2011

Reformulating $R(\ast,m)C$ with Tree Decomposition

Shant Karakashian  
*University of Nebraska-Lincoln*, shantk@cse.unl.edu

Robert J. Woodward  
*University of Nebraska-Lincoln*, rwoodwar@cse.unl.edu

Berthe Y. Choueiry  
*University of Nebraska-Lincoln*, choueiry@cse.unl.edu

Follow this and additional works at: [http://digitalcommons.unl.edu/cseconfwork](http://digitalcommons.unl.edu/cseconfwork)  
Part of the [Computer Sciences Commons](http://digitalcommons.unl.edu/cseconfwork)

Karakashian, Shant; Woodward, Robert J.; and Choueiry, Berthe Y., "Reformulating $R(\ast,m)C$ with Tree Decomposition" (2011). CSE Conference and Workshop Papers. 187.  
[http://digitalcommons.unl.edu/cseconfwork/187](http://digitalcommons.unl.edu/cseconfwork/187)
Reformulating R(*,m)C with Tree Decomposition

Shant Karakashian, Robert J. Woodward, Berthe Y. Choueiry

Constraint Systems Laboratory
University of Nebraska-Lincoln

Acknowledgements
• Experiments conducted at UNL’s Holland Computing Center
• NSF Grant No. RI-111795
Outline

• Introduction
• R(*,m)C Property & Algorithm
• Exploit Tree Decomposition to
  – Avoid useless update & reduce propagation effort
    ↪ Update queue: PROCESSQ ↭ PROCESSMQ
    ↪ The two algorithms *yield the same filtering*
  – Synthesize & add new constraints to improve propagation
    ↪ Property enforced: R(*,m)C ↭ T-R(*,m,z)C
    ↪ The same algorithm *yields stronger filtering*

• Experimental Results
• Conclusion
Constraint Satisfaction Problem

- CSP
  - Variables ($\mathcal{V}$), domains
  - Constraints: relations ($\mathcal{R}$), scope
- Representation
  - Hypergraph
  - Primal graph
  - Dual graph
- Solved with
  - Search
  - Enforcing consistency
- Warning
  - Consistency property vs. algorithms
Tree Decomposition

- **Tree**: Vertices-clusters, edges
- **Each cluster is labeled with**
  - A set of variables $\subseteq V$
  - A set of relations $\subseteq R$
- **Two conditions**
  1. For each relation $R$, $\exists$ cluster $c_i$
     - $R$ appears $c_i$
     - $\text{Scope}(R)$ is also in $c_i$
  2. Every variable
     - Induces a connected subtree
- **Separators**
  - Variables & relations common to 2 adjacent clusters
  - channel communications between clusters
R(*,m)C Property

• A CSP is R(*,m)C iff
  – Every **tuple** in a relation can be extended to the variables in the scope of any \((m-1)\) other relations in an assignment satisfying all \(m\) relations simultaneously

\[\forall m-1\text{ relations}\]
ProcessQ: Algorithm for R(*,m)C

• \( \Phi \): combination of \( m \) connected relations in the dual graph
  \[
  \Phi = \{ \omega_1 = \{R_1, R_2, \ldots, R_m\}, \omega_2, \omega_3, \ldots, \omega_k \}
  \]

• Q propagation queue
  \[
  Q = \{\langle R_1, \omega_1 \rangle, \langle R_1, \omega_2 \rangle, \langle R_1, \omega_3 \rangle, \ldots, \langle R_n, \omega_{k-1} \rangle, \langle R_n, \omega_k \rangle \}
  \]

• For each \( \langle R_i, \omega_j \rangle \) in Q, ProcessQ
  – Deletes from \( R_i \) tuples that cannot extended to relations in \( \omega_j \)
  – As some tuples of relations \( R_x \in \omega_j \) may lose support, it requeues \( \{\langle R_x, \omega_y \rangle\} \) for every threatened relation
ProcessQ: Animation

Define CSP $P_\omega$

For each $\tau$ in $R$

Assign $\tau$ as a value for $R$

Solve $P_\omega$ with forward checking

If no solution found: delete $\tau$

Add $\langle R', \omega' \rangle$ to Q: $R_i \neq R'$, $R_i \in \omega'$ and $R' \in \omega'$

