This work is licensed under the Creative Commons Attribution-Noncommercial-Share Alike 3.0 Unported License. To view a copy of this license, visit http://creativecommons.org/licenses/by-nc-sa/3.0/ or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.
Outline

1. Overview of Course
2. What do you need to know about CP?
3. What is CP good for?
4. A Core Set of Global Constraints
5. Visualization
Main Message

- New ELearning course for ECLiPSe
- Modelling and programming with constraints
- Based on sample problems solved and explained in detail
- A view on core constraint programming skills
- Strong dependence on visualization to explain behavior
Interesting Things to Come

- Which global constraints a system should contain
- How to interpret search trees
- The open challenge of debugging global constraints
Outline

1. Overview of Course
2. What do you need to know about CP?
3. What is CP good for?
4. A Core Set of Global Constraints
5. Visualization
Gift grant from Cisco Systems/Silicon Valley Community Foundation
Cisco owns open-sourced ECLiPSe system
How to expand user-base?
Self-taught course in constraint programming
Intended for Cisco engineers/programmers
Open source/available to community
Format

- Video lectures
- Slides
- Handout
- Exercises
Examples

- Website
  http://4c.ucc.ie/~hsimonis/ELearning/index.htm
- Slides
  ../introduction/slides.pdf
- Handout
  ../introduction/handout.pdf
- Video
  ../wavedemo/DEMO/web/web.html
Outline

1. Overview of Course
2. What do you need to know about CP?
3. What is CP good for?
4. A Core Set of Global Constraints
5. Visualization
Central Topics

- Basic structure of constraint programs
- Global constraints
- User-defined search
- Optimization
- Symmetry breaking
- Choosing the right model
- Limits of propagation
Explaining Concepts

- Explain as you go
- Constraints introduced when used by application
- Concepts/algorithms explained by example
Example: Domain consistent \texttt{alldifferent}

\begin{verbatim}
:-lib(ic).
:-lib(ic_global_gac).

top:-
    [X,Y] :: 1..2,
    Z :: 2..5,
    [T,U] :: 3..5,
    V :: [2,4,6,7],
    ic_global_gac:alldifferent([X,Y,Z,T,U,V]).
\end{verbatim}
Making constraint domain consistent

Problem shown as bipartite graph

X 1
Y 2
Z 3
T 4
U 5
V 6
7
Making constraint domain consistent

Find maximal matching (in blue)
Making constraint domain consistent

Orient graph (edges in matching from variables to values, all others from values to variables), mark edges in matching
Overview of Course
What do you need to know about CP?
What is CP good for?
A Core Set of Global Constraints
Visualization

Making constraint domain consistent

Find strongly connected components (green and brown), mark their edges

\[ X \rightarrow 1 \]
\[ Y \rightarrow 2 \]
\[ Z \leftarrow 3 \]
\[ T \leftarrow 4 \]
\[ U \leftarrow 5 \]
\[ V \rightarrow 6 \]
\[ 7 \]
Making constraint domain consistent

Find unmatched value nodes (here node 7, magenta)
Making constraint domain consistent

Find alternating paths from such nodes (in magenta), mark their edges

X → 1
Y → 2
Z ← 3
T ← 4
U ← 5
V ← 6

7

Helmut Simonis
Lessons Learned
Making constraint domain consistent

All unmarked edges can be removed
Overview of Course
What do you need to know about CP?
What is CP good for?
A Core Set of Global Constraints
Visualization

Making constraint domain consistent

Resulting graph, constraint is domain consistent

X 1
Y 2
Z 3
T 4
U 5
V 6
7
Extended Example

:-lib(ic).
:-lib(ic_global_gac).

top:-
    X :: 1..2,
    Y :: [1,2,7],
    Z :: 2..5,
    [T,U] :: 3..5,
    V :: [2,4,6,7],
ic_global_gac:alldifferent([X,Y,Z,T,U,V]).
No propagation in expanded example

