Constraint Networks

Dario Maggi

University Basel

October 9, 2014

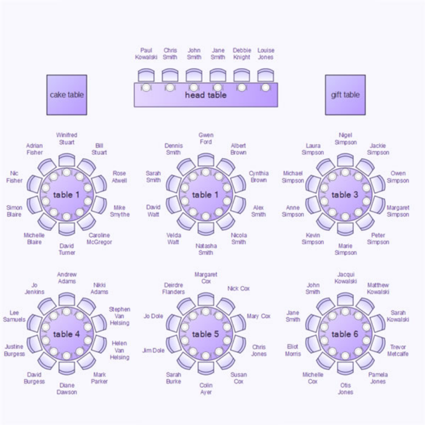
Overview

What is a Constraint Network?

- Groceries Shopping
- Daily routine
- Seat arrangement at a wedding
- Transportation scheduling
- Factory scheduling

Seat arrangement at a wedding

Table Layout



Constraints:

- Bride and groom sit at the "head table"
- Bride and groom sit next to each other
- Parents of the bride and groom sit close to the married couple, but not too close
- Beside every woman sits a man.
- There needs to be a children's table.
- The children's table must not be close to the gifts table.
- ...

What is a constraint network?

- Variables
  (Positions in a bag, slots in a schedule, seats at a wedding, ...)
- Possible values for the variables.
  ((Milk, Bread, Egs), (shower, training, work, homework, eat), (Adam, Beatrice, Carla, ...))
- Constraints:
  "You must not put the milk on top of the eggs."
  "You should take a shower after the training."
  ...

Definition: Constraint Network

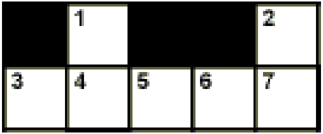
A constraint network $\mathcal{R}$ is given by a triple $\mathcal{R} = (X, D, C)$, where:

- $X = \{x_1, \ldots, x_n\}$ is a finite set of *variables*,
- $D = \{D_1, \ldots, D_n\}$ is the set of *domains*, where each D_i is a finite set that contains the possible values for variable x_i, and
- $C = \{C_1, \ldots, C_k\}$ is a finite set of *constraints*.

Each constraint $C_i \in C$ is given by a tuple $C_i = (S_i, R_i)$. S_i is called the *scope*. The scope provides the variables over which the relation is defined. The scope therefore needs to be a subset of X. The *relation* R_i is a set of tuples. Each of these tuples holds an allowed assignment for the variables of its scope.

- Constraint can be written as its Relation, if the scope is indexed and clear:
  $C_i = \big(\{x, y\}, \{(1,1), (1,2)\}\big)$
  $C_i = R_{xy} = \{(1,1), (1,2)\}$
- Network Scheme: Set of all scopes $S = \{S_1, \ldots, S_k\}$
- Arity of a constraint: $|S_i|$
- unary constraint: arity $= 1$
- binary constraint: arity $= 2$
- binary constraint network: only unary and binary constraints.

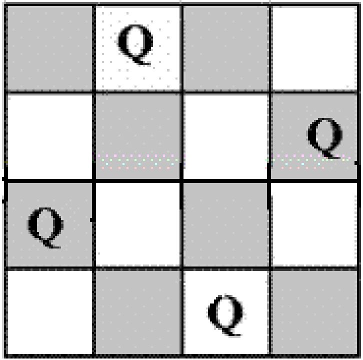
Crossword Puzzle



Constraint network $\mathcal{R} = (X, D, C)$:

- Cells as variables: $X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$.
- Domains are the letters of the alphabet: $D = \{D_i = \{A, \ldots, Z\} \mid 1 \leq i \leq 7\}$.
- Constraints:
  $R_{1,4} = \{(M, E), (H, I), (B, E), (D, O)\}$
  $R_{2,7} = \{(M, E), (H, I), (B, E), (D, O)\}$
  $R_{3,4,5,6,7} = \{(H, E, L, L, O), (T, H, E, R, E), (D, O, I, N, G), (B, E, N, N, I)\}$

4-Queens Problem



Constraint network $\mathcal{R} = (X, D, C)$:

- Columns as variables:

$$X = \{x_1, x_2, x_3, x_4\}$$

- Domains are the rows the queen stands in:

$$D = \{D_i = \{1, 2, 3, 4\} \mid 1 \leq i \leq 4\}$$

- Constraints:

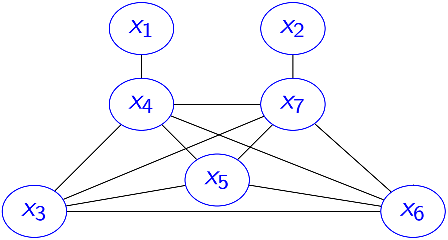
$$R_{ij} = \{(v_i, v_j) \mid v_i \neq v_j, |v_i - v_j| \neq |i - j| \text{ for } 1 \leq i < j \leq 4\}$$

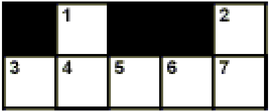
Constraint Graphs

Constraint Graphs

Primal Constraint Graph

Primal constraint graph of the crossword puzzle.

