Improving Constraint Modelling Using Visualization

Helmut Simonis

Cork Constraint Computation Centre
Computer Science Department
University College Cork
Ireland

JFPC 2010
Visualization can play major role in CP modelling
This is not new, but neglected
We can do this more systematically
Visualization is not for tool developers
Not enough to develop tools, we need advice how to use and interpret them
There are other issues we rarely address in CP
  How do I pick best model without implementing all alternatives?
  Is there such a thing as the best model?
  Questions, examples, but no answers here!
Methodology

- Based on 3 case studies
  - Almost square packing
  - Rooms problem, sports scheduling
  - Tight production scheduling system
- Each solves a non-trivial problem
- Each shows a different aspect of the problem
Example 1

- Canonical model: Beldiceanu & Contejean, 94
- Naive search impractical
- Static variable selection, from largest to smallest item
- No value choice preference
- Complex search, phased (interleaved) assignment
- Very large search space explored
Example 1: Almost Square Packing

- Consider almost squares, rectangles $n \times (n + 1)$
- Pack all items $1 \times 2$, $2 \times 3$, ..., $n \times (n + 1)$
- Into smallest possible rectangle
- Items are non-overlapping
Background

- Suggested by Prof. MacHale, Math, UCC
- Solved packing up to 13x14 by hand
- Starting point for Barry’s and my interest in packing
- My objection: “This is too complicated”
  - Look at square packing
  - Known results, benchmark comparison
Simpler Problem: Square Packing

- Consider squares, rectangles $n \times n$
- Pack all items $1 \times 1, 2 \times 2, \ldots, n \times n$
- Into smallest possible rectangle
- Items are non-overlapping
Outline

1. Square Packing
2. Search Strategies
3. Results
4. Almost Square Packing
Problem (N=26)
Search for candidate enclosing rectangle
Area must be larger than sum of items to be placed
Search in order of increasing area
  and increasing “squareness”
Check each candidate for (in)feasibility until first solution is found
Observation: Only limited number of candidates explored
Candidates

Square Packing
Search Strategies
Results
Almost Square Packing

Candidates

35 40 45 50 55 60 65 70 75 80
6200 6220 6240 6260 6280 6300 6320 6340 6360 6380 6400

Optimal

Helmut Simonis Modelling
Basic Model
Outline

1. Square Packing
2. Search Strategies
3. Results
4. Almost Square Packing
Alternatives

- naive
- x then y
- disjunctive
- semantic disjunctive
- dual
- interval
- split
- xy interval
Interval Based Strategies

- Key Idea: Fixing intervals, not values
- Fixing variables to values is too restrictive
- Select “area” in which item is placed
- Allows items to shift slightly
- Restrict domain to intervals
- Only at end fix actual values
Forcing Obligatory Parts

Small Interval

Large Interval

Number of intervals: large

Number of intervals: small

Obligatory parts

\( X_l \) \( X_r \)

\( X_l \) \( X_r \)
Variants

- **(X) Interval**
  - Split all $X$ variables into intervals
  - Then fix $X$ values
  - Then treat all $Y$ variables the same way

- **Split**
  - Split $X$ variables into intervals
  - Split $Y$ variables into intervals
  - Then fix values

- **XY Interval**
  - For each item, split $X$ and $Y$ variables into intervals
  - Then fix values
Model Improvement: Dominance Criterion

\[ X_2 \geq X_1 \]

\[ X_2 + W_2 \leq X_1 + W_1 \]

\[ Y_2 + H_2 \leq Y_1 + H_1 \]

\[ Y_2 \geq Y_1 \]
Outline

1 Square Packing
2 Search Strategies
3 Results
4 Almost Square Packing
Strategies Comparison

![Graph showing time vs. problem size for different search strategies.](image-url)
### Strategy Comparison

<table>
<thead>
<tr>
<th>N</th>
<th>naive</th>
<th>xtheny</th>
<th>disj</th>
<th>semantic4</th>
<th>semantic</th>
<th>dual</th>
<th>interval</th>
<th>split</th>
<th>xy</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>2.92</td>
<td>0.09</td>
<td>12.12</td>
<td>0.55</td>
<td>0.45</td>
<td>2.63</td>
<td>-</td>
<td>0.05</td>
<td>-</td>
</tr>
<tr>
<td>16</td>
<td>10.44</td>
<td>0.11</td>
<td>98.25</td>
<td>1.31</td>
<td>1.03</td>
<td>0.89</td>
<td>-</td>
<td>0.05</td>
<td>-</td>
</tr>
<tr>
<td>17</td>
<td>20.75</td>
<td>0.27</td>
<td>23.57</td>
<td>1.48</td>
<td>1.13</td>
<td>0.33</td>
<td>0.05</td>
<td>0.05</td>
<td>0.81</td>
</tr>
<tr>
<td>18</td>
<td>667.33</td>
<td>18.37</td>
<td>-</td>
<td>30.53</td>
<td>23.05</td>
<td>118.58</td>
<td>1.83</td>
<td>1.13</td>
<td>13.94</td>
</tr>
<tr>
<td>19</td>
<td>4140.09</td>
<td>13.73</td>
<td>-</td>
<td>83.42</td>
<td>63.25</td>
<td>80.66</td>
<td>1.11</td>
<td>1.88</td>
<td>36.78</td>
</tr>
<tr>
<td>20</td>
<td>-</td>
<td>13.08</td>
<td>-</td>
<td>216.07</td>
<td>167.61</td>
<td>149.79</td>
<td>2.14</td>
<td>1.47</td>
<td>108.28</td>
</tr>
<tr>
<td>21</td>
<td>-</td>
<td>143.72</td>
<td>-</td>
<td>1138.98</td>
<td>865.13</td>
<td>-</td>
<td>8.09</td>
<td>10.59</td>
<td>619.45</td>
</tr>
<tr>
<td>22</td>
<td>-</td>
<td>1708.89</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>52.21</td>
<td>32.36</td>
<td>1668.59</td>
</tr>
<tr>
<td>23</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>245.07</td>
<td>265.54</td>
<td>9521.73</td>
</tr>
<tr>
<td>24</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>452.73</td>
<td>545.82</td>
<td>37506.20</td>
</tr>
<tr>
<td>25</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>2533.64</td>
<td>4127.41</td>
<td>-</td>
</tr>
<tr>
<td>26</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>14158.15</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>27</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>43529.87</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Optimal Solution (N=27, CP 2008)

Even better results by R. Korf in IJCAI 2009
Outline

1. Square Packing
2. Search Strategies
3. Results
4. Almost Square Packing
Apply lessons learned to almost square packing
- Added degree of freedom, rotation of items
- Weak impact, length only changes by one
- But $2^n$ additional choices
Expectations: What is feasible?