Problem shown as bipartite graph

X ———— 1
Y ———— 2
Z ———— 3
T ———— 4
U ———— 5
V ———— 6
7
Overview of Course
What do you need to know about CP?
What is CP good for?
A Core Set of Global Constraints
Visualization

No propagation in expanded example

Find maximal matching (in blue)
No propagation in expanded example

Orient graph (edges in matching from variables to values, all others from values to variables), mark edges in matching
Overview of Course
What do you need to know about CP?
What is CP good for?
A Core Set of Global Constraints
Visualization

No propagation in expanded example

Find strongly connected components (green and brown), mark their edges
No propagation in expanded example

Find unmatched value nodes (here node 7, magenta)
No propagation in expanded example

Find alternating paths from such nodes (in magenta), mark their edges.
Overview of Course
What do you need to know about CP?
What is CP good for?
A Core Set of Global Constraints
Visualization

No propagation in expanded example

Continue with alternating paths
No propagation in expanded example

Continue with alternating paths, all edges marked, no propagation, constraint is domain consistent
Overview of Course
What do you need to know about CP?
What is CP good for?
A Core Set of Global Constraints
Visualization

No Theoretical Background

- Target audience
  - Engineers
  - Application programmers
- Not a course in theoretical computer science
- Different from current textbooks on CP
Overview of Course
What do you need to know about CP?
What is CP good for?
A Core Set of Global Constraints
Visualization
Overview of Course
What do you need to know about CP?
What is CP good for?
A Core Set of Global Constraints
Visualization

This was easy ten years ago:

- Based on industrial experience
  - Assignment problems
  - Scheduling
  - Transportation
  - Personal planning
In the meantime

- Dramatic improvements for competitors
  - MILP
  - SAT
- Improvements in hardware performance
New Areas

- Configuration
  - In search of a nice example
- Placement
  - Not in ECLiPSe
- Networks
  - In form of MIP/FD hybrids
Problems Handled in Course

- Must have puzzles!
- Send+More=Money
- Sudoku
- N-queens
- Shikaku (see Wednesday’s talk)
Practical Example Problems

- Test plan generation (BIBD)
- Progressive party problem
- Routing and wavelength assignment
- Optical network design
- Car sequencing
- Costas arrays
- Sports scheduling
- Still to come
  - Production scheduling
  - Nurse rostering
  - Airport stand allocation
Intention

- Realistic, life like problems
- Must address scalability issues
- Often, problem not completely specified
- Issue: Hard to verify by hand
- Complexity still limited, not real problems
- No attempt at integration
What is CP good for: A Problematic View

- CP works if it out-performs everything else
- See which technique produces best (optimal) solutions
- Which technique runs fastest
- A publication game (I like to play too!)
  - Somebody defines the problem
  - Minute detail, no choices left
  - Counting runtime, choices, backtracks
  - Tiny improvements count
From RWA Chapter

