</td>
<td>17</td>
<td>46</td>
</tr>
<tr>
<td>mm-03-0005</td>
<td>1193</td>
<td>2</td>
<td>15</td>
<td>38</td>
<td>0</td>
<td>5</td>
<td>25</td>
<td>70</td>
</tr>
<tr>
<td>mm-03-0011</td>
<td>1172</td>
<td>2</td>
<td>18</td>
<td>42</td>
<td>18</td>
<td>148</td>
<td>757</td>
<td>2016</td>
</tr>
<tr>
<td>mm-03-0014</td>
<td>1172</td>
<td>2</td>
<td>18</td>
<td>43</td>
<td>15</td>
<td>107</td>
<td>544</td>
<td>1488</td>
</tr>
<tr>
<td>mm-04-0012</td>
<td>2224</td>
<td>2</td>
<td>29</td>
<td>56</td>
<td>842</td>
<td>6859</td>
<td>31952</td>
<td>t/o</td>
</tr>
<tr>
<td>mm-04-0013</td>
<td>2224</td>
<td>2</td>
<td>29</td>
<td>58</td>
<td>375</td>
<td>4880</td>
<td>23854</td>
<td>t/o</td>
</tr>
<tr>
<td>mm-04-0014</td>
<td>2224</td>
<td>2</td>
<td>29</td>
<td>60</td>
<td>655</td>
<td>8080</td>
<td>36740</td>
<td>t/o</td>
</tr>
<tr>
<td>mm-04-0015</td>
<td>2224</td>
<td>2</td>
<td>29</td>
<td>57</td>
<td>818</td>
<td>13299</td>
<td>t/o</td>
<td>t/o</td>
</tr>
</tbody>
</table>
Conclusion:

- We present two new search-based algorithms for solving the m-best task extending known algorithm that explore AND/OR space in Best First and in Depth First manner, using heuristic information.

Future work:

- Improve the emplementation
- Perform more extensive the empirical evaluation