Extract $\langle R, \omega \rangle$ from Q

Q

$\langle R_1, \omega_1 \rangle$

$\langle R_2, \omega_1 \rangle$

$\langle R_5, \omega_1 \rangle$

$\langle R_2, \omega_2 \rangle$

$\langle R_5, \omega_2 \rangle$

$\langle R_4, \omega_2 \rangle$

$\langle R_3, \omega_3 \rangle$

$\langle R_4, \omega_3 \rangle$

$\langle R_5, \omega_3 \rangle$
ProcessMQ: Intelligent update scheduling

- Cluster $c_i$ has a local queue $Q(c_i) = \{\langle R_i, \omega \rangle \}$ for relations $R_i$ in cluster but not in parent.
- Using the tree decomposition
  - As an ordering heuristic for checking consistency of $\langle R_i, \omega \rangle$
  - Repeat “leaves up to root, down to leaves,” until quiescence
  - Update relations in only local queue
  - Example: $R_3$ is updated only when root is reached
- Advantage fewer updates, same filtering
  - In previous example, $R_3$ is updated once although it appears in 3 clusters
**T-R(*,m,z)C**

[Rollon+ 10]

Hypergraph

Primal graph

Dual graph

Tree decomposition

Adding $R_5$
T-R(*,m,z)C Strictly Stronger than R(*,m)C

Let A, B, C, D and E be Boolean variables

<table>
<thead>
<tr>
<th></th>
<th>R₁</th>
<th>R₂</th>
<th>R₃</th>
<th>R₄</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>D 0</td>
<td>A 0</td>
<td>B 0</td>
<td>E 0</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>D</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>E</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Assignment A=0 & E=1 is valid
Does not violate R(*,2)C

<table>
<thead>
<tr>
<th></th>
<th>R₁</th>
<th>R₂</th>
<th>R₃</th>
<th>R₄</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>E 0</td>
<td>A 0</td>
<td>B 0</td>
<td>E 0</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>D</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>E</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Assignment A=0 & E=1 is **inconsistent**
Experimental Results

- Experiments for finding all solutions with BTD maintaining $wR(*,\text{best}(2,3,4))C$ and $T-wR(*,\text{best}(2,3,4), \text{best}(5,7,9))$
- Results shown demonstrate the benefits of $\text{ProcessMQ}$ & $T-wR(*,m,z)C$

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>#ins</th>
<th>#vars</th>
<th>tw</th>
<th>ProcessQ $wR(*,\text{best})C$</th>
<th>ProcessMQ $wR(*,\text{best})C$</th>
<th>ProcessQ $T-wR(*,b,b)C$</th>
</tr>
</thead>
<tbody>
<tr>
<td>aim-200</td>
<td>24</td>
<td>200</td>
<td>104.92</td>
<td>17</td>
<td>17</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$t_{\text{avg}}$</td>
<td>$t_{\text{avg}}$</td>
<td>$t_{\text{max}}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>246.35</td>
<td>252.48</td>
<td>238.99</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3,352.54</td>
<td>3,452.98</td>
<td>1,540.94</td>
</tr>
<tr>
<td>ogdVg</td>
<td>59</td>
<td>134</td>
<td>85</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$t_{\text{avg}}$</td>
<td>$t_{\text{avg}}$</td>
<td>$t_{\text{max}}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>283.27</td>
<td>242.06</td>
<td>266.74</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1,834.11</td>
<td>1,508.27</td>
<td>1,720.97</td>
</tr>
<tr>
<td>rand-3-20-20</td>
<td>50</td>
<td>20</td>
<td>13</td>
<td>13</td>
<td>14</td>
<td>(not shown)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$t_{\text{avg}}$</td>
<td>$t_{\text{avg}}$</td>
<td>$t_{\text{max}}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2,191.56</td>
<td>1,949.87</td>
<td>(not shown)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3,481.04</td>
<td>3,145.77</td>
<td>(not shown)</td>
</tr>
</tbody>
</table>
Conclusions

• Contributions
  – Reformulated $R(\ast, m)C$ algorithm
  – New relational consistency property $T-R(\ast, m, z)C$
  – Experimental analysis

• Future work
  – Study impact of choice of parameters $z, m$
  – Develop strategies for dynamically choosing $z, m$
    as a function of the size of clusters & separators