- We did find optimal solution for square packing up to 27
- Show of hands: Which problem size can we handle for almost squares?
- With $2^n$ additional choices
Almost Square, Optimal Solution (N=26)
## Results

<table>
<thead>
<tr>
<th>N</th>
<th>Surface</th>
<th>K</th>
<th>Width</th>
<th>Height</th>
<th>Area</th>
<th>Loss</th>
<th>Back</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>20</td>
<td>1</td>
<td>4</td>
<td>5</td>
<td>20</td>
<td>0.00</td>
<td>2</td>
<td>00:00</td>
</tr>
<tr>
<td>4</td>
<td>40</td>
<td>1</td>
<td>4</td>
<td>10</td>
<td>40</td>
<td>0.00</td>
<td>2</td>
<td>00:00</td>
</tr>
<tr>
<td>5</td>
<td>70</td>
<td>1</td>
<td>5</td>
<td>14</td>
<td>70</td>
<td>0.00</td>
<td>2</td>
<td>00:00</td>
</tr>
<tr>
<td>6</td>
<td>112</td>
<td>3</td>
<td>6</td>
<td>19</td>
<td>114</td>
<td>1.79</td>
<td>5</td>
<td>00:00</td>
</tr>
<tr>
<td>7</td>
<td>168</td>
<td>3</td>
<td>12</td>
<td>14</td>
<td>168</td>
<td>0.00</td>
<td>6</td>
<td>00:00</td>
</tr>
<tr>
<td>8</td>
<td>240</td>
<td>4</td>
<td>15</td>
<td>16</td>
<td>240</td>
<td>0.00</td>
<td>5</td>
<td>00:00</td>
</tr>
<tr>
<td>9</td>
<td>330</td>
<td>6</td>
<td>14</td>
<td>24</td>
<td>336</td>
<td>1.82</td>
<td>20</td>
<td>00:00</td>
</tr>
<tr>
<td>10</td>
<td>440</td>
<td>6</td>
<td>17</td>
<td>26</td>
<td>442</td>
<td>0.45</td>
<td>199</td>
<td>00:00</td>
</tr>
<tr>
<td>11</td>
<td>572</td>
<td>3</td>
<td>22</td>
<td>26</td>
<td>572</td>
<td>0.00</td>
<td>55</td>
<td>00:00</td>
</tr>
<tr>
<td>12</td>
<td>728</td>
<td>8</td>
<td>21</td>
<td>35</td>
<td>735</td>
<td>0.96</td>
<td>340</td>
<td>00:00</td>
</tr>
<tr>
<td>13</td>
<td>910</td>
<td>3</td>
<td>26</td>
<td>35</td>
<td>910</td>
<td>0.00</td>
<td>1218</td>
<td>00:00</td>
</tr>
<tr>
<td>14</td>
<td>1120</td>
<td>4</td>
<td>28</td>
<td>40</td>
<td>1120</td>
<td>0.00</td>
<td>1695</td>
<td>00:00</td>
</tr>
<tr>
<td>15</td>
<td>1360</td>
<td>4</td>
<td>34</td>
<td>40</td>
<td>1360</td>
<td>0.00</td>
<td>8169</td>
<td>00:00</td>
</tr>
<tr>
<td>16</td>
<td>1632</td>
<td>4</td>
<td>32</td>
<td>51</td>
<td>1632</td>
<td>0.00</td>
<td>5362</td>
<td>00:00</td>
</tr>
<tr>
<td>17</td>
<td>1938</td>
<td>3</td>
<td>34</td>
<td>57</td>
<td>1938</td>
<td>0.00</td>
<td>124959</td>
<td>00:07</td>
</tr>
<tr>
<td>18</td>
<td>2280</td>
<td>4</td>
<td>30</td>
<td>76</td>
<td>2280</td>
<td>0.00</td>
<td>6711</td>
<td>00:00</td>
</tr>
<tr>
<td>19</td>
<td>2660</td>
<td>4</td>
<td>35</td>
<td>76</td>
<td>2660</td>
<td>0.00</td>
<td>15887</td>
<td>00:01</td>
</tr>
<tr>
<td>20</td>
<td>3080</td>
<td>4</td>
<td>35</td>
<td>88</td>
<td>3080</td>
<td>0.00</td>
<td>188708</td>
<td>00:15</td>
</tr>
<tr>
<td>21</td>
<td>3542</td>
<td>5</td>
<td>39</td>
<td>91</td>
<td>3549</td>
<td>0.20</td>
<td>7336903</td>
<td>09:31</td>
</tr>
<tr>
<td>22</td>
<td>4048</td>
<td>3</td>
<td>44</td>
<td>92</td>
<td>4048</td>
<td>0.00</td>
<td>20073166</td>
<td>26:19</td>
</tr>
<tr>
<td>23</td>
<td>4600</td>
<td>3</td>
<td>40</td>
<td>115</td>
<td>4600</td>
<td>0.00</td>
<td>7543034</td>
<td>12:37</td>
</tr>
<tr>
<td>24</td>
<td>5200</td>
<td>3</td>
<td>40</td>
<td>130</td>
<td>5200</td>
<td>0.00</td>
<td>11368115</td>
<td>18:23</td>
</tr>
<tr>
<td>25</td>
<td>5850</td>
<td>5</td>
<td>45</td>
<td>130</td>
<td>5850</td>
<td>0.00</td>
<td>15147701</td>
<td>28:44</td>
</tr>
<tr>
<td>26</td>
<td>6552</td>
<td>5</td>
<td>42</td>
<td>156</td>
<td>6552</td>
<td>0.00</td>
<td>35672212</td>
<td>01:04:14</td>
</tr>
</tbody>
</table>
Visualization: Search Tree (N=20)
Problem with Tree View

- Showing complete tree is clearly infeasible
- We only show path to solution
- But most time is spent in non-solution parts of tree
Search Choices

- Based on best method for square packing
- Assign X intervals, fix X values, assign Y intervals, fix Y values
- When to fix orientation?
  - Eager: Before assigning X intervals
  - Lazy: After assigning X intervals
  - Interleaved: Mixed with X interval assignment
Eager Orientation (N=17)
Lazy Orientation (N=17)
Interleaved Orientation (N=17)
Comparison (N=17)
Node Distribution (N=20)
Problem with Slack (N=21)

The graph shows the number of nodes against depth for the problem 21_46_77. The data file is named "flat_21_46_77.dat".
Impact of Interval Size (N=26)
Lessons Learned, Future Work

- Little hope for further improvement of search for perfect problems
- Potential for problems with slack, ignore smaller items
- Need better reasoning
  - Dominance criterion for multiple areas at the same time
  - Dominance criterion for partially fixed items
Problem 2: Sports Scheduling

- A Nero Wolfe Mystery: Too Many Models
- How do we choose a good model
- No search at all
Outline

5 Rooms Problem

6 Modelling

7 Selected Model
Sports Scheduling

Tournament Planning

We plan a tournament with 8 teams, where every team plays every other team exactly once. The tournament is played on 7 days, each team playing on each day. The games are scheduled in 7 venues, and each team should play in each venue exactly once.

As part of the TV arrangements, some preassignments are done: We may either fix the game between two particular teams to a fixed day and venue, or only state that some team must play on a particular day at a given venue. The objective is to complete the schedule, so that all constraints are satisfied.
## Example

<table>
<thead>
<tr>
<th></th>
<th>City 1</th>
<th>City 2</th>
<th>City 3</th>
<th>City 4</th>
<th>City 5</th>
<th>City 6</th>
<th>City 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day 1</td>
<td></td>
<td>8</td>
<td></td>
<td></td>
<td>7, 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 2</td>
<td>2</td>
<td>1, 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 3</td>
<td>7</td>
<td></td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 4</td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>5</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Day 5</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Day 6</td>
<td></td>
<td></td>
<td></td>
<td>5, 4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 7</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td>1, 3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Solution

