Constraint Technology
A Programming Paradigm on the Rise

Pierre Flener

ASTRA Research Group on Constraint Technology
Department of Information Technology
Uppsala University
Sweden

http://www.it.uu.se/research/group/astra/

Invited talks at UU / IT / TDB (2008-03-12)
and at UU / IT / TAPVES (2008-03-13)
Outline

1 What is Constraint Technology?
   - Constraint Problems
   - Constraint Technology
   - Constraint Modelling
   - Constraint Solving by Global Search
   - Constraint Solving by Local Search
   - History & Success Stories

2 Enabling Basic Research and Tools
   - Abstract Constraint Modelling
   - Symmetry
   - Constraint-Based Local Search

3 Technology Transfer
   - Air-Traffic Management
   - Computational Biology
   - Computational Finance
Outline

1. What is Constraint Technology?
   - Constraint Problems
   - Constraint Technology
   - Constraint Modelling
   - Constraint Solving by Global Search
   - Constraint Solving by Local Search
   - History & Success Stories

2. Enabling Basic Research and Tools
   - Abstract Constraint Modelling
   - Symmetry
   - Constraint-Based Local Search

3. Technology Transfer
   - Air-Traffic Management
   - Computational Biology
   - Computational Finance
Outline

1 What is Constraint Technology?
   - Constraint Problems
     - Constraint Technology
     - Constraint Modelling
     - Constraint Solving by Global Search
     - Constraint Solving by Local Search
     - History & Success Stories

2 Enabling Basic Research and Tools
   - Abstract Constraint Modelling
   - Symmetry
   - Constraint-Based Local Search

3 Technology Transfer
   - Air-Traffic Management
   - Computational Biology
   - Computational Finance
### Constraint Problems

#### Example (The Tourist Site Competition (TSC) problem)

<table>
<thead>
<tr>
<th></th>
<th>Ali</th>
<th>Dan</th>
<th>Eva</th>
<th>Jim</th>
<th>Leo</th>
<th>Mia</th>
<th>Ulla</th>
</tr>
</thead>
<tbody>
<tr>
<td>Birka</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Falun</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lund</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mora</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sigtuna</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Uppsala</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ystad</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1. Every tourist site is visited by \( r = 3 \) judges.
2. Every judge visits \( c = 3 \) tourist sites.
3. Every pair of sites is visited by \( \lambda = 1 \) common judge.
### Constraint Problems

**Example (The Tourist Site Competition (TSC) problem)**

<table>
<thead>
<tr>
<th></th>
<th>Ali</th>
<th>Dan</th>
<th>Eva</th>
<th>Jim</th>
<th>Leo</th>
<th>Mia</th>
<th>Ulla</th>
</tr>
</thead>
<tbody>
<tr>
<td>Birka</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Falun</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Lund</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Mora</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sigtuna</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Uppsala</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Ystad</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

1. Every tourist site is visited by \( r = 3 \) judges.
2. Every judge visits \( c = 3 \) tourist sites.
3. Every pair of sites is visited by \( \lambda = 1 \) common judge.
Constraint Problems

Many increasingly ubiquitous and important real-life problems *must* be solved by intelligent search:

- Rostering, scheduling, time-tabling
- Planning
- Configuration, design
- RNA structure prediction, alignment, sequencing, . . .
- Financial investment instrument design
- VLSI circuit layout
- Hardware / software specification verification
- . . .

In a **constraint problem**, decisions have to be made so that:

- Some constraints are **satisfied**.
- Optionally: Some cost/benefit is **minimised/maximised**.
Outline

1 What is Constraint Technology?
   - Constraint Problems
   - **Constraint Technology**
   - Constraint Modelling
   - Constraint Solving by Global Search
   - Constraint Solving by Local Search
   - History & Success Stories

2 Enabling Basic Research and Tools
   - Abstract Constraint Modelling
   - Symmetry
   - Constraint-Based Local Search

3 Technology Transfer
   - Air-Traffic Management
   - Computational Biology
   - Computational Finance
Constraint Technology

Constraint Technology (CT) offers methods and tools for:

- Effectively **modelling** constraint problems.
- Efficiently **solving** constraint problems,
  by **global search** (in Sudoku fashion)
  or by **local search** (see below).

Slogan of CT:

\[
\text{Constraint Program} = \text{Model} + \text{Search}
\]

This is orthogonal and complementary to

- Operations Research (OR):
  linear programming (LP), integer LP (ILP),
  mixed integer linear programming (MILP), . . .
- Boolean satisfiability (SAT)
- . . .

leading to hybridised optimisation technologies!
Scope of Constraint Technology

Constraint Technology addresses:

- satisfaction problems and optimisation problems
- discrete variables and continuous variables
- linear constraints and non-linear constraints

in any combination thereof, by:

- global search, if optimality more important than speed
- local search, if speed is more important than optimality

Built-in support for:

- explanations
- soft constraints
- ...
Outline

1 What is Constraint Technology?
   - Constraint Problems
   - Constraint Technology
   - Constraint Modelling
     - Constraint Solving by Global Search
     - Constraint Solving by Local Search
     - History & Success Stories

2 Enabling Basic Research and Tools
   - Abstract Constraint Modelling
   - Symmetry
   - Constraint-Based Local Search

3 Technology Transfer
   - Air-Traffic Management
   - Computational Biology
   - Computational Finance
Example (TSC integer model)

```plaintext
set Sites, Judges
cst r, c, λ : int
var TSC : array[Sites, Judges] of 0..1
solve
  forall s in Sites : r = sum(j in Judges) TSC[s, j]
  forall j in Judges : c = sum(s in Sites) TSC[s, j]
  forall s₁ ≠ s₂ in Sites :
    λ = count(j in Judges) TSC[s₁, j] = 1 = TSC[s₂, j]
```

Example (Instance data for the Sweden TSC instance)

Sites = \{Birka, Falun, Lund, Mora, Sigtuna, Uppsala, Ystad\}
Judges = \{Ali, Dan, Eva, Jim, Leo, Mia, Ulla\}
⟨r, c, λ⟩ = ⟨3, 3, 1⟩
Constraint Modelling

Example (TSC set model)

```prolog
set Sites, Judges
cst r, c, λ : int
var TSC : array[Sites] of setof(Judges)
solve
   forall s in Sites : r = size(TSC[s])
   forall j in Judges : c = count(s in Sites) j ∈ TSC[s]
   forall s1 ≠ s2 in Sites :
      λ = size(TSC[s1] ∩ TSC[s2])
```

Example (Instance data for the Sweden TSC instance)

Sites = \{Birka, Falun, Lund, Mora, Sigtuna, Uppsala, Ystad\}
Judges = \{Ali, Dan, Eva, Jim, Leo, Mia, Ulla\}
⟨r, c, λ⟩ = ⟨3, 3, 1⟩
Example (Sudoku)

<table>
<thead>
<tr>
<th></th>
<th>6</th>
<th>1</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>3</td>
<td>5</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>7</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>9</td>
<td>1</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>8</td>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>

Fill in the grid so that every row, every column, and every highlighted 3 × 3 box contains the digits 1 through 9.

range \( N = 1..9 \)

\[
\text{var } s : \text{array}[N, N] \text{ of } N
\]

\[
\text{solve}
\]

\[
\ldots \quad \text{// load clues}
\]

\[
\text{forall } r \text{ in } N : \text{AllDifferent}(s[r, *])
\]

\[
\text{forall } c \text{ in } N : \text{AllDifferent}(s[* , c])
\]

\[
\text{forall } r, c \text{ in } 1..9 \text{ by } 3 : \text{AllDifferent}(s[r + 0..2, c + 0..2])
\]
### Example (Sudoku)

Fill in the grid so that every row, every column, and every highlighted 3 × 3 box contains the digits 1 through 9.

