Constraint Networks
basic concepts

Alessandro Farinelli
Department of Computer Science
University of Verona
Verona, Italy
alessandro.farinelli@univr.it
Motivations: Combinatorial Problems

Given a set of possible solutions find the best

Main issue:

Space of possible solutions is huge

Usually exponentially large

Complete search of all solutions is impossible
Graph Colouring

Given a graph

- $k$ colours
- colour each node

such that no two adjacent nodes have the same colour
Combinatorial Problems: Optimisation

Graph Colouring (optimisation)

Given a graph

- \( k \) colours
- colour each node

such that the minimum number of two adjacent nodes have the same colour

Optimisation

Decision

No

No

Best
Portfolio investment
Given a set of investments
Find a subset of them (portfolio)

Such that:
Minimise Risk
Maximise Profit
Combinatorial Problems: Graphical Models

Characteristics:

- A set of **Variables**
- A set of **Domains**, one for each variable
- A set of **Local** functions

**Global** function is an aggregation of local function
Combinatorial Problems: Graphical Models II

Graph Colouring

- Local functions: number of conflict for each link
- Global function: sum of local functions

Graphical Models:
Exploit problem structure
Extremely efficient
Very general, used in many fields:
  - Constraint Reasoning
  - Bayesian Network
  - Error Correcting codes
  - ...
Dense Deployment

To detect events (e.g., vehicle activity)

Energy Harvesting

Energy Neutral Operations

Sense/sleep modes

Sensor Model

Activity can be detected by single sensor

Neighbors (i.e., overlapping sensors) can communicate

Only Neighbors are aware of each other sensors
The Coordination Problem

Energy neutral operation:

Constraints on sense/sleep schedules

Coordination:
Maximise detection probability given constraints on schedules
Minimise periods where no sensor is actively sensing

Combinatorial problem

Similar to Graph Coloring but:
• Overlapping Areas -> weights
• Non binary relationships
W. A. S. Demo

The image displays a software interface for the Adaptive Energy-Aware Sensor Network Demonstrator. The interface includes a map with various sensors marked, and charts showing efficiency and time to detect events. The charts compare different methods: Continuous, Coordinated, Random, and Synchronous.

The map and charts illustrate the performance metrics of the sensor network under various algorithmic conditions, highlighting the effectiveness of different strategies in event detection and energy efficiency.
Field of Constraint Processing

Where it comes from

- Artificial Intelligence (vision)
- Programming Languages (Logic Programming)
- Logic based languages (propositional logic)

Related Areas

- Hardware and Software Verification
- Operation Research (Integer Programming)
- Information Theory (error correcting codes)
- Agents and Multi Agent Systems (coordination)
CP: what can we express with constraints

All problems that can be formulated as follow:

• Given a set of variables and a set of domains
• Find values for variables such that a given relation holds among them

Graph colouring:

• Variables: nodes
• Domain: colour
• Find colour for nodes such that adjacent nodes do not have same colour

N-Queens problem:

• Find positions for N queens on a N by N chessboard such that none of them can eat another in one move
4-Queens problem: first formulation

A possible Formulation:

- 8 variables: 
  \(x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4\)

- No two queens on same row: 
  \(x_1 \neq x_2, x_1 \neq x_3, \ldots\)

- No two queens on same column: 
  \(y_1 \neq y_2, y_1 \neq y_3, \ldots\)

- No two queens on same diagonal: 
  \(|x_1 - x_2| \neq |y_1 - y_2|, \ldots\)

- Example:
  \(x_1 = 1, y_1 = 2\)
  \(x_2 = 2, y_2 = 1\)

  
  \[
  \begin{array}{|c|c|}
  \hline
  Q_2 & \text{ } \\
  \hline
  Q_1 & Q_4 \\
  \hline
  & Q_3 \\
  \hline
  \end{array}
  \]
4-Queens alternative formulation

A (better) Formulation

In every valid solution one column for each queen

- Variables: columns r1, r2, r3, r4
- Domain: rows [1...8]

Constraints:

- Columns are all different
- r1 != r2, ...
- |r1 – r2| != 1, |r1 – r3| != 2, ...

```
Q2
Q1
Q4
Q3
```

\[ r1 = 2 \quad r2 = 1 \]
Constraints encode information

Constraint as information:

- This class is 45 min. Long
- Four nucleotides that make up the a DNA can only combine in a particular sequence
- In a clause all variable are universally quantified
- In a valid n-queen solutions all queens are in different rows

We can exploit constraint to avoid reasoning about useless options

- Encode the n-queens problem with n variables that have n values each
Constraint Network

A constraint network is $R=(X,D,C)$

$X$ set of variables $X = \{x_1, \ldots, x_n\}$

$D$ set of domains $D = \{D_1, \ldots, D_n\}$ $D_i = \{v_1, \ldots, v_{k(i)}\}$

$C$ set of constraints $(S_i, R_i)$ [$S_i \subseteq X$]