<table>
<thead>
<tr>
<th></th>
<th>City 1</th>
<th>City 2</th>
<th>City 3</th>
<th>City 4</th>
<th>City 5</th>
<th>City 6</th>
<th>City 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day 1</td>
<td></td>
<td>6, 8</td>
<td></td>
<td>1, 2</td>
<td>5, 7</td>
<td></td>
<td>3, 4</td>
</tr>
<tr>
<td>Day 2</td>
<td>2, 3</td>
<td></td>
<td></td>
<td></td>
<td>4, 8</td>
<td>6, 7</td>
<td></td>
</tr>
<tr>
<td>Day 3</td>
<td>1, 7</td>
<td>2, 4</td>
<td>3, 8</td>
<td></td>
<td></td>
<td></td>
<td>5, 6</td>
</tr>
<tr>
<td>Day 4</td>
<td></td>
<td></td>
<td>4, 7</td>
<td>2, 6</td>
<td>3, 5</td>
<td>1, 8</td>
<td></td>
</tr>
<tr>
<td>Day 5</td>
<td>5, 8</td>
<td>3, 6</td>
<td></td>
<td></td>
<td>1, 4</td>
<td>2, 7</td>
<td></td>
</tr>
<tr>
<td>Day 6</td>
<td></td>
<td>3, 7</td>
<td>1, 6</td>
<td>4, 5</td>
<td></td>
<td>2, 8</td>
<td></td>
</tr>
<tr>
<td>Day 7</td>
<td>4, 6</td>
<td></td>
<td>2, 5</td>
<td>7, 8</td>
<td>1, 3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Rooms Puzzle, (Thomas G. Room, 1955)

Place numbers 1 to 8 in cells so that each row and each column has each number exactly once, each cell contains either no numbers or two numbers (which must be different from each other), and each combination of two different numbers appears in exactly one cell.
Rooms Puzzle, (Thomas G. Room, 1955)

Place numbers 1 to 8 in cells so that each row and each column has each number exactly once, each cell contains either no numbers or two numbers (which must be different from each other), and each combination of two different numbers appears in exactly one cell.

Puzzle presented by R. Finkel
Outline

5 Rooms Problem

6 Modelling
   • Exploring Ideas
   • Expanding Idea 7
   • Comparing Ideas
   • Channeling

7 Selected Model
How to come up with a model

- What are the variables/what are their values?
- How can we express the constraints?
- Do we have these constraints in our system?
- Does this do good propagation?
- Backtrack to earlier step as required
Requirements

1. There are 8 teams, seven days and seven locations
2. Each team plays each other team exactly once
3. Each team plays 7 games (redundant)
4. Each team plays in each location exactly once
5. Each team plays on each day exactly once
6. A game consists of two (different) teams
7. There are four games on each day (redundant)
8. There are four games at each location (redundant)
9. In any location there is atmost one game at a time
Matrix Day $\times$ Game $(7 \times 4)$
- Each cell contains two variables, denoting teams
- Easy to say that team plays once on each day, \textit{alldifferent}
- Columns don’t have significance
- Model does not mention location, how to add this?
- How to express that each team plays each other once?
Idea 2, Change problem structure

- Matrix of Day × Location (7 × 7)
- Each cell contains two variables, each denoting a team
- How do we avoid symmetry inside cell?
- Need special value (0) to denote that there is no game
- In one cell, either both or none of the variables are 0
- Easy to say that each row and column contains each team exactly once
- Except for value 0, can not use alldifferent
- Link between two variables in cell to state that game needs two different teams
- How to express that each (ordered) pair occurs exactly once?
Idea 3, Add location variables

- Model as in Idea 1, matrix $Day \times Game$
- Each cell contains two variables for teams and one for location
- Easy to state that games on one day are in different locations
- How to express condition that each team plays in each location once?
- Also, how to express that each team plays each other exactly once?
Idea 4, Use variables for pairs

- **Matrix** \( \text{Day} \times \text{Location} \)
- Each cell contains one variable ranging over (sorted) pairs of teams, and special value 0 (no game)
- Each pair value occurs once, except for 0
  - Special constraint \text{alldifferent}0
  - Or use \text{gcc}
- How to state that each team plays once per day?
- How to state that each team plays in each location?
Idea 5: If all else fails, use binary variables

- Binary variable stating that team $i$ plays in location $j$ at day $k$
- Three dimensional matrix
- Each team plays once on each day
- Each team plays once in each location
- Each game has two (different) teams, needs auxiliary variable
- Each pair of team meets once, needs auxiliary variables
Idea 6: An even bigger binary model

- Use four dimensions
- Team $i$ meets team $j$ in location $k$ on day $l$
- $3136 = 8*8*7*7$ variables
- Constraints all linear
- Why use finite domain constraints?
Idea 7: A different mapping

- Each team plays each other exactly once, one variable for each combination (8*7/2=28 variables)
- Decide when and where this game is played, values range over combinations of days and locations (7*7=49 values)
- All variables must be different (no two games at same time and location)
- Each team plays 7 games, by construction
- How to express that each team plays once per day?
- How to express that each team plays in each location once?
Expand Idea 7 into Full Model
## Numbering Values

<table>
<thead>
<tr>
<th></th>
<th>City 1</th>
<th>City 2</th>
<th>City 3</th>
<th>City 4</th>
<th>City 5</th>
<th>City 6</th>
<th>City 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day 1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>Day 2</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>Day 3</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
</tr>
<tr>
<td>Day 4</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
</tr>
<tr>
<td>Day 5</td>
<td>29</td>
<td>30</td>
<td>31</td>
<td>32</td>
<td>33</td>
<td>34</td>
<td>35</td>
</tr>
<tr>
<td>Day 6</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>39</td>
<td>40</td>
<td>41</td>
<td>42</td>
</tr>
<tr>
<td>Day 7</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
<td>48</td>
<td>49</td>
</tr>
</tbody>
</table>
Four games on each day

- Day 1 corresponds to values 1..7
- Four variables can take these values
- Day 2 corresponds to values 8..14, etc
- One constraint per day
- Exactly four of all variables take their value in the set ...
- Seven such constraints

<table>
<thead>
<tr>
<th></th>
<th>City 1</th>
<th>City 2</th>
<th>City 3</th>
<th>City 4</th>
<th>City 5</th>
<th>City 6</th>
<th>City 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day 1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>Day 2</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>Day 3</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
</tr>
<tr>
<td>Day 4</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
</tr>
<tr>
<td>Day 5</td>
<td>29</td>
<td>30</td>
<td>31</td>
<td>32</td>
<td>33</td>
<td>34</td>
<td>35</td>
</tr>
<tr>
<td>Day 6</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>39</td>
<td>40</td>
<td>41</td>
<td>42</td>
</tr>
<tr>
<td>Day 7</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
<td>48</td>
<td>49</td>
</tr>
</tbody>
</table>
Four games at each location

- City 1 corresponds to values
  - 1, 8, 15, 22, 29, 36, 43
- Four variables can take these values
- City 2 corresponds to values
  - 2, 9, 16, 23, 30, 37, 44
- One constraint per location
- Exactly four of all variables take their value in the set ...
- Seven such constraints over 28 variables each

<table>
<thead>
<tr>
<th></th>
<th>City 1</th>
<th>City 2</th>
<th>City 3</th>
<th>City 4</th>
<th>City 5</th>
<th>City 6</th>
<th>City 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day 1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>Day 2</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>Day 3</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
</tr>
<tr>
<td>Day 4</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
</tr>
<tr>
<td>Day 5</td>
<td>29</td>
<td>30</td>
<td>31</td>
<td>32</td>
<td>33</td>
<td>34</td>
<td>35</td>
</tr>
<tr>
<td>Day 6</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>39</td>
<td>40</td>
<td>41</td>
<td>42</td>
</tr>
<tr>
<td>Day 7</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
<td>48</td>
<td>49</td>
</tr>
</tbody>
</table>
Teams plays once on a day (at a location)

- Select those variables which correspond to Team $i$
- Exactly one of those variables takes its value in the set 1..7
- Same for all other days
- Same for all other teams
- 56 Constraints over 7 variables each
- Similar for teams and locations, another 56 constraints
Are we there yet?

