Constraint Networks basic concepts

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

Intro

- Motivations
- Applications
- Examples

Motivations: Combinatorial Problems

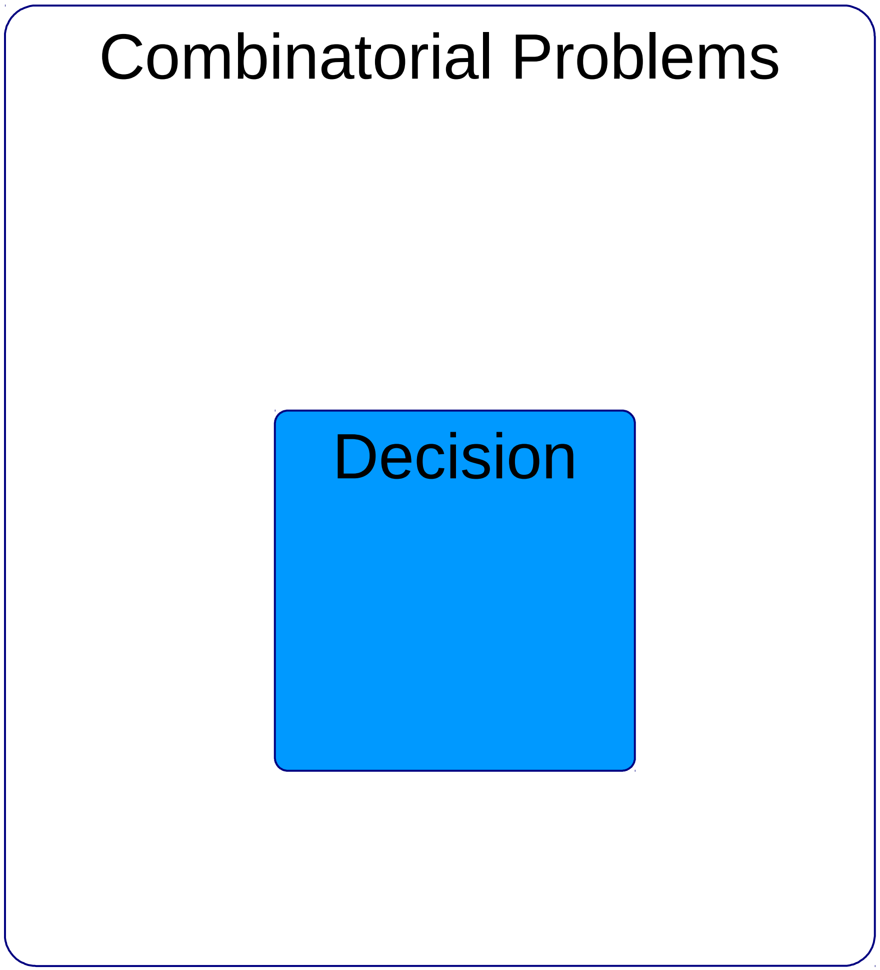
Combinatorial Problems

Given a set of possible solutions find the best one

Main issue:

Space of possible solutions is huge (exponential) hence complete search of all solutions is impossible

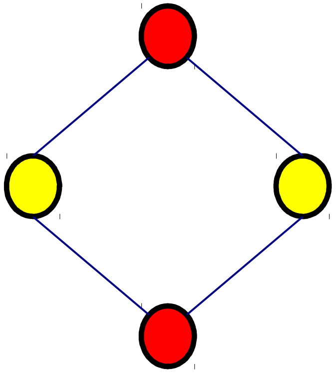
Combinatorial Problems: Decision



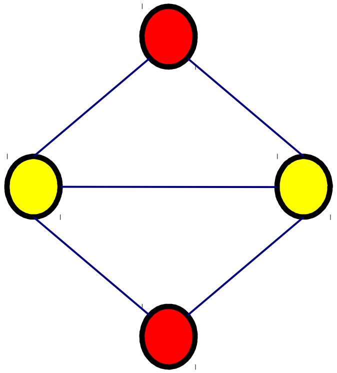
Graph Colouring

Given a graph and k colours colour each node such that no two adjacent nodes have the same colour