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>1</td>
<td>4</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>5</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>7</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>9</td>
<td>1</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>8</td>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>

**Range** $N = 1..9$

**Variables**

\[
\text{var } s : \text{array}[N, N] \text{ of } N
\]

**Solve**

\[
\begin{align*}
\text{\ldots } & \text{ // load clues } \\
\text{forall } r \text{ in } N : \text{AllDifferent}(s[r, \star]) \\
\text{forall } c \text{ in } N : \text{AllDifferent}(s[\star, c]) \\
\text{forall } r, c \text{ in } 1..9 \text{ by } 3 : \\
& \text{AllDifferent}(s[r + 0..2, c + 0..2])
\end{align*}
\]
Consider the $n$-ary constraint $\text{AllDifferent}$, with $n = 4$ here:

$$\text{AllDifferent}([a, b, c, d]) \quad (1)$$

**Declaratively**, (1) is equivalent to the $\Theta(n^2)$ 2-ary constraints

$$a \neq b, \ a \neq c, \ a \neq d, \ b \neq c, \ b \neq d, \ c \neq d \quad (2)$$

but it provides convenient genericity in constraint models. **Operationally**, (1) prunes much stronger than (2). Example:

$$a \in \{2, 3\}, \ b \in \{2, 3\}, \ c \in \{1, 3\}, \ d \in \{1, 2, 3, 4\}$$

No pruning by (2). But perfect pruning by (1).
Example (\textit{AllDifferent})

Consider the $n$-ary constraint \textit{AllDifferent}, with $n = 4$ here:

\begin{equation}
\text{AllDifferent}([a, b, c, d])
\end{equation}

\textbf{Declaratively}, (1) is equivalent to the $\Theta(n^2)$ 2-ary constraints

\begin{equation}
a \neq b, \ a \neq c, \ a \neq d, \ b \neq c, \ b \neq d, \ c \neq d
\end{equation}

but it provides convenient genericity in constraint models. \textbf{Operationally}, (1) prunes much stronger than (2). Example:

\begin{equation}
a \in \{2, 3\}, \ b \in \{2, 3\}, \ c \in \{1, \ 3\}, \ d \in \{1, \ 2, \ 3, \ 4\}
\end{equation}

\textit{No} pruning by (2). \textit{But perfect pruning by (1).}
Global constraints are a much admired feature of CT: they allow the **preservation of combinatorial sub-structures** of a problem, both while modelling it and while solving it.

About 300 global constraints have been implemented so far, declaratively encapsulating advanced algorithms from combinatorics, graph theory, flow theory, matching theory, geometry, automata theory, . . .:

- Rostering under balancing, counting, and coverage constraints
- Scheduling under resource constraints
- Geometrical constraints between points, segments, . . .
  - . . .
Constraint Modelling

Pride:

Constraint programming represents one of the closest approaches computer science has yet made to the Holy Grail of programming: the user states the problem, the computer solves it.

— Eugene Freuder

Prejudice:

The contribution of the article should be the reduction of an engineering problem to a known optimization format. [...] showcases pseudo code [...] submit this work to a journal interested in code semantics [...].

— Reviewer of a paper of ours at a prestigious OR journal
Outline

1 What is Constraint Technology?
   - Constraint Problems
   - Constraint Technology
   - Constraint Modelling
   - Constraint Solving by Global Search
   - Constraint Solving by Local Search
   - History & Success Stories

2 Enabling Basic Research and Tools
   - Abstract Constraint Modelling
   - Symmetry
   - Constraint-Based Local Search

3 Technology Transfer
   - Air-Traffic Management
   - Computational Biology
   - Computational Finance
### Example (TSC: Sweden partial assignment)

<table>
<thead>
<tr>
<th></th>
<th>Ali</th>
<th>Dan</th>
<th>Eva</th>
<th>Jim</th>
<th>Leo</th>
<th>Mia</th>
<th>Ulla</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Birka</strong></td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td><strong>Falun</strong></td>
<td>✔</td>
<td>X</td>
<td>X</td>
<td>✔</td>
<td>✔</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td><strong>Lund</strong></td>
<td>✔</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td><strong>Mora</strong></td>
<td>X</td>
<td>✔</td>
<td>X</td>
<td>✔</td>
<td>X</td>
<td>✔</td>
<td>X</td>
</tr>
<tr>
<td><strong>Sigtuna</strong></td>
<td>?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Uppsala</strong></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Ystad</strong></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Ali cannot be a judge of Sigtuna as that would violate the second constraint (every judge visits $c = 3$ tourist sites). Actually, Ali cannot be a judge of Mora, Uppsala, or Ystad either, for the same reason, and this was already inferred when trying the decision that Ali be a judge of Lund!
### Example (TSC: Sweden partial assignment)

<table>
<thead>
<tr>
<th></th>
<th>Ali</th>
<th>Dan</th>
<th>Eva</th>
<th>Jim</th>
<th>Leo</th>
<th>Mia</th>
<th>Ulla</th>
</tr>
</thead>
<tbody>
<tr>
<td>Birka</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>×</td>
<td></td>
<td>✔</td>
<td>×</td>
</tr>
<tr>
<td>Falun</td>
<td>✔</td>
<td>×</td>
<td>×</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td>×</td>
</tr>
<tr>
<td>Lund</td>
<td>✔</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>✔</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Mora</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Sigtuna</td>
<td></td>
<td></td>
<td>✔</td>
<td>✔</td>
<td></td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Uppsala</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ystad</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

Ali cannot be a judge of Sigtuna as that would violate the second constraint (every judge visits \(c = 3\) tourist sites).

Actually, Ali cannot be a judge of Mora, Uppsala, or Ystad either, for the same reason, and this was already inferred when trying the decision that Ali be a judge of Lund!
### Constraint Solving by Global Search

#### Example (TSC: Sweden partial assignment)

<table>
<thead>
<tr>
<th></th>
<th>Ali</th>
<th>Dan</th>
<th>Eva</th>
<th>Jim</th>
<th>Leo</th>
<th>Mia</th>
<th>Ulla</th>
</tr>
</thead>
<tbody>
<tr>
<td>Birka</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Falun</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Lund</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Mora</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Sigtuna</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Uppsala</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Ystad</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Ali **cannot** be a judge of Sigtuna as that would violate the second constraint (every judge visits $c = 3$ tourist sites).