- 28 variables with 49 possible values
- 1 alldifferent
- 7 exactly constraints over all variables (Days)
- 7 exactly constraints over all variables (Locations)
- 56 exactly constraints over 7 variables each (Days)
- 56 exactly constraints over 7 variables each (Locations)
Idea 8: Mapping games to days and locations

- For each game to be played, we have two variables
  - One ranges over the days
  - The other over the locations
- Easy to state that there are four games per day and location
- Easy to state that each team plays once per day and location
- How do we express that no two games are played at the same location and the same time?
  - If we had an \texttt{alldifferent} over pairs of variables...
  - Not in ECLiPSe
We have four games on each day

- Each row value is taken four times amongst the variables
  \[ \text{gcc}([\text{gcc}(4,4,1),\ldots,\text{gcc}(4,4,7)], \text{Rows}) \]

- Similar for columns:
  \[ \text{gcc}([\text{gcc}(4,4,1),\ldots,\text{gcc}(4,4,7)], \text{Cols}) \]
Reminder: \( gcc(Pattern, Variables) \)

- \( gcc \) *global cardinality constraint*
- *Pattern* is list of terms \( gcc(Low, High, Value) \)
- The overall number of variables taking value *Value* is between *Low* and *High*
- Generalization of *alldifferent*
- Domain consistent version in ECLiPSe
Each team plays once per day

- For the seven variables which describe games of a team
- Each row value is taken exactly once amongst the variables
- Could use
  \[
  \text{gcc}([\text{gcc}(1,1,1),\ldots,\text{gcc}(1,1,7)],\text{Vars})
  \]
- But `alldifferent(Vars)` is more compact
- Similar for columns
How do the models differ?

<table>
<thead>
<tr>
<th>Idea</th>
<th>Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$D \times G \times {f, s} \rightarrow T$</td>
</tr>
<tr>
<td>2</td>
<td>$D \times L \times {f, s} \rightarrow T \cup {0}$</td>
</tr>
</tbody>
</table>
| 3    | $D \times G \times \{f, s\} \rightarrow T$  
          $D \times G \rightarrow L$ |
| 4    | $D \times L \rightarrow T \triangle T \cup \{0\}$ |
| 5    | $T \times D \times L \rightarrow \{0, 1\}$ |
| 6    | $T \times T \times D \times L \rightarrow \{0, 1\}$ |
| 7    | $T \triangle T \rightarrow D \times L$ |
| 8    | $T \triangle T \rightarrow D$  
          $T \triangle T \rightarrow L$ |

D  Days  
T  Teams  
L  Locations  
G  Games
<table>
<thead>
<tr>
<th>Idea</th>
<th>Main point</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>missing locations, first second symmetry</td>
</tr>
<tr>
<td>2</td>
<td>spare value, first second symmetry</td>
</tr>
<tr>
<td>3</td>
<td>first second symmetry</td>
</tr>
<tr>
<td>4</td>
<td>spare value</td>
</tr>
<tr>
<td>5</td>
<td>0/1, non-linear constraints</td>
</tr>
<tr>
<td>6</td>
<td>0/1, large matrix</td>
</tr>
<tr>
<td>7</td>
<td>needs exactly constraint</td>
</tr>
<tr>
<td>8</td>
<td>needs alldifferent on tuples</td>
</tr>
</tbody>
</table>
Viewpoints and Channeling

- Instead of expressing all constraints over one set of variables
- Use multiple sets of variables (*viewpoints*)
- Decide which constraint to express over which variables
- Allows more freedom on how to express problem
- Link the different variables with *channeling* constraints
In Our Case

- Combine ideas 7 and 8
- One set of variables ranging over pairs
- Another using two variables per game for day and location
- How to combine variables?
- Minimize loss of information
Projection

- Link pair variables to row and column variables
- Pair variable uses cell numbers 1-49 as values
- Row and column variables indicate on which day (row) and in which location (column) the game is played
- Pair value \(23 = \) row 4, column 2
- element constraint to link the variables
- Two projections from \(D \times L\) space onto \(D\) and \(L\)

<table>
<thead>
<tr>
<th>Day 1</th>
<th>City 1</th>
<th>City 2</th>
<th>City 3</th>
<th>City 4</th>
<th>City 5</th>
<th>City 6</th>
<th>City 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day 2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>Day 3</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>Day 4</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
</tr>
<tr>
<td>Day 5</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
</tr>
<tr>
<td>Day 6</td>
<td>29</td>
<td>30</td>
<td>31</td>
<td>32</td>
<td>33</td>
<td>34</td>
<td>35</td>
</tr>
<tr>
<td>Day 7</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>39</td>
<td>40</td>
<td>41</td>
<td>42</td>
</tr>
<tr>
<td>Day 7</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
<td>48</td>
<td>49</td>
</tr>
</tbody>
</table>
Channeling Constraints

- This is one common type, a *projection*
- Another common type is the *inverse*
  - Link a variable $A \rightarrow B$ to another $B \rightarrow A$
  - Typically used for bijective mappings
  - Built-in `inverse/2`
- Also used: *Boolean* channeling
  - Link variables $A \rightarrow B$ and $A \times B \rightarrow \{0, 1\}$
  - Built-in `bool_channeling/3`
Outline

5 Rooms Problem

6 Modelling

7 Selected Model
   • Redundant Modelling
Two sets of variables (Req 1, 2, 3, 6, by construction)

Pair variables \((T \triangle T \rightarrow D \times L)\)
- \texttt{alldifferent} (Req 9)

Day and Location variables \((T \triangle T \rightarrow D), (T \triangle T \rightarrow L)\)
- \texttt{gcc} (Req 4, 5)
- \texttt{alldifferent} (Req 7, 8)

Channeling Constraints
- \texttt{element} projection from pairs onto rows and columns

Search only on pair variables
Req 1: There are 8 teams, seven days and seven locations

- Two sets of variables (Req 1, 2, 3, 6, by construction)
- Pair variables ($T \triangle T \rightarrow D \times L$)
  - $\text{alldifferent}$ (Req 9)
- Day and Location variables ($T \triangle T \rightarrow D), (T \triangle T \rightarrow L$)
  - $\text{gcc}$ (Req 4, 5)
  - $\text{alldifferent}$ (Req 7, 8)
- Channeling Constraints
  - element projection from pairs onto rows and columns
- Search only on pair variables
Selected Model

Req 2: Each team plays each other team exactly once

- Two sets of variables (Req 1, 2, 3, 6, by construction)
- Pair variables \((T \triangle T \rightarrow D \times L)\)
  - `alldifferent` (Req 9)
- Day and Location variables \((T \triangle T \rightarrow D), (T \triangle T \rightarrow L)\)
  - `gcc` (Req 4, 5)
  - `alldifferent` (Req 7, 8)
- Channeling Constraints
  - `element` projection from pairs onto rows and columns
- Search only on pair variables
Selected Model

Req 3: Each team plays 7 games

- Two sets of variables (Req 1, 2, 3, 6, by construction)
- Pair variables \( T \triangle T \rightarrow D \times L \)
  - allDifferent (Req 9)
- Day and Location variables \( T \triangle T \rightarrow D \), \( T \triangle T \rightarrow L \)
  - gcc (Req 4, 5)
  - allDifferent (Req 7, 8)
- Channeling Constraints
  - element projection from pairs onto rows and columns
- Search only on pair variables
Selected Model