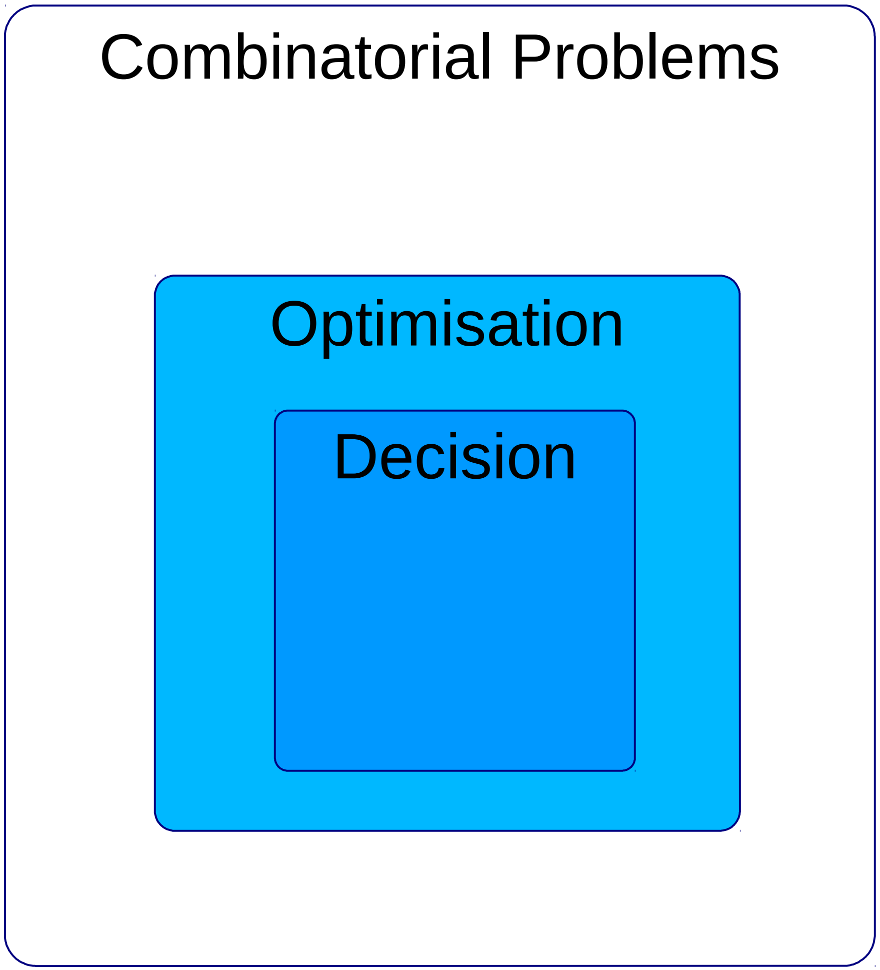
Yes



No



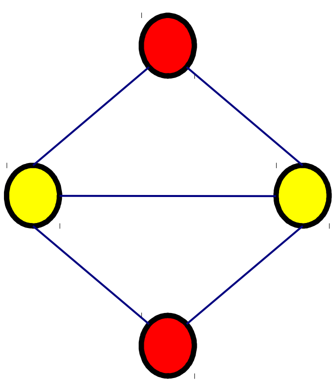
Combinatorial Problems: Optimisation



<u>Graph Colouring</u> (optimisation)

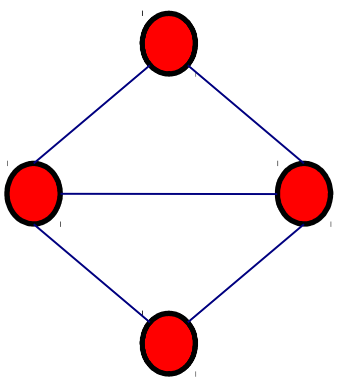
Given a graph and k colours colour each node such that the minimum number of two adjacent nodes have the same colour