<table>
<thead>
<tr>
<th>Network</th>
<th>Dem.</th>
<th>Complete</th>
<th>MIP-MIP</th>
<th>Decomposition</th>
<th>MIP-SAT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>MIP Opt</td>
<td>Avg</td>
<td>MIP-MIP Opt</td>
<td>Avg</td>
</tr>
<tr>
<td>brezil</td>
<td>100</td>
<td></td>
<td>277.14</td>
<td>100</td>
<td>0.91</td>
</tr>
<tr>
<td>brezil</td>
<td>200</td>
<td></td>
<td></td>
<td>100</td>
<td>4.45</td>
</tr>
<tr>
<td>brezil</td>
<td>300</td>
<td></td>
<td></td>
<td>100</td>
<td>8.08</td>
</tr>
<tr>
<td>brezil</td>
<td>400</td>
<td></td>
<td></td>
<td>100</td>
<td>10.93</td>
</tr>
<tr>
<td>brezil</td>
<td>500</td>
<td></td>
<td></td>
<td>100</td>
<td>13.09</td>
</tr>
<tr>
<td>brezil</td>
<td>600</td>
<td></td>
<td></td>
<td>100</td>
<td>16.77</td>
</tr>
<tr>
<td>eon</td>
<td>100</td>
<td></td>
<td>33.62</td>
<td>100</td>
<td>1.51</td>
</tr>
<tr>
<td>eon</td>
<td>200</td>
<td></td>
<td>65.51</td>
<td>100</td>
<td>5.27</td>
</tr>
<tr>
<td>eon</td>
<td>300</td>
<td></td>
<td>121.27</td>
<td>100</td>
<td>5.60</td>
</tr>
<tr>
<td>eon</td>
<td>400</td>
<td></td>
<td>116.64</td>
<td>100</td>
<td>7.38</td>
</tr>
<tr>
<td>eon</td>
<td>500</td>
<td></td>
<td>162.55</td>
<td>100</td>
<td>9.58</td>
</tr>
<tr>
<td>eon</td>
<td>600</td>
<td></td>
<td>232.91</td>
<td>99</td>
<td>14.04</td>
</tr>
<tr>
<td>mci</td>
<td>100</td>
<td></td>
<td>20.27</td>
<td>100</td>
<td>2.08</td>
</tr>
<tr>
<td>mci</td>
<td>200</td>
<td></td>
<td>38.79</td>
<td>100</td>
<td>5.36</td>
</tr>
<tr>
<td>mci</td>
<td>300</td>
<td></td>
<td>55.78</td>
<td>100</td>
<td>5.83</td>
</tr>
<tr>
<td>mci</td>
<td>400</td>
<td></td>
<td>109.85</td>
<td>100</td>
<td>8.71</td>
</tr>
<tr>
<td>mci</td>
<td>500</td>
<td></td>
<td>129.90</td>
<td>100</td>
<td>13.89</td>
</tr>
<tr>
<td>mci</td>
<td>600</td>
<td></td>
<td>257.70</td>
<td>100</td>
<td>22.56</td>
</tr>
<tr>
<td>nsf</td>
<td>100</td>
<td></td>
<td>8.17</td>
<td>100</td>
<td>2.38</td>
</tr>
<tr>
<td>nsf</td>
<td>200</td>
<td></td>
<td>12.75</td>
<td>100</td>
<td>1.81</td>
</tr>
<tr>
<td>nsf</td>
<td>300</td>
<td></td>
<td>17.01</td>
<td>100</td>
<td>1.98</td>
</tr>
<tr>
<td>nsf</td>
<td>400</td>
<td></td>
<td>27.36</td>
<td>100</td>
<td>3.54</td>
</tr>
<tr>
<td>nsf</td>
<td>500</td>
<td></td>
<td>54.60</td>
<td>100</td>
<td>5.77</td>
</tr>
<tr>
<td>nsf</td>
<td>600</td>
<td></td>
<td>88.72</td>
<td>100</td>
<td>9.09</td>
</tr>
</tbody>
</table>
Is MIP-FD decomposition really better than MIP-SAT?
- Both are good enough
- Both are clearly better than complete MIP
- This is only one possible problem variant
  - For other variant difference is much more significant
A More Meaningful Evaluation

- When is a solution good enough?
- How long does it take to get there?
  - Total development time, not runtime
  - For whom: beginner, expert, genius?
- What happens if you change the problem?
- Can you explain what is happening?
- How easy it is to integrate into workflow?
Problem: How do we measure/report this?

- Usability labs?
- Instrument IDE?
- Cost of parallel development
- Commercial sensitivities
- Can this be published?
Outline

1. Overview of Course
2. What do you need to know about CP?
3. What is CP good for?
4. A Core Set of Global Constraints
5. Visualization
Global Constraint Catalog (A-C)