Req 4: Each team plays in each location exactly once

- Two sets of variables (Req 1, 2, 3, 6, by construction)
- Pair variables \((T \Delta T \rightarrow D \times L)\)
  - \texttt{alldifferent} (Req 9)
- Day and Location variables \((T \Delta T \rightarrow D), (T \Delta T \rightarrow L)\)
  - \texttt{gcc} (Req 4, 5)
  - \texttt{alldifferent} (Req 7, 8)
- Channeling Constraints
  - \texttt{element} projection from pairs onto rows and columns
- Search only on pair variables
Selected Model

Req 5: Each team plays on each day exactly once

- Two sets of variables (Req 1, 2, 3, 6, by construction)
- Pair variables \((T \triangle T \rightarrow D \times L)\)
  - \textit{alldifferent} (Req 9)
- Day and Location variables \((T \triangle T \rightarrow D), (T \triangle T \rightarrow L)\)
  - \textit{gcc} (Req 4, 5)
  - \textit{alldifferent} (Req 7, 8)
- Channeling Constraints
  - \textit{element} projection from pairs onto rows and columns
- Search only on pair variables
Req 6: A game consists of two (different) teams

- Two sets of variables (Req 1, 2, 3, 6, by construction)
- Pair variables \((T \triangle T \rightarrow D \times L)\)
  - `alldifferent` (Req 9)
- Day and Location variables \((T \triangle T \rightarrow D), (T \triangle T \rightarrow L)\)
  - `gcc` (Req 4, 5)
  - `alldifferent` (Req 7, 8)
- Channeling Constraints
  - `element` projection from pairs onto rows and columns
- Search only on pair variables
Selected Model

Req 7: There are four games on each day

- Two sets of variables (Req 1, 2, 3, 6, by construction)
- Pair variables \((T \triangle T \rightarrow D \times L)\)
  - \texttt{alldifferent} (Req 9)
- Day and Location variables \((T \triangle T \rightarrow D), (T \triangle T \rightarrow L)\)
  - \texttt{gcc} (Req 4, 5)
  - \texttt{alldifferent} (Req 7, 8)
- Channeling Constraints
  - \texttt{element} projection from pairs onto rows and columns
- Search only on pair variables
Req 8: There are four games at each location

- Two sets of variables (Req 1, 2, 3, 6, by construction)
- Pair variables \((T \triangle T \rightarrow D \times L)\)
  - `alldifferent` (Req 9)
- Day and Location variables \((T \triangle T \rightarrow D), (T \triangle T \rightarrow L)\)
  - `gcc` (Req 4, 5)
  - `alldifferent` (Req 7, 8)
- Channeling Constraints
  - `element` projection from pairs onto rows and columns
- Search only on pair variables
Req 9: In any location there is at most one game at a time

- Two sets of variables (Req 1, 2, 3, 6, by construction)
- Pair variables \( (T \triangle T \rightarrow D \times L) \)
  - `alldifferent` (Req 9)
- Day and Location variables \( (T \triangle T \rightarrow D), (T \triangle T \rightarrow L) \)
  - `gcc` (Req 4, 5)
  - `alldifferent` (Req 7, 8)
- Channeling Constraints
  - `element` projection from pairs onto rows and columns
- Search only on pair variables
Handling of hints (I)

<table>
<thead>
<tr>
<th>Day 1</th>
<th>City 1</th>
<th>City 2</th>
<th>City 3</th>
<th>City 4</th>
<th>City 5</th>
<th>City 6</th>
<th>City 7</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td>7, 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 2</td>
<td>2</td>
<td>1, 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 3</td>
<td>7</td>
<td></td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 4</td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>5</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Day 5</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Day 6</td>
<td></td>
<td></td>
<td></td>
<td>5, 4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 7</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td>1, 3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- This value (17) cannot be used by pairs not involving team 8.
- One of the pairs involving team 8 must use this value (17).
### Handling of hints (I)

<table>
<thead>
<tr>
<th></th>
<th>City 1</th>
<th>City 2</th>
<th>City 3</th>
<th>City 4</th>
<th>City 5</th>
<th>City 6</th>
<th>City 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day 1</td>
<td></td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td>7, 5</td>
<td></td>
</tr>
<tr>
<td>Day 2</td>
<td>2</td>
<td>1, 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 3</td>
<td>7</td>
<td></td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 4</td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>5</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Day 5</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Day 6</td>
<td></td>
<td></td>
<td></td>
<td>5, 4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 7</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1, 3</td>
<td></td>
</tr>
</tbody>
</table>

- This value (17) cannot be used by pairs not involving team 8.
- One of the pairs involving team 8 must use this value (17).
# Handling of hints (I)

<table>
<thead>
<tr>
<th></th>
<th>City 1</th>
<th>City 2</th>
<th>City 3</th>
<th>City 4</th>
<th>City 5</th>
<th>City 6</th>
<th>City 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day 1</td>
<td></td>
<td>8</td>
<td></td>
<td></td>
<td>7, 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 2</td>
<td>2</td>
<td>1, 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 3</td>
<td>7</td>
<td></td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 4</td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>5</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Day 5</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Day 6</td>
<td></td>
<td></td>
<td>5, 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 7</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td>1, 3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- This value (17) can not be used by pairs not involving team 8
- One of the pairs involving team 8 must use this value (17)
The pair involving teams 5 and 7 must take value 5, fixes variable
The pair involving teams 5 and 7 must take value 5, fixes variable
Before Search

Values

Vars
Solution
Missing Propagation

<table>
<thead>
<tr>
<th>Day</th>
<th>City 1</th>
<th>City 2</th>
<th>City 3</th>
<th>City 4</th>
<th>City 5</th>
<th>City 6</th>
<th>City 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day 1</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td>7, 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 2</td>
<td>2, 1, 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 3</td>
<td>7</td>
<td></td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 4</td>
<td></td>
<td>2</td>
<td>5</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 5</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 6</td>
<td></td>
<td></td>
<td>5, 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 7</td>
<td>4</td>
<td></td>
<td></td>
<td>1, 3</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Day</th>
<th>City 1</th>
<th>City 2</th>
<th>City 3</th>
<th>City 4</th>
<th>City 5</th>
<th>City 6</th>
<th>City 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day 1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>Day 2</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>Day 3</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
</tr>
<tr>
<td>Day 4</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
</tr>
<tr>
<td>Day 5</td>
<td>29</td>
<td>30</td>
<td>31</td>
<td>32</td>
<td>33</td>
<td>34</td>
<td>35</td>
</tr>
<tr>
<td>Day 6</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>39</td>
<td>40</td>
<td>41</td>
<td>42</td>
</tr>
<tr>
<td>Day 7</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
<td>48</td>
<td>49</td>
</tr>
</tbody>
</table>
Missing Propagation

<table>
<thead>
<tr>
<th>City 1</th>
<th>City 2</th>
<th>City 3</th>
<th>City 4</th>
<th>City 5</th>
<th>City 6</th>
<th>City 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day 1</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>7, 5</td>
</tr>
<tr>
<td>Day 2</td>
<td>2</td>
<td>1, 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 3</td>
<td>7</td>
<td></td>
<td></td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 4</td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Day 5</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Day 6</td>
<td></td>
<td></td>
<td>5, 4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 7</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1, 3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>City 1</th>
<th>City 2</th>
<th>City 3</th>
<th>City 4</th>
<th>City 5</th>
<th>City 6</th>
<th>City 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day 1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>Day 2</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
</tr>
<tr>
<td>Day 3</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
</tr>
<tr>
<td>Day 4</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
</tr>
<tr>
<td>Day 5</td>
<td>29</td>
<td>30</td>
<td>31</td>
<td>32</td>
<td>33</td>
<td>34</td>
</tr>
<tr>
<td>Day 6</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>39</td>
<td>40</td>
<td>41</td>
</tr>
<tr>
<td>Day 7</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
<td>48</td>
</tr>
</tbody>
</table>
### Missing Propagation