No

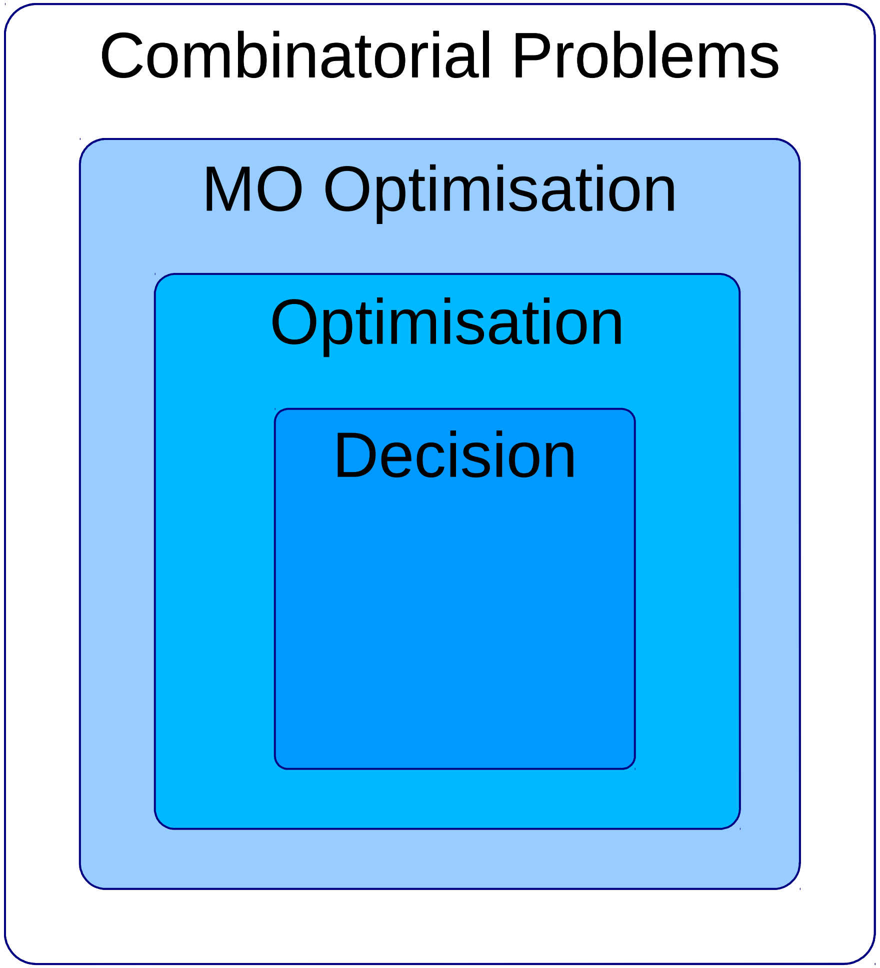


Best

No



Combinatorial Problems: Multi-Objective Opt.