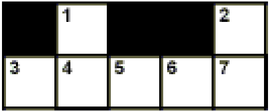


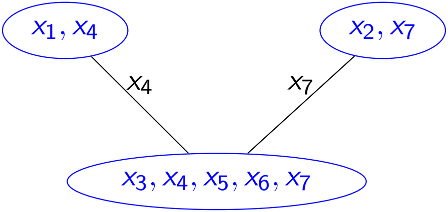


- Variables are vertices/nodes.
- Variables which share a common scope are connected with edges.

Dual Constraint Graph

Dual constraint graph of the crossword puzzle.





- Relations are vertices/nodes.
- Edges connect relations sharing a common variable.
- Edge is labeled with the shared variables.

Solutions of Constraint Networks

Solutions of Constraint Networks

Instantiation

If a variable x_i gets assigned value the variable has been instantiated.

Notations:

- $\bar{a} = \big((x_i, a_i), \ldots, (x_k, a_k)\big)$
- $\bar{a} = \big(x_i = a_i, \ldots, x_k = a_k\big)$
- $\bar{a} = \big(a_1, a_2, \ldots\big)$

Satisfying a Constraint

An instantiation $\bar{a}$ satisfies a constraint $C_i = (S_i, R_i)$ if:

- Every variable in the scope S_i is assigned by $\bar{a}$.
- There must be a tuple in R_i that corresponds to the values of $\bar{a}$ on the variables in S_i.

Satisfying a Constraint: Example

$\bar{a} = ((x, 1), (y, 2), (z, 3))$
$R^1_{xy} = \{(1, 2)\} \longleftarrow \bar{a}$ satisfies R^1_{xy}
$R^2_{xy} = \{(1, 3)\} \longleftarrow \bar{a}$ does not satisfy R^2_{xy} (y cannot take the value 2)
$R^3_{xy} = \{(1, 1), (1, 2), (2, 4)\} \longleftarrow \bar{a}$ satisfies R^3_{xy}
$R^4_{xy} = \{(1, 1), (1, 3), (2, 4)\} \longleftarrow \bar{a}$ does not satisfy R^4_{xy}
$R^4_{ux} \longleftarrow \bar{a}$ does not satisfy R^4_{ux} ($\bar{a}$ does not assign a value to u).

A Consistent Partial Instantiation

A partial instantiation is consistent if it satisfies all constraints which scopes are covered by $\bar{a}$

$\bar{a} = ((x, 1), (y, 2), (z, 3))$
$\bar{a}$ has to satisfy every constraint where: $S_i \subseteq \{x, y, z\}$
R_{xy}, R_{xz}, R_{xyz} have to be satisfied.
R_{wxy}, R_{axy} do not have to be satisfied.

Solution

An instantiation is a solution of a constraint network if all variables are instantiated and the instantiation is consistent.

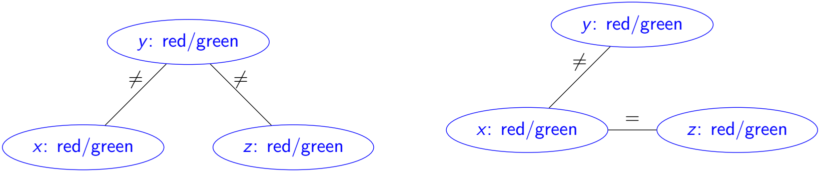
Solution($\mathcal{R}$) is the set of all complete consistent instantiations.

Solution($\mathcal{R}$) can also be interpreted as a relation ρ_X

We say that $\mathcal{R}$ expresses relation ρ_X.

Equivalence of Constraint Networks

Two networks are equivalent if they are defined over the same variables and express the same solutions.



Deduction with Constraints

The goal is to infer or deduct additional constraints.
The network has to stay equivalent.
Constraint deduction can be accomplished through composition:

Composition

$$R_{xy} \cdot R_{yz} = R_{xz} = \{(a,c) \mid a \in D_x, c \in D_z, \exists b \in D_y \text{ such that } (a,b) \in R_{xy} \text{ and } (b,c) \in R_{yz}\}$$

$$R_{xy} \cdot R_{yz} = R_{xz} = \pi_{\{x,z\}}(R_{xy} \bowtie R_{yz})$$

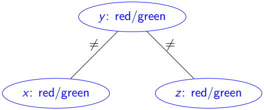
Composition Example

R_{xy}:

x	y
red	green
green	red

R_{yz}:

y	z
green	red
red	green



The natural join $R_{xy} \bowtie R_{yz}$:

x	y	z
red	green	red
green	red	green

The projection on $\{x, z\}$, R_{xz}:

x	z
green	green
red	red

Properties of Binary Constraint Networks

Properties of Binary Constraint Networks

Expressive Power of Binary Networks

We want to get a feeling of the expressive powers of binary networks.
Can any relation be represented as a binary network?
For this to be the case, every relation has to be representable by a binary network.

