CSP: An Introduction

Pedro Meseguer
IIIA-CSIC
Bellaterra, Spain

Overview

Definitions
Examples
• Map colouring
• N-queens
• Car sequencing
• Job-shop scheduling
Relevance
Constraint graphs
**Some Definitions**

Constraint Network (CN): \((X, D, C)\)
- \(X = \{x_1, x_2, \ldots, x_n\}\) variables
- \(D = \{d_1, d_2, \ldots, d_n\}\) domains (finite)
- \(C = \{c_1, c_2, \ldots, c_r\}\) constraints

\[c \in C\] var\((c) = \{x_i, x_j, \ldots, x_k\}\] scope

\[rel(c) \subseteq d_i \times d_j \times \ldots \times d_k\] permitted tuples

\[arity(c) = |\text{var}(c)|\] (unary, binary, ternary, …)

**Constraint Satisfaction Problem (CSP):**
- CN solving: assign. satisfying every constraint
- NP-complete task

---

**Example: Map Colouring**

**GOAL:** Given a map and a number of colours, assign a colour to each region such that adjacent regions have different colours

**Formulation:**
- Variables: regions
- Domains: colours
- Constraints: if \(\text{adjacent}(x_i, x_j)\) then \(x_i \neq x_j\)

**Constraint Graph:**

---
**Example: n-queens**

**GOAL:** Locate $n$ queens in an $n \times n$ chessboard, such that they do not attack each other.

**Formulation:**
- **Variables:** one queen per row
- **Domains:** available columns
- **Constraints:**
  - different columns and different diagonals
  \[ x_i \neq x_j \quad \land \quad |x_i - x_j| \neq |i - j| \]

**Constraint Graph:**

---

**Example: Car Sequencing**

**OPTIONS / MODELS**

<table>
<thead>
<tr>
<th></th>
<th>anti-fog</th>
<th>sun-roof</th>
<th>climatiser</th>
</tr>
</thead>
<tbody>
<tr>
<td>anti-fog</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>sun-roof</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>climatiser</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
</tbody>
</table>

**GOAL:** ordering satisfying capacity constraints

**Formulation:**
- **variables:** $n$ cars to produce
- **domains:** car models
- **constraints:** box capacity
Job-Shop Scheduling

**GIVEN:**  
- $n$ jobs, each with $m$ operations  
- $m$ resources, each operation requires a resource for a period  
- precedence between operations of each job

**GOAL:** can $n$ jobs be performed in time $D$?

Formulation:
- variables: operations
- dominios: start times
- constraints:  
  - precedence  
  - exclusivity

Relevance

**CSP:** formal model to express problems

- Artificial Intelligence  
  - temporal reasoning
- Control Theory  
  - controllers for sensory based robots
- Concurrency  
  - process comm. and synchr.
- Computer Graphics  
  - geometric coherence
- Database Systems  
  - constraint databases
- Bioinformatics  
  - sequence alignment
- Operations research  
  - optimization

Real-life applications
- Production planning
- Staff scheduling
- Resource allocation
- Circuit design
- Option trading
- DNA sequencing
- …
Constraint Graphs

Primal graph:
- Nodes: variables
- Arcs: between two constrained variables

Dual graph:
- Nodes: constraints
- Arcs: between two constraints sharing a variable

Hypergraph:
- Nodes: variables
- Hyperarcs: constraints

Example: Map Colouring

\[ N \neq S \neq F \neq N \]

\[ N \neq F \]
\[ N \neq S \]
\[ S \neq F \]

all-different \((N,S,F)\)

\[ F \]
\[ N \neq S \neq F \]

\[ N \neq F \]
\[ N \neq S \]
\[ S \neq F \]

all-diff\((N,S,F)\)

\[ F \]
\[ N \]
\[ S \]

all-diff