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/computersciences)
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: \text{PROCESSQ} \rightsquigarrow \text{PROCESSMQ}
    ↩ The two algorithms \textit{yield the same filtering}
  – Synthesize & add new constraints to improve propagation
    ↩ Property enforced: R(\*,m)C \rightsquigarrow T-R(\*,m,z)C
    ↩ The same algorithm \textit{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 \( \subseteq \mathcal{V} \)
  - A set of relations \( \subseteq \mathcal{R} \)
- **Two conditions**
  1. For each relation \( R \), \( \exists \) cluster \( c_i \)
     - \( R \) appears \( c_i \)
     - 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 [Karakashian+ 10]

• 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

\[
\forall \text{ tuple} \implies \forall \text{ relation} \implies \forall m-1 \text{ relations}
\]
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
For each τ in R
Assign τ as a value for R
Solve $P_{\omega}$ with forward checking
Extract $\langle R, \omega \rangle$ from Q
Define CSP $P_{\omega}$
For each τ in R
Assign τ as a value for R
Solve $P_{\omega}$ with forward checking
If no solution found: delete τ
Add $\langle R', \omega' \rangle$ to Q: $R_i \neq R'$, $R_i \in \omega'$ and $R' \in \omega'$
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

Hypergraph

Primal graph

Dual graph

Tree decomposition

Adding $R_5$

[Rollon+ 10]
T-R(*,m,z)C Strictly Stronger than R(*,m)C

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

Assignment $A=0 \land E=1$ is valid

Does not violate $R(*,2)C$

Assignment $A=0 \land E=1$ is inconsistent
Experimental Results

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

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>#ins</th>
<th>#vars</th>
<th>tw</th>
<th>ProcessQ $wR(*)$ &amp; $wR(best)C$</th>
<th>ProcessMQ $wR(*)$ &amp; $wR(best)C$</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>$t_{\text{max}}$</td>
<td>$t_{\text{max}}$</td>
</tr>
<tr>
<td></td>
<td>246.35</td>
<td>3,352.54</td>
<td>3,352.54</td>
<td>252.48</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>283.27</td>
<td>1,834.11</td>
<td>1,834.11</td>
<td>242.06</td>
<td>1,508.27</td>
<td>266.74</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>14</td>
</tr>
<tr>
<td></td>
<td>2,191.56</td>
<td>3,481.04</td>
<td>3,481.04</td>
<td>1,949.87</td>
<td>3,145.77</td>
<td>1,720.97</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