INFERENCE SCHEMES FOR M BEST SOLUTIONS FOR SOFT CSPS

Emma Rollón, Natalia Flerova and Rina Dechter

erollon@lsi.upc.edu, flerova@ics.uci.edu, dechter@ics.uci.edu
Universitat Politècnica de Catalunya
University of California, Irvine
Summary

- Optimization problems:
  - Finding the best solution
  - Finding the m-best solutions

- Applications of the m-best solutions:
  - Set of diverse solutions desired (e.g., haplotaping)
  - Constraints are hard to formalized (e.g., portfolio mgmt)
  - Sensitivity analysis (e.g., biological sequence alignment)
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;
Our contribution:

- We provide a formalization of the m-best task within the unifying framework of c-semiring, making many known inference schemes immediately applicable.

- In particular, we focus on Graphical Models and extend:
  - Bucket Elimination (exact algorithm)
  - Mini-Bucket Elimination (bounding algorithm)

- We show how to tighten the bound on the best solution using a bound on the m-best solutions.
1-best vs. m-best optimization

<table>
<thead>
<tr>
<th>1-best optimization</th>
<th>m-best optimization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variables: ( X = {X_1, X_2, X_3, \ldots, X_n} )</td>
<td>( X = {X_1, X_2, X_3, \ldots, X_n} )</td>
</tr>
<tr>
<td>Finite domain values: ( D = {D_1, D_2, \ldots, D_n} )</td>
<td>( D = {D_1, D_2, \ldots, D_n} )</td>
</tr>
<tr>
<td>Objective function: ( F : \bigotimes_{i=1}^{n} D_i \rightarrow A )</td>
<td>( {F(t_1), \ldots, F(t_m)} ) such that ( \forall t' \not\in t : F(t) \leq F(t') )</td>
</tr>
</tbody>
</table>

\( A \) is a totally ordered set (\( < \)) of valuations

\( F(t) \) such that \( \forall t' \not\in t : F(t) \leq F(t') \)

\( \{F(t_1), \ldots, F(t_m)\} \) such that \( \forall t' \not\in t_i, t_j \ 1 \leq i \leq j \leq m \ F(t_i) \leq F(t_j) \leq F(t') \)
Graphical Model

- $X = \{x_1, \ldots, x_n\}$: a set of variables
- $D = \{D_1, \ldots, D_n\}$: a set of domain values
- $\{f_1, \ldots, f_e\}$: a set of local functions

$$f_j : D_Y \rightarrow A$$

$Y \subseteq X$ scope of $f_j$

- $\otimes$: combination operator over functions

- Interaction graph:
Graphical Model

- Global view (objective function): \( F(X) = \bigotimes_{k=1}^{e} f_k \)

- Reasoning task: \( F(X) \downarrow_{\nabla X} \)

- Particular instantiations:

<table>
<thead>
<tr>
<th>( A )</th>
<th>( \otimes )</th>
<th>( \downarrow )</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>( NU{\infty} )</td>
<td>+</td>
<td>min</td>
<td>WCSP</td>
</tr>
<tr>
<td>[0...1]</td>
<td>( \times )</td>
<td>max</td>
<td>MPE</td>
</tr>
<tr>
<td>[0...1]</td>
<td>( \times )</td>
<td>+</td>
<td>P(e)</td>
</tr>
</tbody>
</table>
Bucket Elimination

Select a var | Combination \( \times \) | Marginalization \( \downarrow \) | Output

\[ \lambda() = \left( \bigotimes_{k=1}^{e} f_k \right) \downarrow X \]

**Complete and correct:** whenever the task can be defined over a semiring

[Shafer et. Al, Srivanas et al, Kohlas et al.]
## Bucket Elimination

<table>
<thead>
<tr>
<th>Select a var</th>
<th>Combination $\otimes$</th>
<th>Marginalization $\downarrow$</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x_1)</td>
<td>(x_1)</td>
<td>(x_5)</td>
<td>(\lambda() = c_1) = WCSP</td>
</tr>
<tr>
<td>(x_2)</td>
<td>(x_2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(x_3)</td>
<td>(x_3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(x_4)</td>
<td>(x_4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(x_5)</td>
<td>(x_5)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- \(e\) = WCSP

\(\lambda() = \left( \bigotimes_{k=1}^{e} f_k \right) |_{X} \)

Select a var \(x_1\) and eliminate it.