Actually, Ali **cannot** be a judge of Mora, Uppsala, or Ystad either, for the same reason, and this was already inferred when trying the decision that Ali be a judge of Lund!
### Example (TSC: Sweden partial assignment)

<table>
<thead>
<tr>
<th></th>
<th>Ali</th>
<th>Dan</th>
<th>Eva</th>
<th>Jim</th>
<th>Leo</th>
<th>Mia</th>
<th>Ulla</th>
</tr>
</thead>
<tbody>
<tr>
<td>Birka</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Falun</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Lund</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Mora</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Sigtuna</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Uppsala</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Ystad</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Ali **cannot** be a judge of Sigtuna as that would violate the second constraint (every judge visits $c = 3$ tourist sites). Actually, Ali **cannot** be a judge of Mora, Uppsala, or Ystad either, for the same reason, and this was already inferred when trying the decision that Ali be a judge of Lund!
### Constraint Solving by Global Search

#### Example (TSC: Sweden partial assignment)

<table>
<thead>
<tr>
<th>City</th>
<th>Ali</th>
<th>Dan</th>
<th>Eva</th>
<th>Jim</th>
<th>Leo</th>
<th>Mia</th>
<th>Ulla</th>
</tr>
</thead>
<tbody>
<tr>
<td>Birka</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
</tr>
<tr>
<td>Falun</td>
<td>✔</td>
<td>✗</td>
<td>✗</td>
<td>✔</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
</tr>
<tr>
<td>Lund</td>
<td>✔</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Mora</td>
<td>✗</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
<td>✗</td>
</tr>
<tr>
<td>Sigtuna</td>
<td>✗</td>
<td></td>
<td></td>
<td>✔</td>
<td></td>
<td></td>
<td>✗</td>
</tr>
<tr>
<td>Uppsala</td>
<td>✗</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ystad</td>
<td>✗</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Ali **cannot** be a judge of Sigtuna as that would violate the second constraint (every judge visits \( c = 3 \) tourist sites). Actually, Ali **cannot** be a judge of Mora, Uppsala, or Ystad either, for the same reason, and this was already inferred when trying the decision that Ali be a judge of Lund!
### Example (TSC: Sweden partial assignment)

<table>
<thead>
<tr>
<th></th>
<th>Ali</th>
<th>Dan</th>
<th>Eva</th>
<th>Jim</th>
<th>Leo</th>
<th>Mia</th>
<th>Ulla</th>
</tr>
</thead>
<tbody>
<tr>
<td>Birka</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Falun</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Lund</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>Mora</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>x</td>
</tr>
<tr>
<td>Sigtuna</td>
<td>x</td>
<td>?</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Uppala</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Ystad</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
## Constraint Solving by Global Search

### Example (TSC: Sweden partial assignment)

<table>
<thead>
<tr>
<th>Location</th>
<th>Ali</th>
<th>Dan</th>
<th>Eva</th>
<th>Jim</th>
<th>Leo</th>
<th>Mia</th>
<th>Ulla</th>
</tr>
</thead>
<tbody>
<tr>
<td>Birka</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Falun</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Lund</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Mora</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>x</td>
</tr>
<tr>
<td>Sigtuna</td>
<td>x</td>
<td>?</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
</tr>
<tr>
<td>Uppsala</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ystad</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Decision:** Dan is a judge of Sigtuna. (✓ decisions first.)
## Constraint Solving by Global Search

### Example (TSC: Sweden partial assignment)

<table>
<thead>
<tr>
<th>City</th>
<th>Ali</th>
<th>Dan</th>
<th>Eva</th>
<th>Jim</th>
<th>Leo</th>
<th>Mia</th>
<th>Ulla</th>
</tr>
</thead>
<tbody>
<tr>
<td>Birka</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✗️</td>
<td>✗️</td>
<td>✗️</td>
<td>✗️</td>
</tr>
<tr>
<td>Falun</td>
<td>✔️</td>
<td>✗️</td>
<td>✗️</td>
<td>✔️</td>
<td>✔️</td>
<td>✗️</td>
<td>✗️</td>
</tr>
<tr>
<td>Lund</td>
<td>✔️</td>
<td>✗️</td>
<td>✗️</td>
<td>✗️</td>
<td>✗️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Mora</td>
<td>✗️</td>
<td>✔️</td>
<td>✗️</td>
<td>✔️</td>
<td>✗️</td>
<td>✔️</td>
<td>✗️</td>
</tr>
<tr>
<td>Sigtuna</td>
<td>✗️</td>
<td>✔️</td>
<td>✗️</td>
<td>✗️</td>
<td>✔️</td>
<td>✔️</td>
<td>✗️</td>
</tr>
<tr>
<td>Uppsala</td>
<td>✗️</td>
<td>✗️</td>
<td>✔️</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ystad</td>
<td>✗️</td>
<td>✗️</td>
<td>✔️</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Decision:** Dan is a judge of Sigtuna. (✔️ decisions first.)
### Example (TSC: Sweden partial assignment)

<table>
<thead>
<tr>
<th></th>
<th>Ali</th>
<th>Dan</th>
<th>Eva</th>
<th>Jim</th>
<th>Leo</th>
<th>Mia</th>
<th>Ulla</th>
</tr>
</thead>
<tbody>
<tr>
<td>Birka</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Falun</td>
<td>✔</td>
<td>✗</td>
<td>✗</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✗</td>
</tr>
<tr>
<td>Lund</td>
<td>✔</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Mora</td>
<td>✗</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
<td>✗</td>
</tr>
<tr>
<td>Sigtuna</td>
<td>✗</td>
<td>✔</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✔</td>
<td>✗</td>
</tr>
<tr>
<td>Uppsala</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Ystad</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

**Inference:** Dan cannot be a judge of Uppsala and Ystad as otherwise the second constraint (every judge visits \( c = 3 \) tourist sites) would be violated for Dan.
## Constraint Solving by Global Search

### Example (TSC: Sweden partial assignment)

<table>
<thead>
<tr>
<th></th>
<th>Ali</th>
<th>Dan</th>
<th>Eva</th>
<th>Jim</th>
<th>Leo</th>
<th>Mia</th>
<th>Ulla</th>
</tr>
</thead>
<tbody>
<tr>
<td>Birka</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Falun</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Lund</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Mora</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Sigtuna</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Upssa</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ystad</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Inference:** Dan cannot be a judge of Uppsala and Ystad as otherwise the second constraint (every judge visits \( c = 3 \) tourist sites) would be violated for Dan.
### Constraint Solving by Global Search

#### Example (TSC: Sweden partial assignment)

<table>
<thead>
<tr>
<th></th>
<th>Ali</th>
<th>Dan</th>
<th>Eva</th>
<th>Jim</th>
<th>Leo</th>
<th>Mia</th>
<th>Ulla</th>
</tr>
</thead>
<tbody>
<tr>
<td>Birka</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Falun</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Lund</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Mora</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Sigtuna</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Uppsala</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Ystad</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

**Inference:** Eva, Jim, and Mia cannot be judges of Sigtuna as otherwise the third constraint (every pair of sites is visited by $\lambda = 1$ common judge) would be violated.
### Example (TSC: Sweden partial assignment)