<table>
<thead>
<tr>
<th>City 1</th>
<th>City 2</th>
<th>City 3</th>
<th>City 4</th>
<th>City 5</th>
<th>City 6</th>
<th>City 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day 1</td>
<td>8</td>
<td></td>
<td></td>
<td>7, 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 2</td>
<td>2</td>
<td>1, 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 3</td>
<td>7</td>
<td></td>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 4</td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Day 5</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Day 6</td>
<td></td>
<td></td>
<td></td>
<td>5, 4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 7</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1, 3</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>City 1</th>
<th>City 2</th>
<th>City 3</th>
<th>City 4</th>
<th>City 5</th>
<th>City 6</th>
<th>City 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day 1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>Day 2</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
</tr>
<tr>
<td>Day 3</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
</tr>
<tr>
<td>Day 4</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
</tr>
<tr>
<td>Day 5</td>
<td>29</td>
<td>30</td>
<td>31</td>
<td>32</td>
<td>33</td>
<td>34</td>
</tr>
<tr>
<td>Day 6</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>39</td>
<td>40</td>
<td>41</td>
</tr>
<tr>
<td>Day 7</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
<td>48</td>
</tr>
</tbody>
</table>
Missing Propagation

<table>
<thead>
<tr>
<th>Day</th>
<th>City 1</th>
<th>City 2</th>
<th>City 3</th>
<th>City 4</th>
<th>City 5</th>
<th>City 6</th>
<th>City 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day 1</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td>7, 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 2</td>
<td>2, 1, 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 3</td>
<td>7</td>
<td></td>
<td></td>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 4</td>
<td></td>
<td>2</td>
<td>5</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 5</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 6</td>
<td></td>
<td></td>
<td>5, 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 7</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1, 3</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Day 1</th>
<th>City 1</th>
<th>City 2</th>
<th>City 3</th>
<th>City 4</th>
<th>City 5</th>
<th>City 6</th>
<th>City 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day 1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>Day 2</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>Day 3</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
</tr>
<tr>
<td>Day 4</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
</tr>
<tr>
<td>Day 5</td>
<td>29</td>
<td>30</td>
<td>31</td>
<td>32</td>
<td>33</td>
<td>34</td>
<td>35</td>
</tr>
<tr>
<td>Day 6</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>39</td>
<td>40</td>
<td>41</td>
<td>42</td>
</tr>
<tr>
<td>Day 7</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
<td>48</td>
<td>49</td>
</tr>
</tbody>
</table>
Missing Propagation

<table>
<thead>
<tr>
<th>Day</th>
<th>City 1</th>
<th>City 2</th>
<th>City 3</th>
<th>City 4</th>
<th>City 5</th>
<th>City 6</th>
<th>City 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day 1</td>
<td></td>
<td>8</td>
<td></td>
<td></td>
<td>7, 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 2</td>
<td>2</td>
<td>1, 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 3</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Day 5</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Day 6</td>
<td></td>
<td></td>
<td></td>
<td>5, 4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 7</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1, 3</td>
<td></td>
</tr>
</tbody>
</table>
### Missing Propagation

<table>
<thead>
<tr>
<th>Day</th>
<th>City 1</th>
<th>City 2</th>
<th>City 3</th>
<th>City 4</th>
<th>City 5</th>
<th>City 6</th>
<th>City 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day 1</td>
<td></td>
<td>8</td>
<td></td>
<td></td>
<td>7, 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 2</td>
<td>2</td>
<td>1, 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 3</td>
<td>7</td>
<td></td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 4</td>
<td></td>
<td></td>
<td>2</td>
<td>5</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 5</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 6</td>
<td></td>
<td></td>
<td>5, 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 7</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1, 3</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Day</th>
<th>City 1</th>
<th>City 2</th>
<th>City 3</th>
<th>City 4</th>
<th>City 5</th>
<th>City 6</th>
<th>City 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day 1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>Day 2</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>Day 3</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
</tr>
<tr>
<td>Day 4</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
</tr>
<tr>
<td>Day 5</td>
<td>29</td>
<td>30</td>
<td>31</td>
<td>32</td>
<td>33</td>
<td>34</td>
<td>35</td>
</tr>
<tr>
<td>Day 6</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>39</td>
<td>40</td>
<td>41</td>
<td>42</td>
</tr>
<tr>
<td>Day 7</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
<td>48</td>
<td>49</td>
</tr>
</tbody>
</table>
Missing Propagation

<table>
<thead>
<tr>
<th>Day</th>
<th>City 1</th>
<th>City 2</th>
<th>City 3</th>
<th>City 4</th>
<th>City 5</th>
<th>City 6</th>
<th>City 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day 1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>Day 2</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>Day 3</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
</tr>
<tr>
<td>Day 4</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
</tr>
<tr>
<td>Day 5</td>
<td>29</td>
<td>30</td>
<td>31</td>
<td>32</td>
<td>33</td>
<td>34</td>
<td>35</td>
</tr>
<tr>
<td>Day 6</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>39</td>
<td>40</td>
<td>41</td>
<td>42</td>
</tr>
<tr>
<td>Day 7</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
<td>48</td>
<td>49</td>
</tr>
</tbody>
</table>
Why is this?

- Constraints involved:
  - \( g_{CC} \) constraint on row: four variables can use values from this row
  - four occurrence constraints for hints: One of the variables must take this value
- No interaction between constraints, only between constraints and variables
- We do not detect that value 1 can not be used
- Eventual solution respects condition, model is correct
- We are concerned about propagation, not just correctness
Adding Redundant Constraints

- Add constraints which do more propagation, but do not affect solutions
- Lead to smaller search tree, hopefully faster solution
- Introduction requires understanding of (lack of) propagation
- Visualization is key to detect missing propagation
First Attempt: Adding 0/1 Viewpoint

- *Day* × *Location* matrix of 0/1 variables
- Indicates if there is a game on this day at this location
- Row/column sums: 4 games in each row/column
- Hint given for cell: Game variable is 1
Channeling Constraint

- Link pair variables $P_i$ to 0/1 variables $Y_j$ as value-index
- $(\exists i \text{ s.t. } P_i = v) \iff Y_v = 1$

Propagation:
- $P_i = v \implies Y_v = 1$
- $Y_v = 0 \implies \forall i : P_i \neq v$
- $(\forall i : v \notin d(P_i)) \implies Y_v = 0$
- $Y_v = 1 \implies \text{occurrence}(V, P_1 \ldots P_n, N), N \geq 1$
Before Search
Impact of Redundant Constraints

Without

With value index channeling
Search tree with redundant constraints
Search tree with redundant constraints
Search tree with redundant constraints
Search tree with redundant constraints

12
3
14
28 34
16
Search tree with redundant constraints
Search tree with redundant constraints
Search tree with redundant constraints
Search tree with redundant constraints
Search tree with redundant constraints