\(\bigotimes \equiv +\)

\(\downarrow \equiv \min\)
## Bucket Elimination

<table>
<thead>
<tr>
<th>Select a var</th>
<th>Combination $\otimes$</th>
<th>Marginalization $\downarrow$</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>$x_1$</td>
<td></td>
<td>$\lambda() = {c_1, \ldots, c_m}$ = m-best WCSP</td>
</tr>
<tr>
<td>$x_2$</td>
<td>$x_2$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$x_3$</td>
<td>$x_3$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$x_4$</td>
<td>$x_4$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$x_5$</td>
<td>$x_5$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Select a variable $x_i$ and eliminate it to simplify the problem. Combination and marginalization steps are applied to reduce the problem size. The output is the solution to the most likely solution of the WCSP problem.
Bucket Elimination

Select a var | Combination \( \otimes \) | Marginalization \( \downarrow \) | Output
---|---|---|---

\[ \lambda() = \left( e \otimes f_k \right) \downarrow_x \]

\[ \lambda() = \{ c_1, \ldots, c_m \} = \text{m-best WCSP} \]

Each tuple is the best cost extension to \( x_1 \)

Each tuple has to be the m-best cost extensions to \( x_1 \)
## From 1-best to m-best optimization

<table>
<thead>
<tr>
<th></th>
<th>WCSP</th>
<th>m-best WCSP ((m = 2))</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Functions</strong></td>
<td>(f : l(Y) \rightarrow [0...1])</td>
<td>(\tilde{f} : l(Y) \rightarrow [0...1]^m)</td>
</tr>
<tr>
<td></td>
<td>(f(t) = c_1)</td>
<td>(\tilde{f}(t) = {c_1, c_2})</td>
</tr>
<tr>
<td><strong>\times</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(f(t) = 6; g(t) = 5)</td>
<td>(\tilde{f}(t) = {3,5}; \tilde{g}(t) = {1,6})</td>
</tr>
<tr>
<td><strong>\downarrow</strong></td>
<td>(\text{min})</td>
<td>(\text{min})</td>
</tr>
<tr>
<td></td>
<td>(f(x \leftarrow a) = 6; f(x \leftarrow b) = 5)</td>
<td>(\tilde{f}(x \leftarrow a) = {3,5}; \tilde{f}(x \leftarrow b) = {1,6})</td>
</tr>
</tbody>
</table>
From 1-best to m-best optimization

<table>
<thead>
<tr>
<th></th>
<th>WCSP</th>
<th>m-best WCSP (m = 2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functions</td>
<td>$f : l(Y) \rightarrow [0...1]$</td>
<td>$\tilde{f} : l(Y) \rightarrow [0...1]^m$</td>
</tr>
<tr>
<td></td>
<td>$f(t) = c_1$</td>
<td>$\tilde{f}(t) = {c_1, c_2}$</td>
</tr>
<tr>
<td>$\times$</td>
<td>$+$</td>
<td>$+$</td>
</tr>
<tr>
<td></td>
<td>$f(t) = 6; g(t) = 5$</td>
<td>$\tilde{f}(t) = {3, 5}; \tilde{g}(t) = {1, 6}$</td>
</tr>
<tr>
<td></td>
<td>$f(t) + g(t) = 6 + 5 = 11$</td>
<td>$\tilde{f}(t) + \tilde{g}(t) = {4, 9, 6, 11} = {4, 6}$</td>
</tr>
<tr>
<td>$\downarrow$</td>
<td>min</td>
<td>min</td>
</tr>
<tr>
<td></td>
<td>$f(x \leftarrow a) = 6; f(x \leftarrow b) = 5$</td>
<td>$\tilde{f}(x \leftarrow a) = {3, 5}; \tilde{f}(x \leftarrow b) = {1, 6}$</td>
</tr>
<tr>
<td></td>
<td>$\min f = \min {5, 6} = 5$</td>
<td>$\min \tilde{f} = {3, 5, 1, 6} = {1, 3}$</td>
</tr>
</tbody>
</table>
How to combine two ordered sets

\[ S = \{1,3,6\} \quad T = \{2,4,5\} \]

\[ 1 + 2 \]
How to combine two ordered sets

\[ S = \{1, 3, 6\} \quad T = \{2, 4, 5\} \]