<table>
<thead>
<tr>
<th></th>
<th>Ali</th>
<th>Dan</th>
<th>Eva</th>
<th>Jim</th>
<th>Leo</th>
<th>Mia</th>
<th>Ulla</th>
</tr>
</thead>
<tbody>
<tr>
<td>Birka</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Falun</td>
<td>✓</td>
<td>X</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Lund</td>
<td>✓</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
</tr>
<tr>
<td>Mora</td>
<td>X</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
<td>X</td>
</tr>
<tr>
<td>Sigtuna</td>
<td>X</td>
<td>✓</td>
<td>X</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Uppsala</td>
<td>X</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
</tr>
<tr>
<td>Ystad</td>
<td>X</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

**Inference:** Eva, Jim, and Mia **cannot** be judges of Sigtuna as otherwise the third constraint (every pair of sites is visited by $\lambda = 1$ common judge) would be violated.
### Example (TSC: Sweden partial assignment)

<table>
<thead>
<tr>
<th></th>
<th>Ali</th>
<th>Dan</th>
<th>Eva</th>
<th>Jim</th>
<th>Leo</th>
<th>Mia</th>
<th>Ulla</th>
</tr>
</thead>
<tbody>
<tr>
<td>Birka</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Falun</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Lund</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Mora</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>x</td>
</tr>
<tr>
<td>Sigtuna</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Uppsala</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ystad</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Inference:** Leo and Ulla must now be judges of Sigtuna as otherwise the first constraint (every tourist site is visited by \( r = 3 \) judges) would be violated for Sigtuna.
## Example (TSC: Sweden partial assignment)

<table>
<thead>
<tr>
<th></th>
<th>Ali</th>
<th>Dan</th>
<th>Eva</th>
<th>Jim</th>
<th>Leo</th>
<th>Mia</th>
<th>Ulla</th>
</tr>
</thead>
<tbody>
<tr>
<td>Birka</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Falun</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Lund</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Mora</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>x</td>
</tr>
<tr>
<td>Sigtuna</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>Uppsala</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ystad</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Inference:** Leo and Ulla must now be judges of Sigtuna as otherwise the first constraint (every tourist site is visited by \( r = 3 \) judges) would be violated for Sigtuna.
### Constraint Solving by Global Search

#### Example (TSC: Sweden partial assignment)

<table>
<thead>
<tr>
<th></th>
<th>Ali</th>
<th>Dan</th>
<th>Eva</th>
<th>Jim</th>
<th>Leo</th>
<th>Mia</th>
<th>Ulla</th>
</tr>
</thead>
<tbody>
<tr>
<td>Birka</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Falun</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Lund</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Mora</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>x</td>
</tr>
<tr>
<td>Sigtuna</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>Uppsala</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ystad</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Inference:** Eva must now be a judge of Uppsala and Ystad as otherwise the second constraint (every judge visits \(c = 3\) tourist sites) would be violated for Eva.
### Constraint Solving by Global Search

#### Example (TSC: Sweden partial assignment)

<table>
<thead>
<tr>
<th></th>
<th>Ali</th>
<th>Dan</th>
<th>Eva</th>
<th>Jim</th>
<th>Leo</th>
<th>Mia</th>
<th>Ulla</th>
</tr>
</thead>
<tbody>
<tr>
<td>Birka</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Falun</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Lund</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>Mora</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>x</td>
</tr>
<tr>
<td>Sigtuna</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>Uppsala</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ystad</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Inference:** Eva must now be a judge of Uppsala and Ystad as otherwise the second constraint (every judge visits \(c = 3\) tourist sites) would be violated for Eva.
### Constraint Solving by Global Search

#### Example (TSC: Sweden partial assignment)

<table>
<thead>
<tr>
<th></th>
<th>Ali</th>
<th>Dan</th>
<th>Eva</th>
<th>Jim</th>
<th>Leo</th>
<th>Mia</th>
<th>Ulla</th>
</tr>
</thead>
<tbody>
<tr>
<td>Birka</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Falun</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
</tr>
<tr>
<td>Lund</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Mora</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>x</td>
</tr>
<tr>
<td>Sigtuna</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>Uppsala</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ystad</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fixpoint reached: No more propagation possible.
### Constraint Solving by Global Search

**Example (TSC: Sweden partial assignment)**

<table>
<thead>
<tr>
<th></th>
<th>Ali</th>
<th>Dan</th>
<th>Eva</th>
<th>Jim</th>
<th>Leo</th>
<th>Mia</th>
<th>Ulla</th>
</tr>
</thead>
<tbody>
<tr>
<td>Birka</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Falun</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Lund</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Mora</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Sigtuna</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Uppsala</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ystad</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Decision:** Jim is a judge of Uppsala. (✓ decisions first.)

**Inference:** etc.
Example (Local Propagation to Domain Consistency)

Find \( a \in \{1, 2, \ldots, 9\} \) and \( b \in \{0, 1, \ldots, 8\} \) such that

\[
\begin{align*}
2 \cdot a + 4 \cdot b &= 24 \\
a + b &= 9
\end{align*}
\]

\[
\begin{array}{cccccccccc}
a & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
b & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8
\end{array}
\]
Example (Local Propagation to \textbf{Domain} Consistency)

Find \(a \in \{1, 2, \ldots, 9\}\) and \(b \in \{0, 1, \ldots, 8\}\) such that

\[
2 \cdot a + 4 \cdot b = 24 \\
\begin{align*}
a + b &= 9
\end{align*}
\]

Posting \(2 \cdot a + 4 \cdot b = 24\): \textbf{pruning} unwitnessed values of \(a\):

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(b)</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
</tbody>
</table>
Example (Local Propagation to **Domain** Consistency)

Find $a \in \{1, 2, \ldots, 9\}$ and $b \in \{0, 1, \ldots, 8\}$ such that

\[
\begin{align*}
2 \cdot a + 4 \cdot b &= 24 \\
\quad a + b &= 9
\end{align*}
\]

Posting $2 \cdot a + 4 \cdot b = 24$: **pruning** unwitnessed values of $a$:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$b$</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
</tbody>
</table>
Constraint Solving by Global Search

Example (Local Propagation to Domain Consistency)

Find \( a \in \{1, 2, \ldots, 9\} \) and \( b \in \{0, 1, \ldots, 8\} \) such that

\[
2 \cdot a + 4 \cdot b = 24 \\
a + b = 9
\]

Posting \( 2 \cdot a + 4 \cdot b = 24 \): pruning unwitnessed values of \( b \):

<table>
<thead>
<tr>
<th>( a )</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>( b )</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
</tbody>
</table>
Example (Local Propagation to Domain Consistency)

Find \( a \in \{1, 2, \ldots, 9\} \) and \( b \in \{0, 1, \ldots, 8\} \) such that

\[
2 \cdot a + 4 \cdot b \quad = \quad 24 \\
\]

\[
a + b \quad = \quad 9
\]

Posting \( 2 \cdot a + 4 \cdot b = 24 \): **pruning** unwitnessed values of \( b \):

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>8</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Constraint Solving by Global Search

Example (Local Propagation to Domain Consistency)

Find \( a \in \{1, 2, \ldots, 9\} \) and \( b \in \{0, 1, \ldots, 8\} \) such that

\[
\begin{align*}
2 \cdot a + 4 \cdot b & = 24 \\
a + b & = 9
\end{align*}
\]