- Nodes: 12, 14, 25, 28, 34, 35
- Edges: 3, 4, 16

Redundant Modelling
Search tree with redundant constraints

- Rooms Problem
- Modelling
- Selected Model
- Redundant Modelling

Helmut Simonis
Modelling

Back to Start
Skip Animation
Search tree with redundant constraints
Search tree with redundant constraints
Search tree with redundant constraints
Search tree with redundant constraints
Search tree with redundant constraints
Search tree with redundant constraints
Search tree with redundant constraints
Solution
Search Tree
Game 12 cannot be played on day 1 at locations 5 or 6.
Still Missing Propagation

<table>
<thead>
<tr>
<th>City 1</th>
<th>City 2</th>
<th>City 3</th>
<th>City 4</th>
<th>City 5</th>
<th>City 6</th>
<th>City 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day 1</td>
<td>8</td>
<td></td>
<td></td>
<td>7, 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 2</td>
<td>2</td>
<td>1, 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 3</td>
<td>7</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 4</td>
<td></td>
<td></td>
<td>2</td>
<td>5</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Day 5</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Day 6</td>
<td></td>
<td>5, 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 7</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td>1, 3</td>
<td></td>
</tr>
</tbody>
</table>

Game 12 can not be played on day 1 at locations 5 or 6
Game 12 can not be played on day 1 at locations 5 or 6
Game 12 can not be played on day 1 at locations 5 or 6.
Game 12 can not be played on day 1 at locations 5 or 6
Still Missing Propagation

Game 12 can not be played on day 1 at locations 5 or 6
Our model does not deal well with hints

- Preset game is ok, leads to variable assignment
- Preset team is weak, adds new constraint
- As there is no interaction of this constraint with the other constraints, there is no initial domain restriction
- Model is correct, but lazy
## Second Attempt: Improving the handling of hints

<table>
<thead>
<tr>
<th></th>
<th>City 1</th>
<th>City 2</th>
<th>City 3</th>
<th>City 4</th>
<th>City 5</th>
<th>City 6</th>
<th>City 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day 1</td>
<td></td>
<td>8</td>
<td></td>
<td></td>
<td>7, 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 2</td>
<td>2</td>
<td>1, 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 3</td>
<td>7</td>
<td></td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 4</td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>5</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Day 5</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Day 6</td>
<td></td>
<td></td>
<td>5, 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 7</td>
<td>4</td>
<td></td>
<td></td>
<td>1, 3</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- This value cannot be used by pairs not involving team 8
- One of the pairs involving team 8 must use this value
- These values cannot be used by any pair involving team 8
## Second Attempt: Improving the handling of hints

<table>
<thead>
<tr>
<th></th>
<th>City 1</th>
<th>City 2</th>
<th>City 3</th>
<th>City 4</th>
<th>City 5</th>
<th>City 6</th>
<th>City 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day 1</td>
<td></td>
<td>8</td>
<td></td>
<td></td>
<td>7, 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 2</td>
<td>2</td>
<td>1, 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 3</td>
<td>7</td>
<td></td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Day 5</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Day 6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5, 4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 7</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td>1, 3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- This value cannot be used by pairs not involving team 8
- One of the pairs involving team 8 must use this value
- These values cannot be used by any pair involving team 8
### Second Attempt: Improving the handling of hints

<table>
<thead>
<tr>
<th></th>
<th>City 1</th>
<th>City 2</th>
<th>City 3</th>
<th>City 4</th>
<th>City 5</th>
<th>City 6</th>
<th>City 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day 1</td>
<td></td>
<td>8</td>
<td></td>
<td></td>
<td>7, 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 2</td>
<td>2</td>
<td>1, 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 3</td>
<td>7</td>
<td></td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 4</td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>5</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Day 5</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Day 6</td>
<td></td>
<td></td>
<td></td>
<td>5, 4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 7</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td>1, 3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- This value can not be used by pairs not involving team 8
- One of the pairs involving team 8 must use this value
- These values can not be used by any pair involving team 8
## Second Attempt: Improving the handling of hints

<table>
<thead>
<tr>
<th>Day</th>
<th>City 1</th>
<th>City 2</th>
<th>City 3</th>
<th>City 4</th>
<th>City 5</th>
<th>City 6</th>
<th>City 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>1, 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2, 5, 1</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5, 4</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1, 3</td>
<td></td>
</tr>
</tbody>
</table>

- This value cannot be used by pairs not involving team 8.
- One of the pairs involving team 8 must use this value.
- These values cannot be used by any pair involving team 8.
Redundant Constraints

- Red value cannot be used by pairs not involving team 8
  - disequalities
- One of the pairs involving team 8 must use red value
  - occurrences (gcc) constraint
- Yellow values cannot be used by any pair involving team 8
  - disequalities

<table>
<thead>
<tr>
<th></th>
<th>City 1</th>
<th>City 2</th>
<th>City 3</th>
<th>City 4</th>
<th>City 5</th>
<th>City 6</th>
<th>City 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day 1</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td>7, 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 2</td>
<td>2</td>
<td>1, 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 3</td>
<td>7</td>
<td></td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 4</td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>5</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Day 5</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Day 6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5, 4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day 7</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1, 3</td>
<td></td>
</tr>
</tbody>
</table>
Before Search
Impact of improved hint handling

With index set channeling

Improved Hints
Conclusions

- Many ways of modelling even simple problems
- Selection of “best” model difficult
  - Depends on constraints available
  - Often needs experimentation
- How do we measure if one model is “better” than another?
  - Execution time?
  - Size of search tree?
  - Scalability?
- Definition of variables is key
- Explore choices by considering mapping operators
Why CP-Viz?
Why Visualization?

- Constraint Programming = Declarative Programming?
- If they work, they are easy to understand
- What to do if they don’t work?
- Visualization shows what is happening
  - At the right level of abstraction
  - In terms a user can understand
Previous Work

- Sepia
- GRACE
- Oz-Explorer
- DISCiPI (COSYTEC, PrologIA, Madrid, INRIA,...)
- OADymPPaC (COSYTEC, ILOG, INRIA, EMN, IRISA, LIFO)
Problems

- Tools either very system specific (nearly all existing systems)
- or too generic (OADymPPaC)
- Required: Light weight, generic visualization tool
Features

- Detailed or compact tree view
- Views of state after each choice
- Specialized visualizers for different global constraints
- State, path, (tree) and evolution views
- Invariant checking
- No view of internals of propagation
- No view of constraint network
Background

- Initially developed for ECLiPSe
- Funded by Cisco through gift grant
- ECLiPSe ELEarning Course
- Intended reuse for multiple systems
How do we understand behavior?

- Mental model
- Formal analysis
- Debugging
- Tracing
- Life visualization
- Post-mortem analysis
How do we understand behavior?

- Mental model
- Formal analysis
- Debugging
- Tracing
- Life visualization
- Post-mortem analysis
Why Visualize?