1 + 2

1^{st} best
How to combine two ordered sets

$S = \{1, 3, 6\}$ \hspace{1cm} $T = \{2, 4, 5\}$

$1 + 2$ \hspace{1cm} $1 + 4$ 

$1^{st}$ best
How to combine two ordered sets

\[ S = \{1,3,6\} \quad T = \{2,4,5\} \]
How to combine two ordered sets

\[ S = \{1, 3, 6\} \quad T = \{2, 4, 5\} \]

1. Combine elements from \( S \) and \( T \) to get the first best:
   \[ 1 + 2 \]

2. Combine elements from \( S \) and \( T \) to get the second best:
   \[ 1 + 4 \]

3. Combine elements from \( S \) and \( T \) to get the third best:
   \[ 3 + 2 \]

The first best combination is 1 + 2, the second best combination is 1 + 4, and the third best combination is 3 + 2.
How to combine two ordered sets

\[ S = \{1, 3, 6\} \quad \text{and} \quad T = \{2, 4, 5\} \]

Diagram:

1. 1 + 2
2. 1 + 4
3. 1 + 5
4. 3 + 2

1\textsuperscript{st} best

2\textsuperscript{nd} best
How to combine two ordered sets

\[ S = \{1, 3, 6\} \]
\[ T = \{2, 4, 5\} \]
How to combine two ordered sets

\[ S = \{1, 3, 6\} \quad T = \{2, 4, 5\} \]
How to combine two ordered sets

\[ S = \{1,3,6\} \quad T = \{2,4,5\} \]

- 1st best: \(1 + 2\)
- 2nd best: \(1 + 4\)
- 3rd best: \(3 + 2\)
- 1st best: \(1 + 5\)
- 3rd best: \(3 + 4\)
How to combine two ordered sets

$S = \{1,3,6\}$

$T = \{2,4,5\}$

1\textsuperscript{st} best

1\textsuperscript{st} best

2\textsuperscript{nd} best

2\textsuperscript{nd} best

3\textsuperscript{rd} best

3\textsuperscript{rd} best

$1 + 2$

$1 + 2$

$1 + 4$

$1 + 4$

$3 + 2$

$3 + 2$

$1 + 5$

$1 + 5$

$3 + 4$

$3 + 4$

$6 + 2$

$6 + 2$
How to combine two ordered sets

\[ S = \{1, 3, 6\} \quad T = \{2, 4, 5\} \]
How to combine two ordered sets

\[ S = \{1, 3, 6\} \quad T = \{2, 4, 5\} \]

\[ O(m^2) \quad \Rightarrow \quad O(m \times \log(m+1)) \]
## Bucket Elimination

<table>
<thead>
<tr>
<th>Select a var</th>
<th>Combination ( \times )</th>
<th>Marginalization ( \downarrow )</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_1 )</td>
<td>( x_1 )</td>
<td>( x_1 )</td>
<td></td>
</tr>
<tr>
<td>( x_5 )</td>
<td>( x_5 )</td>
<td>( x_5 )</td>
<td></td>
</tr>
<tr>
<td>( x_2 )</td>
<td>( x_2 )</td>
<td>( x_2 )</td>
<td></td>
</tr>
<tr>
<td>( x_4 )</td>
<td>( x_4 )</td>
<td>( x_4 )</td>
<td></td>
</tr>
<tr>
<td>( x_3 )</td>
<td>( x_3 )</td>
<td>( x_3 )</td>
<td></td>
</tr>
</tbody>
</table>

\[ \lambda() = \left( \bigotimes_{k=1}^{e} f_k \right) \downarrow_X \]

\[ \lambda() = c_1 = \text{WCSP} \]

\[ \lambda() = \{c_1, \ldots, c_m\} = \text{m-best WCSP} \]
Correct and complete: the m-best problem can be formulated as a **commutative semiring** using the new operators.
A commutative semiring is a triplet \((A, \otimes, \oplus)\), where operators satisfy three axioms:

- A1. The operation \(\oplus\) is associative, commutative and idempotent, and there is an additive identity element called 0 such that \(a \oplus 0 = a\) for all \(a \in A\).
- A2. The operation \(\otimes\) is also associative and commutative, and there is a multiplicative identity element called 1 such that \(a \otimes 1 = a\) for all \(a \in A\).
- A3. \(\otimes\) distributes over \(\oplus\), i.e., \((a \otimes b) \oplus (a \otimes c) = a \otimes (b \oplus c)\).

Example: MPE task is defined over semiring \(K = (R, \times, \max)\), a CSP is defined over semiring \(K = (\{0, 1\}, \land, \lor)\), and a Weighted CSP is defined over semiring \(K = (\mathbb{N} \cup \{\infty\}, +, \min)\).

It was showed that the correctness of inference algorithms over a reasoning task \(P\) is ensured whenever \(P\) is defined over a semiring. [Shafer et. Al, Srivanaras et al, Kohlas et al.]
m-space and semirings

Consider an optimization problem: $F : \bigotimes_{i=1}^{n} D_i \to A$

Let $S$ be a subset of a set of valuation $A$.

We define the set of ordered $m$-best elements of $S$, $S^m = \{s_1, ..., s_j\}$ such that $s_1 \leq s_2 \leq ... \leq s_j$ where $j = m$ if $|S| \geq m$ and $j = |S|$ otherwise, and $\forall S \not\subset S^m, s_j \leq s$

**m-space of $A$:** denoted $\overline{A}$, is the set of subsets of ordered $m$-best elements of $A$.

Operators $+$ and $\min$ described above are defined over m-space.

**Theorem 1:** the triplet $(\overline{A}, +, \min)$ is a semiring, defining the $m$-best WCSP task.
## Bucket Elimination

<table>
<thead>
<tr>
<th>Select a var</th>
<th>Combination $\odot$</th>
<th>Marginalization $\downarrow$</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td></td>
<td></td>
<td>$\lambda() = {c_1, \ldots, c_m}$ = m-best WCSP</td>
</tr>
<tr>
<td>$x_5$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$x_2$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$x_4$</td>
<td>$\odot$ $\equiv$ $+$</td>
<td>$\downarrow$ $\equiv$ $\min$</td>
<td></td>
</tr>
<tr>
<td>$x_3$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$x_5$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$x_2$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$x_4$</td>
<td>$\odot$ $\equiv$ $+$</td>
<td>$\downarrow$ $\equiv$ $\min$</td>
<td></td>
</tr>
<tr>
<td>$x_3$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Time:** $O(\exp(w^*+1) \cdot n \cdot m \cdot \log(m+1))$

**Space:** $O(\exp(w^*+1) \cdot n \cdot m)$
## Mini-Bucket Elimination

<table>
<thead>
<tr>
<th>Select a var</th>
<th>Combination ( \otimes )</th>
<th>Marginalization ( \downarrow )</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MBE((z = 3))</strong></td>
<td><img src="image1" alt="Diagram" /></td>
<td><img src="image2" alt="Diagram" /></td>
<td><img src="image3" alt="Diagram" /></td>
</tr>
</tbody>
</table>

\[ \lambda() \geq \left( \bigotimes_{k=1}^{e} f_k \right) \downarrow X \]
Mini-Bucket Elimination