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

Suspending \( 2 \cdot a + 4 \cdot b = 24 \), as it is not definitely true.
Example (Local Propagation to Domain Consistency)

Find \( a \in \{1, 2, \ldots, 9\} \) and \( b \in \{0, 1, \ldots, 8\} \) such that

\[
2 \cdot a + 4 \cdot b = 24
\]
\[
a + b = 9
\]

Posting \( a + b = 9 \): pruning unwitnessed values of \( a \):

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

TDB / TAPVES
Uppsala University
- 21 -
Pierre.Flener@it.uu.se
Example (Local Propagation to Domain Consistency)

Find $a \in \{1, 2, \ldots, 9\}$ and $b \in \{0, 1, \ldots, 8\}$ such that

\[
\begin{align*}
2 \cdot a + 4 \cdot b &= 24 \\
a + b &= 9
\end{align*}
\]

Posting $a + b = 9$: pruning unwitnessed values of $a$:

<table>
<thead>
<tr>
<th>$a$</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$b$</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>
Constraint Solving by Global Search

Example (Local Propagation to \textbf{Domain} Consistency)

Find $a \in \{1, 2, \ldots, 9\}$ and $b \in \{0, 1, \ldots, 8\}$ such that

\[
2 \cdot a + 4 \cdot b = 24 \\
a + b = 9
\]

Posting $a + b = 9$: \textit{pruning} unwitnessed values of $b$:

\[
\begin{array}{cccc}
\hline
a & & 4 & 6 \\
\hline
b & 2 & 3 & 4 & 5 \\
\hline
\end{array}
\]
Example (Local Propagation to \textbf{Domain Consistency})

Find $a \in \{1, 2, \ldots, 9\}$ and $b \in \{0, 1, \ldots, 8\}$ such that

\[2 \cdot a + 4 \cdot b = 24\]
\[a + b = 9\]

Posting $a + b = 9$: \textbf{pruning} unwitnessed values of $b$:

\[
\begin{array}{|c|c|c|c|}
\hline
a & & 4 & 6 \\
\hline
b & 2 & 3 & 4 & 5 \\
\hline
\end{array}
\]
Example (Local Propagation to Domain Consistency)

Find \( a \in \{1, 2, \ldots, 9\} \) and \( b \in \{0, 1, \ldots, 8\} \) such that

\[
2 \cdot a + 4 \cdot b = 24 \\
a + b = 9
\]

<table>
<thead>
<tr>
<th></th>
<th>4</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td></td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>3</td>
<td>5</td>
</tr>
</tbody>
</table>
Example (Local Propagation to Domain Consistency)

Find \( a \in \{1, 2, \ldots, 9\} \) and \( b \in \{0, 1, \ldots, 8\} \) such that

\[
\begin{align*}
2 \cdot a + 4 \cdot b &= 24 \\
 2 \cdot a + b &= 9
\end{align*}
\]

Waking \( 2 \cdot a + 4 \cdot b = 24 \): pruning unwitnessed values of \( a \):

\[
\begin{array}{cccc}
a & & 4 & 6 \\
b & 3 & 5 & \\
\end{array}
\]
Example (Local Propagation to Domain Consistency)

Find $a \in \{1, 2, \ldots, 9\}$ and $b \in \{0, 1, \ldots, 8\}$ such that

$$2 \cdot a + 4 \cdot b = 24$$
$$a + b = 9$$

Waking $2 \cdot a + 4 \cdot b = 24$: pruning unwitnessed values of $a$:

<table>
<thead>
<tr>
<th>$a$</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$b$</td>
<td>3</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Find $a \in \{1, 2, \ldots, 9\}$ and $b \in \{0, 1, \ldots, 8\}$ such that

$$2 \cdot a + 4 \cdot b = 24$$
$$a + b = 9$$

Waking $2 \cdot a + 4 \cdot b = 24$: pruning unwitnessed values of $b$:
Example (Local Propagation to Domain Consistency)

Find \( a \in \{1, 2, \ldots , 9\} \) and \( b \in \{0, 1, \ldots , 8\} \) such that

\[
2 \cdot a + 4 \cdot b = 24 \\
a + b = 9
\]

Waking \( 2 \cdot a + 4 \cdot b = 24 \): pruning unwitnessed values of \( b \):

<table>
<thead>
<tr>
<th>a</th>
<th></th>
<th></th>
<th></th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>3</td>
<td>5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TDB / TAPVES

Uppsala University

Pierre.Flener@it.uu.se
Example (Local Propagation to Domain Consistency)

Find \(a \in \{1, 2, \ldots, 9\}\) and \(b \in \{0, 1, \ldots, 8\}\) such that

\[
\begin{align*}
2 \cdot a + 4 \cdot b & = 24 \\
a + b & = 9
\end{align*}
\]

Deactivating \(2 \cdot a + 4 \cdot b = 24\), as it is definitely true.
Example (Local Propagation to Domain Consistency)

Find \( a \in \{1, 2, \ldots, 9\} \) and \( b \in \{0, 1, \ldots, 8\} \) such that

\[
\begin{align*}
2 \cdot a + 4 \cdot b &= 24 \\
    a + b &= 9
\end{align*}
\]

Waking \( a + b = 9 \): pruning unwitnessed values of \( a \):

\[
\begin{array}{cccccc}
a & \text{ } & \text{ } & \text{ } & \text{ } & 6 \\
3 & \text{ } & \text{ } & \text{ } & \text{ } & \\
\end{array}
\]
Example (Local Propagation to Domain Consistency)

Find \(a \in \{1, 2, \ldots, 9\}\) and \(b \in \{0, 1, \ldots, 8\}\) such that

\[
2 \cdot a + 4 \cdot b = 24
\]

\[
a + b = 9
\]

Waking \(a + b = 9\): pruning unwitnessed values of \(a\):

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| |</p>
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
</tr>
</tbody>
</table>
Constraint Solving by Global Search

Example (Local Propagation to Domain Consistency)

Find \( a \in \{1, 2, \ldots, 9\} \) and \( b \in \{0, 1, \ldots, 8\} \) such that

\[
2 \cdot a + 4 \cdot b = 24 \\
a + b = 9
\]

Waking \( a + b = 9 \): pruning unwitnessed values of \( b \):

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Constraint Solving by Global Search

Example (Local Propagation to Domain Consistency)

Find \( a \in \{1, 2, \ldots, 9\} \) and \( b \in \{0, 1, \ldots, 8\} \) such that

\[
\begin{align*}
2 \cdot a + 4 \cdot b &= 24 \\
a + b &= 9
\end{align*}
\]

Waking \( a + b = 9 \): pruning unwitnessed values of \( b \):

\[
\begin{array}{cccc}
\hline
a & & & 6 \\
\hline
b & & 3 & \\
\hline
\end{array}
\]
Constraint Solving by Global Search

Example (Local Propagation to Domain Consistency)

Find \( a \in \{1, 2, \ldots, 9\} \) and \( b \in \{0, 1, \ldots, 8\} \) such that

\[
\begin{align*}
2 \cdot a + 4 \cdot b &= 24 \\
2 \cdot a + b &= 9
\end{align*}
\]


Deactivating \( a + b = 9 \), as it is definitely true.
Constraint Solving by Global Search