- all differ from at least k pos
- all equal
- all min dist
- alldifferent
- alldifferent between sets
- alldifferent consecutive values
- alldifferent cst
- alldifferent except 0
- alldifferent interval
- alldifferent modulo
- alldifferent on intersection
- alldifferent partition
- alldifferent same value
- allperm
- among
- among diff 0
- among interval
- among low up
- among modulo
- among seq
- among var
- and
- arith
- arith or
- arith sliding
- assign and counts
- assign and nvalues
- atleast
- atleast nvalue
- atleast nvector
- atmost
- atmost 1
- atmost nvalue
- atmost nvector
- balance
- balance interval
- balance modulo
- balance partition
- between min max
- bin packing
- bin packing capa
- binary tree
- bipartite
- calendar
- cardinality at least
- cardinality at most
- cardinality at most partition
- change
- change continuity
- change pair
- circuit
- circuit cluster
- circular change
- clause and
- clause or
- clique
- colored matrix
- coloured cumulative
- coloured cumulates
- common
- common interval
- common modulo
- common partition
- cond lex cost
- cond lex greater
- cond lex greater or
- cond lex less
- cond lex lesseq
Core Global Constraints

- element
- alldifferent
- gcc
- lex-ordering
- bin packing
- sequence
- cumulative
- regular
- diffn/disjoint/geost

Not sure about:

- min weight
- alldifferent
- gcc with cost
- cycle
- nvalue
Sources

- Good survey papers and PhD thesis
  - W. van Hoeve: alldifferent
  - C. Quimper: alldifferent, gcc
  - Z. Kiziltan: lex ordering

- Clear descriptions of specific constraints
  - P. Shaw: bin packing
  - G. Pesant: regular
Families of Constraints

- Matching/flow based global constraints
  - Build graph
  - Find matching/flow
  - Reorient links
  - Find SCC
  - (BFS reachable)
  - Remove unmarked edges
Examples

- alldifferent
- gcc
- alldifferent matrix
- gcc matrix
- same
- sequence
Some Constraints Still Opaque

- cumulative
- cycle
- geost
Which Variant to Implement

- **gcc**
  - Low/high limits or variables for counters
  - Open/closed version

- **cumulative**
  - Capacity limit or variable
  - End limit
  - Task surface
Use in Course

<table>
<thead>
<tr>
<th>Constraint</th>
<th>Sudoku</th>
<th>N-queens</th>
<th>RWA</th>
<th>BIBD</th>
<th>Party</th>
<th>Sports</th>
<th>Car</th>
<th>Skikaku</th>
<th>Network Design</th>
<th>Costas</th>
<th>Scheduling</th>
<th>Airport</th>
<th>Nurse</th>
</tr>
</thead>
<tbody>
<tr>
<td>element</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>alldifferent</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>gcc</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lex</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
<td>x</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sequence</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>bin packing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cumulative</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>somedifferent</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Open Challenge

- How do we implement/test global constraints?
- No publication, ever
- No methodology
Techniques Used

- Profiling
- Line coverage
- Generic GAC
Overview of Course
What do you need to know about CP?
What is CP good for?
A Core Set of Global Constraints
Visualization

Profiling

Helmut Simonis
Lessons Learned
Generic **GAC**: `alldifferent_sum`

```
alldifferent_sum(L,N):-
  (ic:alldifferent(L),
   sum(L) #= N,
   labeling(L)) infers ac.
```

Propia library (Le Provost, Wallace)
Overview of Course
What do you need to know about CP?
What is CP good for?
A Core Set of Global Constraints
Visualization

Worst Case Scenario: Progressive Party Problem

- Progressive Party Problem
- Adding stronger global constraint, bin packing (P. Shaw)
- Found different solution
  - Variable selection changed by improved propagation?
  - Consistent value removed by bug in constraint?
Outline

1. Overview of Course
2. What do you need to know about CP?
3. What is CP good for?
4. A Core Set of Global Constraints
5. Visualization
How do we understand behavior?

