G53CLP
Constraint Logic Programming

Dr Rong Qu

Modeling CSPs – Case Study I
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 C. Freuder
1999
Constraint Programming

- The computer solves it
  - A number of CP software tools available
  - One can use the tools effectively after some training
  - Necessary settings

- We’ve seen foundations of constraint satisfaction
  - Many software tools are based on this theory and algorithms
Constraint Programming

- The user states the problem
  - Not the way we write/code the problem using languages of tools
  - How we model the problem effectively in formulations

- In practice
  - Model the problem understood by computers
  - Code in specific languages
  - Own codes for specific problems
Solving CSP

- How can we **state** the problem?
  - Languages of different tools are different
  - Model is the same
Solving CSP

- To improve the efficiency of CP solving
  - Improve the algorithms
    - Specialised domain information
    - Higher level (complex) constraints (alldifferent)
  - Different models for the problem
    - Add redundant constraints
    - More fundamental way
    - Number of variables
    - Number of constraints
Solving CSP

G53CLP – Constraint Logic Programming

Dr R. Qu

7 of 26
“... makes stating combinatorial problems easy in terms of a constraint satisfaction problem, however, it is more complicated to design solvable models.”

Roman Barták
2002
Solving CSP

- Primary role of constraint programmers
  - Model the problem
  - Translating high level constraints in the problem
  \[\rightarrow\]
  low level constraints supported by CP solvers
These Two Lectures

- Case study on AI puzzle *n-Queen*

- Some in-class exercises
- Interactive
  - Questions
  - Discussions

- You can see the implementation/coding of these models in ILOG Solver
The n-Queen Problem

Any solution?
How many solutions?
The n-Queen Problem
The n-Queen Problem

Any solution?
How many solutions?

Some demos
The n-Queen Problem

- Some questions
  - How about $n$ is (much) larger?
  - Is the solution you have the only one?
  - ...

- Number of solutions for n-queen*

<table>
<thead>
<tr>
<th>$n$:</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>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>10</td>
<td>4</td>
<td>40</td>
<td>92</td>
<td>352</td>
<td>724</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$n$:</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2,680</td>
<td>14,200</td>
<td>73,712</td>
<td>365,596</td>
<td>2,279,184</td>
<td>...</td>
</tr>
</tbody>
</table>

- from wikipedia
The n-Queen Problem

- Why modeling?
  - Large computational time
  - We have large problems
  - We have problems of different sizes
  - We have different problem (but can be modeled the same, see later)
The n-Queen Problem – model 1

- **Variables**
  - $x_{i,j}$
    - 1, if there is a queen in row $i$, column $j$;
    - 0, otherwise

- **Domain**
  - $x_{i,j} = \{0, 1\}$
The n-Queen Problem – model 1

Constraints

1. One queen each row

2. One queen each column
The n-Queen Problem – model 1

Constraints

3. One queen diagonally
   ?
The n-Queen Problem – model 2

- Variables
  - $x_1, x_2, ..., x_n$: position of queens on the chessboard

- Domain
  - $\{0 \ldots n^2-1\}$: tile index of each queen placed
The n-Queen Problem – model 2

Constraint

1. One queen each row
2. One queen each column
The n-Queen Problem – model 2

- Constraint

3. One queen diagonally
   ?
The n-Queen Problem – model 3

- Variables
  - $x_1, x_2, \ldots, x_n$: rows of the chessboard

- Domain
  - $\{1 \ldots n\}$: column index of each queen placed
The n-Queen Problem – model 3

- Constraint

1. One queen each column
   - ?
2. One queen diagonally
   - ?
3. One queen each row
   - ?

Row constraint is in the formulation
The n-Queen Problem – models

- The search space
  - contains all possible assignments

- For the above three models
  - module 3: \( \{x_1, ..., x_n\}, \{1, ..., n\} \)
  - module 2: \( \{x_1, ..., x_n\}, \{0, ..., (n^2 - 1)\} \)
  - module 1: \( \{x_{i,j}\}, \{0, 1\} \)

<table>
<thead>
<tr>
<th>model</th>
<th>variables</th>
<th>domain</th>
<th>search space</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>n</td>
<td>n</td>
<td>( n! ) (or ( n^n ))</td>
</tr>
</tbody>
</table>
The n-Queen Problem – models

- Fewer number of variables with small domains of variables are preferable
  - Smaller search space
  - Problems solved more quickly

- Fewer number of variables with large domains of variables are preferable

<table>
<thead>
<tr>
<th>model</th>
<th>variables</th>
<th>domain</th>
<th>search space</th>
<th>n=4</th>
<th>n = 8</th>
<th>n = 10</th>
<th>n = 20</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>n</td>
<td>n</td>
<td>n!</td>
<td>24</td>
<td>4 E4</td>
<td>3.6 E6</td>
<td>2.4 E18</td>
</tr>
<tr>
<td>2</td>
<td>n</td>
<td>n²</td>
<td>(n²)^n</td>
<td>6.6 E4</td>
<td>2.8 E14</td>
<td>1 E20</td>
<td>1.1 E52</td>
</tr>
<tr>
<td>1</td>
<td>n²</td>
<td>2</td>
<td>2^(n²)</td>
<td>6.6 E4</td>
<td>1.84 E19</td>
<td>1.27 E30</td>
<td>2.6 E120</td>
</tr>
</tbody>
</table>
Summary

- Modeling in constraint programming

- Case study on 8-queen problem
  - Comparison on 3 models

- Model in constraint programming
  - Few number of variables
  - Small domain