Example (Local Propagation to Domain Consistency)

Find \( a \in \{1, 2, \ldots, 9\} \) and \( b \in \{0, 1, \ldots, 8\} \) such that

\[
\begin{align*}
2 \cdot a + 4 \cdot b &= 24 \\
\quad a + b &= 9
\end{align*}
\]

\[
\begin{array}{ccc}
& a & \\
\hline
b & 3 & 6 \\
\end{array}
\]

No suspended constraints: All solutions found. No search!
Example (Local Propagation to Domain Consistency)

Find \( a \in \{1, 2, \ldots, 9\} \) and \( b \in \{0, 1, \ldots, 8\} \) such that

\[
\begin{align*}
2 \cdot a + 4 \cdot b &= 24 \\
2 \cdot a + b &= 9
\end{align*}
\]

This general propagation method works for all systems of constraints (linear or not, equalities or inequalities, etc), no matter how many constraints and decision variables.
Example (Local Propagation to **Bounds** Consistency)

Find $a \in \{1, 2, \ldots, 9\}$ and $b \in \{0, 1, \ldots, 8\}$ such that

$$2 \cdot a + 4 \cdot b = 24$$

Posting $2 \cdot a + 4 \cdot b = 24$: pruning unwitnessed values of $b$:

<table>
<thead>
<tr>
<th>$a$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$b$</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
</tbody>
</table>

Suspending $2 \cdot a + 4 \cdot b = 24$, as it is not definitely true. Suspended constraints: No solutions found yet. Search!
Constraint Solving by Global Search

Example (Local Propagation to Bounds Consistency)

Find \( a \in \{1, 2, \ldots, 9\} \) and \( b \in \{0, 1, \ldots, 8\} \) such that

\[
2 \cdot a + 4 \cdot b = 24
\]

Posting \( 2 \cdot a + 4 \cdot b = 24 \): pruning unwitnessed values of \( a \):

<table>
<thead>
<tr>
<th>( a )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>( b )</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
</tbody>
</table>

Suspending \( 2 \cdot a + 4 \cdot b = 24 \), as it is not definitely true.

Suspended constraints: No solutions found yet. Search!
Constraint Solving by Global Search

Example (Local Propagation to \textbf{Bounds} Consistency)

Find \(a \in \{1, 2, \ldots, 9\}\) and \(b \in \{0, 1, \ldots, 8\}\) such that

\[
2 \cdot a + 4 \cdot b = 24
\]

Posting \(2 \cdot a + 4 \cdot b = 24\): \textbf{pruning} unwitnessed values of \(a\):

<table>
<thead>
<tr>
<th>(a)</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>(b)</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
</tbody>
</table>

Suspending \(2 \cdot a + 4 \cdot b = 24\), as it is not definitely true.
Suspended constraints: No solutions found yet. Search!
Example (Local Propagation to **Bounds** Consistency)

Find $a \in \{1, 2, \ldots, 9\}$ and $b \in \{0, 1, \ldots, 8\}$ such that

$$2 \cdot a + 4 \cdot b = 24$$

Posting $2 \cdot a + 4 \cdot b = 24$: **pruning** unwitnessed values of $b$:

<table>
<thead>
<tr>
<th>$a$</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$b$</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>

Suspending $2 \cdot a + 4 \cdot b = 24$, as it is not definitely true. Suspended constraints: No solutions found yet. Search!
Constraint Solving by Global Search

Example (Local Propagation to **Bounds Consistency**)

Find \( a \in \{1, 2, \ldots, 9\} \) and \( b \in \{0, 1, \ldots, 8\} \) such that

\[
2 \cdot a + 4 \cdot b = 24
\]

Posting \( 2 \cdot a + 4 \cdot b = 24 \): **pruning** unwitnessed values of \( b \):

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>a</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>b</strong></td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Suspending \( 2 \cdot a + 4 \cdot b = 24 \), as it is not definitely true. Suspended constraints: No solutions found yet. Search!
Constraint Solving by Global Search

Example (Local Propagation to **Bounds** Consistency)

Find \( a \in \{1, 2, \ldots, 9\} \) and \( b \in \{0, 1, \ldots, 8\} \) such that

\[
2 \cdot a + 4 \cdot b = 24
\]

Posting \( 2 \cdot a + 4 \cdot b = 24 \): pruning unwitnessed values of \( b \):

<table>
<thead>
<tr>
<th>( a )</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>( b )</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Suspending \( 2 \cdot a + 4 \cdot b = 24 \), as it is not definitely true.

Suspended constraints: No solutions found yet. Search!
Constraint Solving by Global Search

Example (Local Propagation to **Bounds** Consistency)

Find \( a \in \{1, 2, \ldots, 9\} \) and \( b \in \{0, 1, \ldots, 8\} \) such that

\[
2 \cdot a + 4 \cdot b = 24
\]

Posting \( 2 \cdot a + 4 \cdot b = 24 \): pruning unwitnessed values of \( b \):

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>a</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>b</strong></td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Suspending \( 2 \cdot a + 4 \cdot b = 24 \), as it is not definitely true. Suspended constraints: No solutions found yet. Search!
Constraint Solving by Global Search

Local Propagation and Local Consistency

Use of polynomial-time propagators.
Use of incomplete propagators, when necessary/sufficient.

Example (**AllDifferent**(*x_1*, ..., *x_n*))

Bounds consistency in $O(n \log n)$ time, but often $O(n)$ time.
Domain consistency in $O(n^{2.5})$ time.
Compare with the $\Theta(n^2)$ constraints $\forall 1 \leq i < j \leq n : x_i \neq x_j$.

Example (Arithmetic constraints over *n* variables)

Bounds consistency in $O(n)$ time.
Domain consistency in $O(2^n)$ time.
Constraint Solving by Global Search

Global Search Algorithm

1: **post** all given constraints (including propagation)
2: **while** there is at least one suspended constraint **do**
3: **pick** a variable $x$ with $|\text{dom}(x)| \geq 2$
4: **pick** some values $d_i \in \text{dom}(x)$
5: **branch** on mutually exclusive constraints, say $x = d$ and $x \neq d$, or $x > d$ and $x \leq d$
6: **end while**

Heuristics

- Line 3: variable ordering heuristic: smallest domain, . . .
- Line 4: value ordering heuristic: highest, middle, . . .
- Tree exploration: depth-first search, . . . with backtracking when a constraint is definitely false
Outline

1 What is Constraint Technology?
   - Constraint Problems
   - Constraint Technology
   - Constraint Modelling
   - Constraint Solving by Global Search
   - Constraint Solving by Local Search
   - History & Success Stories

2 Enabling Basic Research and Tools
   - Abstract Constraint Modelling
   - Symmetry
   - Constraint-Based Local Search

3 Technology Transfer
   - Air-Traffic Management
   - Computational Biology
   - Computational Finance
Example (TSC: Sweden configuration after $i$ moves)

1. Every tourist site is visited by $r = 3$ judges. (Invariant)
2. Every judge visits $c = 3$ tourist sites.
3. Ex: Mora and Sigtuna have $2 > \lambda = 1$ common judges: let Mia visit Mora instead of Leo.
Example (TSC: Sweden configuration after $i+1$ moves)