Expressive Power of Binary Networks II

Given:

- n variables
- each variable can have k different values

Number of possible relations:

2^{k^n}

Number of possible binary networks:

$2^{k^2 \frac{n(n-1)}{2}}$

The Binary Projection Network

The Binary Projection Network

Given a relation ρ defined over $X = \{x_1, \ldots, x_n\}$, the binary projection network $P(\rho)$ on **each possible pair** of its variables, is given as $\mathcal{P}(\rho) = (X, D, P)$:

- $D = \{D_i\}$ with $D_i = \pi_{x_i}(\rho)$ for $1 \leq i \leq n$
- $P = \{P_{ij}\}$ with $Pij = \pi_{x_i x_j}(\rho)$ for $1 \leq i < j \leq n$

The projection network P(ρ):

ρ_{xyz}:

x	y	z
1	1	2
1	2	2
1	2	1

P_{xy}:

x	y
1	1
1	2

P_{xz}:

x	z
1	2
1	1

P_{yz}:

y	z
1	2
2	2
2	1

$sol(P(\rho))$:

x	y	z
1	1	2
1	2	2
1	2	1

The Binary Projection Network II

The projection network P(ρ):

ρ_{xyz}:

x	y	z
1	1	2
1	2	2
2	1	3
2	2	2

P_{xy}:

x	y
1	1
1	2
2	1
2	2

P_{xz}:

x	z
1	2
2	3
2	2

P_{yz}:

y	z
1	2
2	2
1	3

$sol\big(P(\rho)\big)$:

x	y	z
1	1	2
1	2	2
2	1	2
2	1	3
2	2	2

Not every relation can be expressed by a binary network.

The binary projection network is an upper bound network approximation.

"Tighter than"

As least as tight

Consider constraint networks $\mathcal{R}$ and $\mathcal{R}'$. $\mathcal{R}$ is at least as tight as $\mathcal{R}'$ if for every relation R_{ij} of $\mathcal{R}$ i holds that $R_{ij} \subseteq R'_{ij}$. R'_{ij} is the corresponding relation in $\mathcal{R}'$.

$\mathcal{R}$ with only one relation

x	y
1	1
1	2
2	2

$\mathcal{R}'$ with only one relation

x	y
1	1
1	2
2	1
2	2
2	3

Intersection of Binary Networks

Intersection of $\mathcal{R}$ and $\mathcal{R}'$

The intersection $\mathcal{R} \cap \mathcal{R}'$ of two networks $\mathcal{R}$ and $\mathcal{R}'$ is the network obtained by pairwise intersection of the corresponding constraints.

Intersection of Binary Networks II

Intersection of two equivalent networks

The intersection of two equivalent networks produces a network equivalent to both. The produced network is at least as tight as both.

Intersection of Binary Networks III

R_{xy}

x	y
red	green
green	red

R_{yz}

y	z
red	green
green	red

R_{xz}

x	z
red	green
green	red
green	green
red	red

R'_{xy}

x	y
red	green
green	red

R'_{yz}

y	z
red	green
green	red
green	green
red	red

R'_{xz}

x	z
green	green
red	red

Intersection of Binary Networks IV

$R_{xy} \cap R'_{xy}$

x	y
red	green
green	red

$R_{yz} \cap R'_{yz}$

y	z
red	green
green	red

$R_{xz} \cap R'_{xz}$

x	z
green	green
red	red

Minimal Constraint Network

Minimal Constraint Network

Let $\{\mathcal{R}_1, \ldots, \mathcal{R}_l\}$ be the set of all networks equivalent to $\mathcal{R}_0$ and let $\rho = sol(\mathcal{R}_0)$. Then the minimal network $\mathcal{M}$ of $\mathcal{R}_0$ or ρ is defined by $\mathcal{M}(\mathcal{R}_0) = \mathcal{M}(\rho) = \bigcap_{1 \leq i \leq n} \mathcal{R}_i$.

Every tuple in a relation of a minimal network is part of a solution.

If a relation is representable by a binary projection network the binary projection network is minimal.

Minimal Constraint Network II

This relation is representable by its binary projection network

w	x	y	z
1	1	1	1
1	2	2	2
2	2	1	3

Representable by binary projection network

If a binary network is minimal then every tuple in its relations can be extended into a solution.

BUT:

We cannot just take a tuple of a minimal network and extend the tuple with another tuple and expect the consistent instantiation to have a solution.

Binary-Decomposable Relation

A relation is Binary-Decomposable:

- the relation is equivalent to its binary projection network.
- each of its possible projected relations is binary-decomposable.

If a relation is decomposable it is simple to extend consistent instantiations into another consistent instantiation which is also part of a solution.

Binary-Decomposable Networks II

M_{x_1,x_2}

x_1	x_2
2	4
3	1

M_{x_1,x_3}

x_1	x_3
2	1
3	4

M_{x_1,x_4}

x_1	x_4
2	3
3	2

M_{x_2,x_3}

x_2	x_3
1	4
4	1

M_{x_2,x_4}

x_2	x_4
1	2
4	3

M_{x_3,x_4}

x_3	x_4
1	3
4	2

References

Rina Dechter (2003)
Constraint Processing
The Morgan Kaufmann Series in Artificial Intelligence.
Elsevier Science

Questions?