- **scope**: variables involved in $R_i$
- $R_i$ subset of cartesian product of variables in $S_i$
- $R_i$ expresses **allowed** tuples over $S_i$

**Solution**: assignment of **all** variables that satisfies all constraints

**Tasks**: consistency check, find one or all solutions, count solutions, find best solution (optimisation)
4-Queens example

Four variables all with domain $[1,\ldots,4]$

<table>
<thead>
<tr>
<th></th>
<th>r1</th>
<th>r2</th>
<th>r3</th>
<th>r4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Q</td>
<td>no</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>no</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>ok</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>ok</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$C_1 = (S_1,R_1)$
$S_1 = \{r_1,r_2\}$
$R_1 = \{(1,3)(1,4)(2,4)(3,1)(4,1)(4,2)\} = R_{1-2}$

$C_4 = (S_4,R_4)$
$S_4 = \{r_2,r_3\}$
$R_4 = \{(1,3)(1,4)(2,4)(3,1)(4,1)(4,2)\} = R_{2-3}$
Solution and partial consistent solutions

Partial Solution

- Assignment of a subset of variables

Consistent partial solution:

- Partial solution that satisfies all the constraints whose scope contains no un-instantiated variables

- A consistent partial solution may not be a subset of a solution

```
<table>
<thead>
<tr>
<th></th>
<th>r1</th>
<th>r2</th>
<th>r3</th>
<th>r4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Q</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>Q</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td>Q</td>
<td></td>
</tr>
</tbody>
</table>

consistent

```

```
<table>
<thead>
<tr>
<th></th>
<th>r1</th>
<th>r2</th>
<th>r3</th>
<th>r4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Q</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>Q</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td>Q</td>
<td></td>
</tr>
</tbody>
</table>

inconsistent

```

```
<table>
<thead>
<tr>
<th></th>
<th>r1</th>
<th>r2</th>
<th>r3</th>
<th>r4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Q</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>Q</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td>Q</td>
<td></td>
</tr>
</tbody>
</table>

solution
```
Map Colouring

Given a map decide whether the map can be coloured with 4 different colours so that no adjacent countries have the same colour.

\[
\begin{array}{c|c}
\hline
x_1 & x_2 \\
\hline
x_3 & x_4 & x_5 \\
\hline
\end{array}
\]

\[C_1 = (\{x_1, x_2\}, x_1 \neq x_2)\]
\[C_2 = (\{x_1, x_3\}, x_1 \neq x_3)\]
\[\ldots\]

Solution
Constraint Network

Map Colouring

4-Queens
Constraint Graph

Primal graph

- Node: variable
- Arc: constraint holding between variables

Map Colouring
Dual Graph

Nodes: constraints’ scopes

Arcs: shared variables

Map Colouring
Crossword Puzzle: Primal graph

Possible words: \{MAP, ARC\}
Only word of correct length

Di : letters of the alphabet
C1 [{x1, x2, x3}, (MAP)(ARC)]
C2 [{x2, x4, x5}, (MAP)(ARC)]
Di : letters of the alphabet
C1 [{x1,x2,x3},(MAP)(ARC)]
C2 [{x2,x4,x5},(MAP)(ARC)]
Hypergraphs and Dual Graphs

\[ x_1, x_2, x_3 \]
\[ x_2, x_4, x_5 \]
Hybergraph and Binary graphs

Can always convert a hypergraph into a binary graph

- The dual graph of an Hypergraph is a binary graph
- We can use it to represent our problem

But each variable has an \textit{exponentially larger domain}

- This is a problem for efficiency
Representing Constraints

Tables

- Show all allowed tuples
  - Words in the crossword puzzle

Arithmetic expressions

- Give an arithmetic expression that allowed tuples should meet
  - $X_1 \neq X_2$ in the n-queen problem

Propositional formula

- Boolean values of variables
  - Boolean values that satisfy the formula
  - $(a \text{ or } b) = \{(0,1)(1,0)(1,1)\}$
Propositional CNF

Consider the set of clauses:

- \{x_1 \text{ or not } x_2, \text{ not } x_2 \text{ or not } x_3, \text{ not } x_3\}

- Constraint formulation for SAT
  - C_1 (\{x_1, x_2\},(0,0)(1,0)(1,1))
  - C_2 (\{x_2, x_3\},(0,0)(1,0)(0,1))
  - C_3 (\{x_3\},(0)) Unary constraint

- Ex: Compute dual graph

Ex: Consider the set of clauses:

- \{\text{not } C, \text{ A or B or C}, \text{ not } A \text{ or B or E}, \text{ not } B \text{ or C or D}\}