<table>
<thead>
<tr>
<th></th>
<th>Ali</th>
<th>Dan</th>
<th>Eva</th>
<th>Jim</th>
<th>Leo</th>
<th>Mia</th>
<th>Ulla</th>
</tr>
</thead>
<tbody>
<tr>
<td>Birka</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Falun</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Lund</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mora</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Sigtuna</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Uppsala</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ystad</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

1. Every tourist site is visited by $r = 3$ judges. (Invariant)
2. Leo visits $2 < c = 3$ sites; Mia visits $4 > c = 3$ sites.
3. Ex: Falun and Lund have $2 > \lambda = 1$ common judges: let Leo visit Falun instead of Mia.
Example (TSC: Sweden configuration after $i+2$ moves)

<table>
<thead>
<tr>
<th></th>
<th>Ali</th>
<th>Dan</th>
<th>Eva</th>
<th>Jim</th>
<th>Leo</th>
<th>Mia</th>
<th>Ulla</th>
</tr>
</thead>
<tbody>
<tr>
<td>Birka</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Falun</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lund</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Mora</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Sigtuna</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Uppsala</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ystad</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

1. Every tourist site is visited by $r = 3$ judges. (Invariant)
2. Every judge visits $c = 3$ tourist sites.
3. Every pair of sites is visited by $\lambda = 1$ common judge.
# Constraint Solving by Local Search

## Local Search Algorithm

1. let $k$ and $k^*$ be the same computed initial configuration
2. for $i := 1$ to $MaxIterations$ do
3.   let $k$ be a picked neighbour configuration of $k$
4.   if $k$ is a solution and is better than $k^*$ then $k^* := k$
5. end for
6. return $k^*$

## Heuristics: What Neighbour To Move To?

- Line 3: assign, flip, swap, add, drop, transfer, . . .

## Meta-Heuristics: How To Escape Local Optima?

- Lines 2 – 5: simulated annealing, tabu search, . . .
Outline

1. What is Constraint Technology?
   - Constraint Problems
   - Constraint Technology
   - Constraint Modelling
   - Constraint Solving by Global Search
   - Constraint Solving by Local Search
   - History & Success Stories

2. Enabling Basic Research and Tools
   - Abstract Constraint Modelling
   - Symmetry
   - Constraint-Based Local Search

3. Technology Transfer
   - Air-Traffic Management
   - Computational Biology
   - Computational Finance
#### History of CT

Stand-alone languages:
- **ALICE** by Jean-Louis Laurière, France, 1976
- **CHIP** at ECRC, Germany, 1987 – 1990, then marketed by Cosytec, France, 1990 – 1992
- **OPL**, by P. Van Hentenryck, USA, and ILOG, France: front-end to both ILOG CP Optimizer and ILOG CPLEX
- **Comet**, by P. Van Hentenryck and L. Michel, USA

Libraries (the first few ones listed are open-source!):
- Prolog: . . ., ECLiPSe, GNU Prolog, SICStus Prolog, . . .
- C++: Gecode, ILOG CP Optimizer (ex Solver), . . .
- Java: Choco, Gecode/J, Koalog, . . .

The Association for Computing Machinery (ACM) identified CT as a strategic direction in computing research.
CT Success Stories I

CT is deployed in many products from industry leaders:

- Planning of satellite missions
- Vehicle production optimisation
- Routing
- Cabling
- Supply chain management
- Crew rostering
- Production scheduling
CT Success Stories II

CT has become the technology of choice in short-term scheduling, timetabling, and rostering.

Scheduling
Logistics software
Control software validation, circuit verification
Resource allocation
Manufacturing
Copier component specification
Constraint programming is the ideal paradigm for encoding correct programs that must run efficiently on multi-processor hardware.

— Mark Wallace
Outline

1 What is Constraint Technology?
   - Constraint Problems
   - Constraint Technology
   - Constraint Modelling
   - Constraint Solving by Global Search
   - Constraint Solving by Local Search
   - History & Success Stories

2 Enabling Basic Research and Tools
   - Abstract Constraint Modelling
   - Symmetry
   - Constraint-Based Local Search

3 Technology Transfer
   - Air-Traffic Management
   - Computational Biology
   - Computational Finance
Outline

1. What is Constraint Technology?
   - Constraint Problems
   - Constraint Technology
   - Constraint Modelling
   - Constraint Solving by Global Search
   - Constraint Solving by Local Search
   - History & Success Stories

2. Enabling Basic Research and Tools
   - Abstract Constraint Modelling
   - Symmetry
   - Constraint-Based Local Search

3. Technology Transfer
   - Air-Traffic Management
   - Computational Biology
   - Computational Finance
Abstract Constraint Modelling

Research Objectives

- **Vision**: Abstract and **solver-independent** constraint modelling will lead to simpler and leaner languages, to more intuitive and analysable models, as well as to more effective model formulation and maintenance.

- **Specific Objectives**: Development of intelligent **analysis** and **compilation** tools, so as to (help a modeller to) translate a high-level model into a lower-level program not unlike what an expert would have written.

- **Benefit**: Empowerment of a wider range of users to unleash the proven benefits of constraint solvers on a broader range of real-life constraint problems.
Outline

1. What is Constraint Technology?
   - Constraint Problems
   - Constraint Technology
   - Constraint Modelling
   - Constraint Solving by Global Search
   - Constraint Solving by Local Search
   - History & Success Stories

2. Enabling Basic Research and Tools
   - Abstract Constraint Modelling
   - Symmetry
   - Constraint-Based Local Search

3. Technology Transfer
   - Air-Traffic Management
   - Computational Biology
   - Computational Finance
**Symmetry**

### Example (Symmetry in the TSC problem)

<table>
<thead>
<tr>
<th></th>
<th>Ali</th>
<th>Dan</th>
<th>Eva</th>
<th>Jim</th>
<th>Leo</th>
<th>Mia</th>
<th>Ulla</th>
</tr>
</thead>
<tbody>
<tr>
<td>Birka</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Falun</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lund</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Mora</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sigtuna</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Uppsala</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Ystad</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- The **tourist sites (row indices)** are indistinguishable.
- ...
### Symmetry

#### Example (Symmetry in the TSC problem)

<table>
<thead>
<tr>
<th></th>
<th>Ali</th>
<th>Dan</th>
<th>Eva</th>
<th>Jim</th>
<th>Leo</th>
<th>Mia</th>
<th>Ulla</th>
</tr>
</thead>
<tbody>
<tr>
<td>Birka</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Falun</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Lund</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mora</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sigtuna</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Uppsala</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ystad</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

- The **tourist sites (row indices)** are indistinguishable.
- ...
Symmetry

Example (Symmetry in the TSC problem)

<table>
<thead>
<tr>
<th></th>
<th>Ali</th>
<th>Dan</th>
<th>Eva</th>
<th>Jim</th>
<th>Leo</th>
<th>Mia</th>
<th>Ulla</th>
</tr>
</thead>
<tbody>
<tr>
<td>Birka</td>
<td>✅</td>
<td>✅</td>
<td>✅</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Falun</td>
<td>✅</td>
<td></td>
<td></td>
<td>✅</td>
<td>✅</td>
<td>✅</td>
<td></td>
</tr>
<tr>
<td>Lund</td>
<td></td>
<td>✅</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✅</td>
</tr>
<tr>
<td>Mora</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✅</td>
<td>✅</td>
<td></td>
</tr>
<tr>
<td>Sigtuna</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Uppsala</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ystad</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

... 