Select a var | Combination $\otimes$ | Marginalization $\Downarrow$ | Output

$\lambda() = \{a_1, \ldots, a_m\}$

$\lambda() = a_1 \leq \text{Cost(WCSP)}$

$\lambda() \geq \left( \bigotimes_{k=1}^e f_k \right) \downarrow X$

Time: $O(\exp(z+1) \cdot n)$  
Space: $O(\exp(z) \cdot n)$

MBE($z = 3$)

mbe-opt

Time: $O(\exp(z+1) \cdot n \cdot \log(m+1))$

Space: $O(\exp(z+1) \cdot n \cdot m)$

mbe-m-opt
The m-best WCSP is the set \( \{c_1, c_2, c_3, \ldots, c_k\} \), where each cost \( c_i \) is less than or equal to the next one:

\[
c_1 \leq c_2 \leq c_3 \leq \cdots \leq c_k
\]
m-best WCSP = \{c_1, c_2, c_3, \ldots, c_k\}

\[ c_1 \leq c_2 \leq c_3 \leq \ldots \leq c_k \]
**mbe-m-opt Output**

\[ m\text{-best WCSP} = \{c_1, c_2, c_3, \ldots, c_k\} \]

\[
c_1 \leq c_2 \leq c_3 \leq \ldots \leq c_k
\]

\[ \lambda() = \{c_1, c_2, c_3, \ldots, c_k\} \]
**mb-m-opt Output**

**m-best WCSP**

\[ m\text{-best WCSP} = \{c_1, c_2, c_3, \ldots, c_k\} \]

\[ c_1 \leq c_2 \leq c_3 \leq \ldots \leq c_k \]

\[ \lambda() = \{a_1, a_2, a_3, c_1, a_4, a_5, c_2, a_6, a_7, a_8, \ldots, a_j, c_3, \ldots, a_{j+l}, \ldots, c_k\} \]

\[ a_1 \leq a_2 \leq a_3 \leq c_1 \leq a_4 \leq \ldots \leq c_k \]
mb-m-opt Output

\[ \text{m-best WCSP} = \{c_1, c_2, c_3, \ldots, c_k\} \]
\[ c_1 \leq c_2 \leq c_3 \leq \ldots \leq c_k \]

\[ \lambda() = \{a_1, a_2, a_3, c_1, a_4, a_5, c_2\} \]
\[ a_1 \geq a_2 \geq a_3 \geq c_1 \geq \ldots \geq c_k \]
mbe-m-opt Output

$m$-best WCSP = \{c_1, c_2, c_3, \ldots, c_k\}

c_1 \leq c_2 \leq c_3 \leq \ldots \leq c_k

\lambda() = \{a_1, a_2, a_3\}

a_1 \geq a_2 \geq a_3 \geq c_1 \geq \ldots \geq c_k
Empirical Evaluation

- Benchmarks (UAI 2008 competition):
  - Linkage analysis networks (pedigree)
  - Grid networks
  - WCSPs

- Algorithm: mbe-m-opt($z = 10$)

- Evaluate:
  - The runtime as a function of the number of solutions $m$
  - The improvement of the 1-bound as a function of $m$
Runtime as a function of m

Pedigrees: run-time as a function of m

- Pedigree13
- Pedigree19
- Pedigree20
- Pedigree23
- Pedigree37
- Pedigree38
- Pedigree39
- Pedigree41
Runtime as a function of m
Bound’s improvement as a function of $m$

The index of the solution

log(UB)

-pedigree20-
Bound’s improvement as a function of $m$

WCSP instances
Conclusions

- We presented two new bucket elimination algorithms for solving the m-best task by extending the combination and marginalization operators.

- The same extension yields a general formalization of m-best task over a semiring and can be used for:
  - solving other optimization problems
  - applying other known inference algorithms to m-best task.

- Future work:
  - Improve the empirical evaluation.
  - Investigate an extension of the loopy-belief propagation for the m-best task.
  - Investigate the use of bounds on m-best solutions as possible heuristics.
INFERENCE SCHEMES FOR $M$ BEST SOLUTIONS FOR SOFT CSPS

Emma Rollón, Natalia Flerova and Rina Dechter

erollon@lsi.upc.edu, flerova@ics.uci.edu, dechter@ics.uci.edu
Universitat Politècnica de Catalunya
University of California, Irvine