- Mental model
- Formal analysis
- Debugging
- Tracing
- Life visualization
- Post-mortem analysis
Design Choices

- No deep integration with solver
- Post-mortem visualization
- Intermediate file format
- No view of detailed propagation
  - Tool not intended for debugging constraint engine
Conceptual Model

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

- Search tree
- Variables
- Constraints
Overview of Course
What do you need to know about CP?
What is CP good for?
A Core Set of Global Constraints
Visualization

Visualization Tool

- Developed in Java
- Show two panes: tree and state
- Navigate along timeline
Overview of Course
What do you need to know about CP?
What is CP good for?
A Core Set of Global Constraints
Visualization

Visualization Tool: Car Sequencing
How many visualizers do we need?

- Develop few primitives
  - Cell based view
  - Domain vector
- Allow aggregation
  - Vector/matrix
  - General layout
- Which global constraints require more?
  - Task based view for cumulative
  - Matching/flow based representation does not scale
How to Interpret Visualization

- Search tree
  - Good/bad choices
  - Place of backtracking
- State
  - Missing propagation
Overview of Course
What do you need to know about CP?
What is CP good for?
A Core Set of Global Constraints
Visualization

Costas Array Search tree (Size 16)

- Deep backtracking
- Third choice wrong
- Last choice wrong
- Value selection strategy useless

Helmut Simonis
Lessons Learned
Overview of Course
What do you need to know about CP?
What is CP good for?
A Core Set of Global Constraints
Visualization

Costas Array Search tree (Size 16)

- Deep backtracking
- Third choice wrong
- Last choice wrong
- Value selection strategy useless
Overview of Course
What do you need to know about CP?
What is CP good for?
A Core Set of Global Constraints
Visualization

Costas Array Search tree (Size 16)

- Deep backtracking
- Third choice wrong
- Last choice wrong
- Value selection strategy useless

Helmut Simonis
Lessons Learned
Overview of Course
What do you need to know about CP?
What is CP good for?
A Core Set of Global Constraints
Visualization

Costas Array Search tree (Size 16)

- Deep backtracking
- Third choice wrong
- Last choice wrong
- Value selection strategy useless
Overview of Course
What do you need to know about CP?
What is CP good for?
A Core Set of Global Constraints
Visualization

Missing Propagation

The model is doing this
Missing Propagation

It could be doing that!
Overview of Course
What do you need to know about CP?
What is CP good for?
A Core Set of Global Constraints
Visualization

Comparison (Search Tree, size 16)

Initial Model

Improved Model

Helmut Simonis
Lessons Learned
Overview of Course
What do you need to know about CP?
What is CP good for?
A Core Set of Global Constraints
Visualization

Progressive Party Problem, 9 Time Periods
2 Restarts Before Solution Found
Value Choice Strategy Not Focused
Progressive Party

- Clearly impossible to explore search space
- Either many solutions or good value selection
- Value selection at end rather poor
- Probably many solutions
From Shikaku Presentation (Wednesday)
What is still missing?

- Explanation of failures
  - Procedural: Show propagation which leads to failure
  - Declarative: Find conflict set
- Comparison of trees
Outline

6 Some Technical Details

7 Conclusions
How was this generated?

- Slides produced with LaTeX `beamer` class
- Templates for chapters
- Visualization imported as pdf files
- Using `inkscape` as SVG to PDF converter
- Allows to produce multiple versions from one source
  - slides
  - handout
  - article
Videos

- Recorded and produced with Camtasia Studio
- Long takes, minimal editing
- Screen capture at full resolution
- Produce video in different target formats
  - Web based (640x480)
  - iPhone (480x320)
  - HD video possible
Some Technical Details

Conclusions

Metrics *without* Program Development

- Slides: 2-4 days per chapter
- Video: 1:10 ratio for finished product
  - 2-3 trials
  - No fine grain editing
  - Only for bulk production
Outline

6 Some Technical Details

7 Conclusions
Conclusions

- New ELearning course for ECLiPSe
- Open source material, Creative Commons BY-NC-SA license
  - Application driven
  - Modelling with global constraints
  - Customizing search
- Effort only justifiable through Cisco grant
Conclusions

- If you are not using visualization, why not?
- Generic tools, not just for ECLiPSe