- The judges (column indices) are indistinguishable.
Symmetry

Example (Symmetry in the TSC problem)

<table>
<thead>
<tr>
<th></th>
<th>Ali</th>
<th>Dan</th>
<th>Eva</th>
<th>Jim</th>
<th>Leo</th>
<th>Mia</th>
<th>Ulla</th>
</tr>
</thead>
<tbody>
<tr>
<td>Birka</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Falun</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Lund</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Mora</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Sigtuna</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Uppsala</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ystad</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- The judges (column indices) are indistinguishable.
Symmetry

Problems with Unhandled Symmetry

- For every solution, there are at most as many symmetric solutions as symmetries.
- Worse: For every non-solution, there are at most as many symmetric non-solutions as symmetries.
- Similarly for partial solutions and partial non-solutions.

Hence: A problem solver may waste a lot of time exploring symmetric parts of the search space, when looking for a solution, or all solutions, or optimal solutions.
Symmetry Handling

We have shown that the $|Sites|! \cdot |Judges|!$ combinations of row and column symmetries (that is 25,401,600 symmetries for the Sweden instance) of the TSC problem can be:

- **Detected** automatically, in polynomial time, from an abstract model (in our ESRA language, say).
- **Broken** automatically and partially, in polynomial time and space, by requiring the rows and columns to be lexicographically ordered (another global constraint).

This sufficiently reduces the search space to solve problem instances that are one order of magnitude larger.
Symmetry

Research Objectives

- **Vision**: Real-life problem instances of unconsidered hardness can be solved via better symmetry handling.
- **Specific Objectives**: Automatable, effective, efficient techniques for symmetry **detection** and **breaking**.
- **Benefit**: Burden of symmetry handling shifted from the problem modeller to the developer of problem solvers.
Outline

1. What is Constraint Technology?
   - Constraint Problems
   - Constraint Technology
   - Constraint Modelling
   - Constraint Solving by Global Search
   - Constraint Solving by Local Search
   - History & Success Stories

2. Enabling Basic Research and Tools
   - Abstract Constraint Modelling
   - Symmetry
   - Constraint-Based Local Search

3. Technology Transfer
   - Air-Traffic Management
   - Computational Biology
   - Computational Finance
Research Objectives

■ **Vision**: The design and maintenance of efficient local search algorithms can be simplified even more, with the user having to resort to even less low-level handcoding.

■ **Specific Objectives**: Improve the modelling **versatility**, language **extensibility**, and solving **efficiency** of constraint-based local search.

■ **Benefit**: Increased likelihood of experiments with local search. Hence: Increased chances of environmental, time, financial, or material savings.
Outline

1. What is Constraint Technology?
   - Constraint Problems
   - Constraint Technology
   - Constraint Modelling
   - Constraint Solving by Global Search
   - Constraint Solving by Local Search
   - History & Success Stories

2. Enabling Basic Research and Tools
   - Abstract Constraint Modelling
   - Symmetry
   - Constraint-Based Local Search

3. Technology Transfer
   - Air-Traffic Management
   - Computational Biology
   - Computational Finance
Outline

1. What is Constraint Technology?
   - Constraint Problems
   - Constraint Technology
   - Constraint Modelling
   - Constraint Solving by Global Search
   - Constraint Solving by Local Search
   - History & Success Stories

2. Enabling Basic Research and Tools
   - Abstract Constraint Modelling
   - Symmetry
   - Constraint-Based Local Search

3. Technology Transfer
   - Air-Traffic Management
   - Computational Biology
   - Computational Finance
Example (Demonstration project to EuroControl, the European Organisation for the Safety of Air Navigation)
Air-Traffic Management

Example (Planned temporal profile)
Air-Traffic Management

Example (Resolved temporal profile)

What is Constraint Technology?
Enabling Basic Research and Tools
Technology Transfer
Air-Traffic Management
Computational Biology
Computational Finance
Air-Traffic Management

Example (Planned vertical profile)

\[ z = \begin{cases} 
\text{FL 340} & \text{now} \\
\text{FL 245} & \text{m} \\
\text{p6} & \text{m+L} \\
\text{p5} & \text{m+2L} \\
\text{p4} & \text{p3} \\
\text{p2} & \text{p1} \\
\end{cases} \]

\[ x, y \text{ of chosen airspace} \]
Air-Traffic Management

Example (Planned and resolved vertical profiles)

What is Constraint Technology?
Enabling Basic Research and Tools
Technology Transfer
Air-Traffic Management
Computational Biology
Computational Finance

Air-Traffic Management

- 54 -
Outline

1. What is Constraint Technology?
   - Constraint Problems
   - Constraint Technology
   - Constraint Modelling
   - Constraint Solving by Global Search
   - Constraint Solving by Local Search
   - History & Success Stories

2. Enabling Basic Research and Tools
   - Abstract Constraint Modelling
   - Symmetry
   - Constraint-Based Local Search

3. Technology Transfer
   - Air-Traffic Management
   - Computational Biology
   - Computational Finance
Cross-Fertilisation

Biology can benefit from constraint technology. Constraint technology can benefit from biology:

Indeed, the current CT research on

- Symmetry handling
- Graph decision variables
- Our *tree* partitioning constraint (see below)
- . . .

was actually motivated by biological problems, but has also found applications in routing, linguistics, etc.
Computational Biology

Existing Collaboration

With:
- École des Mines de Nantes, France
- University of East Anglia, Norwich, UK
- Sabancı University, İstanbul, Turkey

On:
- Phylogenetic supertree construction
- RNA secondary structure prediction
- Haplotype inference
- Motif discovery in promoter sequences of DNA
Computational Biology: Phylogenetic Trees

Objective: Construct a supertree that is (maximally) consistent with several given species trees.
Outline

1. What is Constraint Technology?
   - Constraint Problems
   - Constraint Technology
   - Constraint Modelling
   - Constraint Solving by Global Search
   - Constraint Solving by Local Search
   - History & Success Stories

2. Enabling Basic Research and Tools
   - Abstract Constraint Modelling
   - Symmetry
   - Constraint-Based Local Search

3. Technology Transfer
   - Air-Traffic Management
   - Computational Biology
   - Computational Finance
### Computational Finance

**Example (Financial investment instrument design with Merrill Lynch, New York, USA)**

<table>
<thead>
<tr>
<th></th>
<th>Acer</th>
<th>Apple</th>
<th>Dell</th>
<th>HP</th>
<th>IBM</th>
<th>Siemens</th>
<th>Sony</th>
</tr>
</thead>
<tbody>
<tr>
<td>B 1</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B 2</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>B 3</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>B 4</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>B 5</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>B 6</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>B 7</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

1. Every basket contains $r = 3$ credits.
2. (Every credit is in any amount of baskets.)
3. **Minimise** maximum overlap $\lambda$ between any two baskets.