- Give CP formulation

- Give Primal and dual graph
Set operations with relations

Relations are subsets of the cartesian product of the variables in their scope

- S: x1,x2,x3
- R: {(a,b,c,)(c,b,a)(a,a,b)}

We can apply standard set-operations on relations

- Intersection
- Union
- Difference

Scope must be the same
## Selection, Projection and Join

<table>
<thead>
<tr>
<th></th>
<th>$R$</th>
<th>$R'$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$x_1$</td>
<td>$x_2$</td>
</tr>
<tr>
<td></td>
<td>b</td>
<td>b</td>
</tr>
<tr>
<td></td>
<td>c</td>
<td>b</td>
</tr>
<tr>
<td></td>
<td>c</td>
<td>n</td>
</tr>
</tbody>
</table>

\[
\sigma_{x_3=c}(R)
\]

\[
\pi_{x_2,x_3}(R)
\]

\[R \bowtie R'\]
Constraint Inference

Given R13 and R23

- $R13 = R23 = (R,Y)(Y,R)$

We can infer $R12$

- $R12 = (R,R)(Y,Y)$

Composition

$$\pi_{x_1, x_2}(R_{1,3} \bowtie R_{2,3})$$
R12 is redundant

- Every deduced constraint is redundant

Equivalence of Constraint Networks:

- Same set of variables
- Same set of solutions

Redundant Constraint

- RC constraint network
- RC’ = removing R* from RC
- If RC is equivalent to RC’ then R* is redundant
Relations vs Binary Networks

Can we represent every relation with binary constraint?

**No** (unfortunately)

- most relations cannot be represented by binary networks (i.e. graphs): $2^{kn}$

Given $n$ variables with domain size $k$

- # of relations (subsets of joint tuples)
- # of binary networks ($k^2$ tuples for each couple, $n^2$ couples at most)
Representing general relations: Projection network

Represent a general relation using a binary network:

- Project a relation onto each pair of its variables
  - \( R = \{(1,1,2)(1,2,2)(1,2,1)\} \)
  - \( P[R]: P_{12} = \{(1,1)(1,2)\} \quad P_{13} = \{(1,2)(1,1)\} \quad P_{23} = \{(1,2)(2,2)(2,1)\} \)

- \( \text{Sol}(P[R]) = \{(1,1,2)(1,2,2)(1,2,1)\} = R \)

Is it always the case?
Approximation with Projection Network

- \( R = \{(1,1,2)(1,2,2)(2,1,3)(2,2,2)\} \)

- \( P[R]: P12 = \{(1,1)(1,2)(2,1)(2,2)\} \)
  \( P23 = \{(1,2)(2,2)(2,3)\} \)
  \( P13 = \{(1,2)(2,3)(2,2)\} \)

- \( \text{Sol}(P[R]) = \{(1,1,2)(1,2,2)(2,1,2)(2,1,3)(2,2,2)\} \)

\( \text{Sol}(P[R]) \neq R \) but...

If \( N \) is a projection network of \( R \) this is always true
\[ R \subseteq \text{Sol}(P[R]) \]

The projection network \( N \) is the **tightest** upper bound for \( R \)
\[ \neg \exists R' \ R \subseteq R' \subseteq \text{Sol}(P[R]) \]
“Tighter than” and intersection for networks

- Given two binary networks, $N'$ and $N$, on the same set of variables, $N'$ is at least as tight as $N$ iff for each $i,j$ we have
  \[ R'_{i,j} \subseteq R_{i,j} \]
  
- $N'$ tighter than $N$ then $\text{Sol}(N')$ are included in $\text{Sol}(N)$
  \[ \text{Sol}(N') \subseteq \text{Sol}(N) \]

- The intersection of two network is the pair-wise intersection of their constraints
  \[ N' \cap N \Rightarrow \forall i, j \quad R'_{i,j} \cap R_{i,j} \]

- If $N$ and $N'$ are two equivalent networks then $\cap$ intersection $N'$ is as tight as both and equivalent to both
Minimal network

The minimal network is obtained intersecting all equivalent networks

The minimal network is identical to the projection network of its solutions

\[
\{N_1, \cdots, N_l\} \text{ equivalent to } N_0
\]

\[
M(N_0) = \bigcap_{i=1}^{l} N_i
\]

\[
\rho = \text{Sol}(N) \quad M(\rho) = P[\rho]
\]
Minimal Network and explicit constraints

The minimal network is perfectly explicit for every constraints (unary, binary)

- a couple (value) appears in at least one binary (unary) constraint
- the couple (value) will appear in at least one solution
- Ex: find minimal network for 4-queen problem

Finding a solution for minimal network is still hard.
Binary Decomposable Network

Given minimal network:

- Easy to find a couple which is part of a solution
- **Not easy** to extend partial solutions
- Ex: minimal network for 4-queen problem

Binary decomposable network

- Every projection is expressible by a binary network
- For a binary decomposable network $N(X,D,C)$, $P[N]$ expresses $Sol(N)$ and all $Sol(S)$ where $S \subseteq X$
- Ex: $r = \{(a,a,a,a)(a,b,b,b)(b,b,a,c)\}$ binary dec.?