- Understand what is done
- Understand what is done in which order
- Understand what is not done
- Understand when to give up
Conceptual Model

- Stable state at defined program points
- Granularity
  - Assign value
  - Post constraint
- Show stable state after propagation
- Do not show individual propagation steps
Architecture

Program + Annotation

ECLiPSe

TreeLog

VisualizationLog

Viz

Treemap

SVG

Graph

Statistics

Inkscape

Browser

Batch

VizTool

Annotated Image

PDF
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!- Helmut Simonis (University College Cork) ->
<tree version="1.0"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:noNamespaceSchemaLocation="tree.xsd">
  <root id="0"/>
  <try id="1" parent="0" name="S" size="1" value="9"/>
  <fail id="2" parent="1" name="E" size="4" value="4"/>
  <try id="3" parent="1" name="E" size="4" value="5"/>
  <try id="4" parent="3" name="N" size="1" value="6"/>
  <try id="5" parent="4" name="D" size="1" value="7"/>
  <try id="6" parent="5" name="M" size="1" value="1"/>
  <try id="7" parent="6" name="O" size="1" value="0"/>
  <try id="8" parent="7" name="R" size="1" value="8"/>
  <try id="9" parent="8" name="Y" size="1" value="2"/>
  <succ id="9"/>
  <fail id="10" parent="1" name="E" size="4" value="6"/>
  <fail id="11" parent="1" name="E" size="4" value="7"/>
</tree>
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!- Helmut Simonis (University College Cork) ->
<tree version="1.0"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:noNamespaceSchemaLocation="tree.xsd">
    <root id="0"/>
    <try id="1" parent="0" name="S" size="1" value="9"/>
    <fail id="2" parent="1" name="E" size="4" value="4"/>
    <try id="3" parent="1" name="E" size="4" value="5"/>
    <try id="4" parent="3" name="N" size="1" value="6"/>
    <try id="5" parent="4" name="D" size="1" value="7"/>
    <try id="6" parent="5" name="M" size="1" value="1"/>
    <try id="7" parent="6" name="O" size="1" value="0"/>
    <try id="8" parent="7" name="R" size="1" value="8"/>
    <try id="9" parent="8" name="Y" size="1" value="2"/>
    <succ id="9"/>
    <fail id="10" parent="1" name="E" size="4" value="6"/>
    <fail id="11" parent="1" name="E" size="4" value="7"/>
</tree>
Search Tree Schema

```
<tree>
  <attributes>
    <version/>
  </attributes>
  <try/>
    <attributes>
      <id/>
      <parent/>
      <name/>
      <size/>
      <value/>
    </attributes>
  </try>
  <eucc/>
    <attributes>
      <id/>
    </attributes>
  </eucc>
  <root/>
    <attributes>
      <id/>
    </attributes>
  </root>
  <fail/>
    <attributes>
      <id/>
      <parent/>
      <name/>
      <size/>
      <value/>
    </attributes>
  </fail>
</tree>
```
Example Search Tree Output
Outline

8 Introduction

9 Search Tree Visualizer

10 Constraint and Variable Visualizers
   - Example
     - Basic Types
     - Structured Types
   - Schema
   - Output

11 Tools
Example Log for Cumulative Constraint

```xml
</visualizer_state>
<visualizer_state id="6" >
<argument index="tasks" >
<tuple index="1" >
<dvar index="start" domain="1 .. 8" />
<integer index="dur" value="1" />
<integer index="res" value="1" />
</tuple>
<tuple index="2" >
<dvar index="start" domain="1 .. 8" />
<integer index="dur" value="1" />
<integer index="res" value="1" />
</tuple>
<tuple index="3" >
<dvar index="start" domain="1 .. 8" />
<integer index="dur" value="1" />
<integer index="res" value="1" />
</tuple>
<tuple index="4" >
<dvar index="start" domain="1 .. 8" />
<integer index="dur" value="1" />
<integer index="res" value="1" />
</tuple>
</argument>
<argument index="limit" >
<integer index="1" value="1" />
</argument>
</visualizer_state>
```
Outline

8 Introduction

9 Search Tree Visualizer

10 Constraint and Variable Visualizers

11 Tools
<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XMLSpy v2010 (http://www.altova.com)-->
<configuration version="1.0" directory="examples/mix/RESULT"
    xsi:noNamespaceSchemaLocation="configuration.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <tool show="tree" type="layout" display="expanded" repeat="all"
        width="700" height="700" fileroot="tree" />
    <tool show="viz" type="layout" display="compact" repeat="final"
        width="900" height="900" fileroot="viz" />
</configuration>
VizTool: Car Sequencing
How to Interpret Visualization

- **Search tree**
  - Good/bad choices
  - Place of backtracking

- **State**
  - Missing propagation
Background

- Provided as challenge by Roberto Nieuwenhuis, Barcelona
- Production problem in steel industry
- Modelled with SMT (Barcelologic)
- They could not find optimal solution with CP
Problem

- Jobs to be processed (different types, multiple instances per type)
- Nearly flow shop: Loading, Oven, Cooling
- 2 Ovens, 1 Cooling unit
- 3 activities at same time (all types)
- No waiting between loading, oven and cooling
- Minimize makespan
Model

- One variable per job, Start time in oven
- Three cumulative constraints: Ovens, Cooling, All
- Overall End is max of all job ends
- Try and find solution with optimal makespan (imposed)
Search

- Standard variable assignment hopeless
- 20 Jobs, 150 time points
- Custom routine:
  - Select job (non-determ)
  - Fix start as early as possible (first feasible value)
Initial State
Observation

- Initial estimated *End* pathetic
- No symmetry breaking
- Ovens very near capacity, except at very end
- Cooling resource has spare capacity, but imposes constraints
Problems with *cumulative* Implementation
Problems with \textit{cumulative} Implementation
Need for Invariant Checking

- ECLiPSe cumulative does not do a good job
- Written in C, unchanged since 199x
- Slower than state-of-the-art, but why?
- Only implements edge finding, no obligatory part reasoning
What we know about cumulative

\[ \forall t : \sum_{i \text{ s.t. } s_i \leq t < s_i + d_i} r_i \leq L \]

\[ E = \max (s_k + d_k) \]

\[ \sum d_i \times r_i \leq L \times E \]
Invariant 1

\[ \sum d_i \times r_i \leq L \times E \]

\[ E \geq \frac{\sum d_i \times r_i}{L^+} \]

\[ E^+ < \frac{\sum d_i \times r_i}{L^+} \Rightarrow \text{inconsistent} \]

\[ E^- < \frac{\sum d_i \times r_i}{L^+} \Rightarrow \text{missing prop} \]
Invariant 2

\[ \forall t : \sum_{i \text{ s.t. } s_i \leq t < s_i + d_i} r_i \leq L \]

\[ L \geq \sum_{i \text{ s.t. } s_i^+ \leq t < s_i^- + d_i} r_i \]

\[ L^+ < \sum_{i \text{ s.t. } s_i^+ \leq t < s_i^- + d_i} r_i \Rightarrow \text{inconsistent} \]

\[ L^- < \sum_{i \text{ s.t. } s_i^+ \leq t < s_i^- + d_i} r_i \Rightarrow \text{missing prop} \]
Invariant Checking

- At each state, check invariants for each constraint
- Mark violations in log, also in tree
- Different levels
  - Ground instance violates invariant: bug
  - State inconsistent: node could be pruned
  - Missing propagation: domain could be reduced
Tree with Invariant Warnings
Improving the model

- Better lower bound for *End*
- Symmetry breaking by ordering jobs of same type
- Dummy oven task at end to make profile near-perfect
- Discover and fix bug in search routine
- (Add obligatory part reasoning to cumulative)
- (Change search to order cooling tasks)
Improved Model - Initial State
Improved Model - Solution Found (0.02 seconds)
Future Work

- Interfaces to other systems
- Direct Java interface without XML intermediate
- Better visualizations for some global constraints
- Derive invariants automatically
- Use Java-based constraint system to check invariants
### Introduction
Search Tree Visualizer
Constraint and Variable Visualizers

### Tools

---

**Latest Issue**

**Regulars**

**Features**

**Buy Stuff etc**

---

**Visualization Tool**

**The Magazine That's Better Than Nothing**

---

---

Helmut Simonis
Modelling