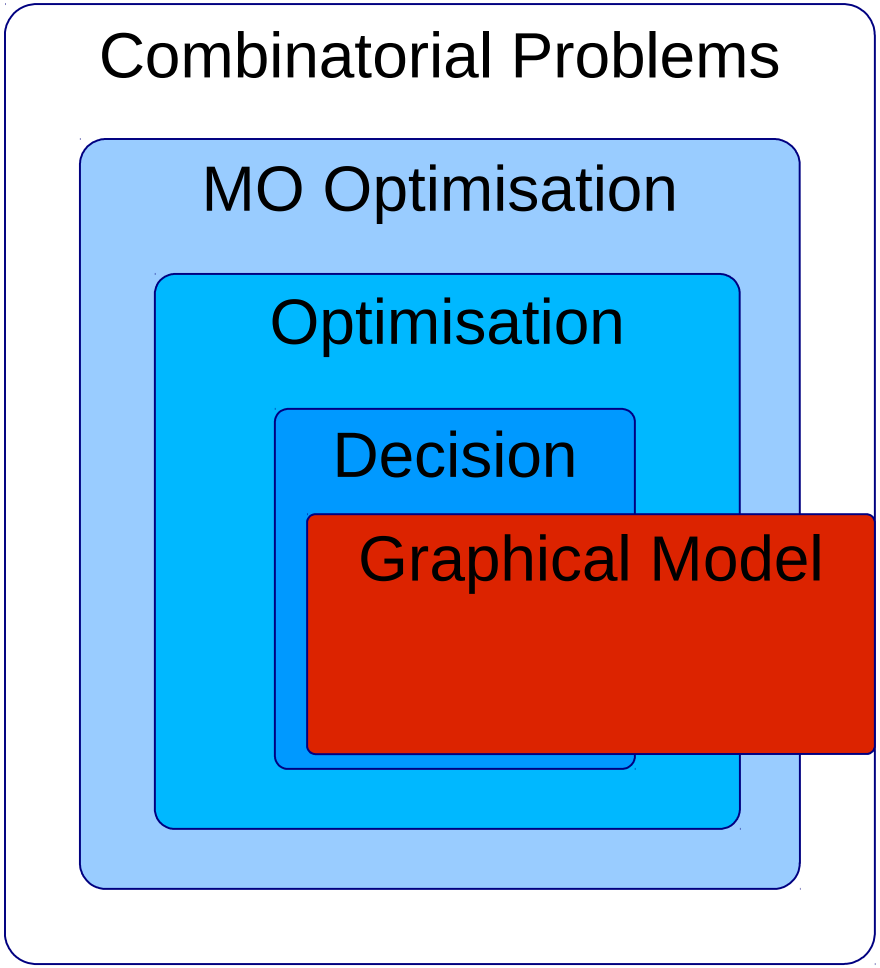
Portfolio investment

Given a set of investments
Find a subset of them
(portfolio)

Such that:

Minimise Risks
Maximise Profits

Combinatorial Problems: Graphical Models



Characteristics:

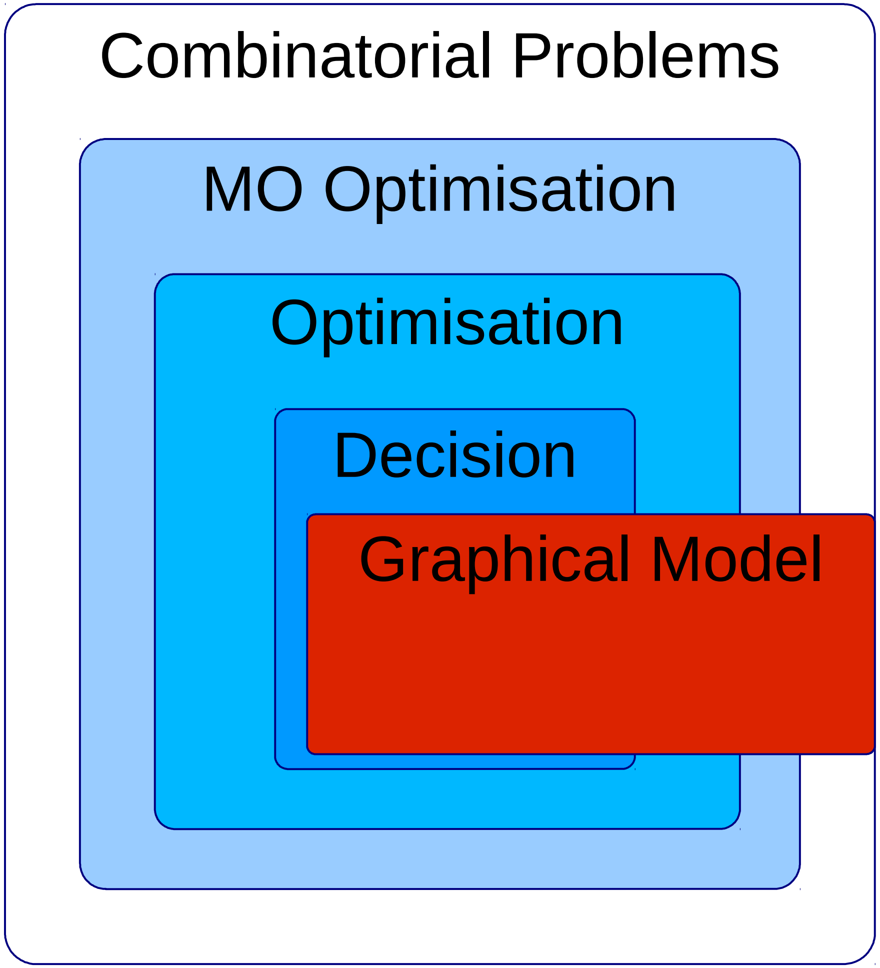
1) A set of Variables

2) A set of Domains, one for each variable

3) A set of Local functions

Global function is an aggregation of local function

Combinatorial Problems: Graphical Models II



Graph Colouring

- Local functions: number of conflicts for each link

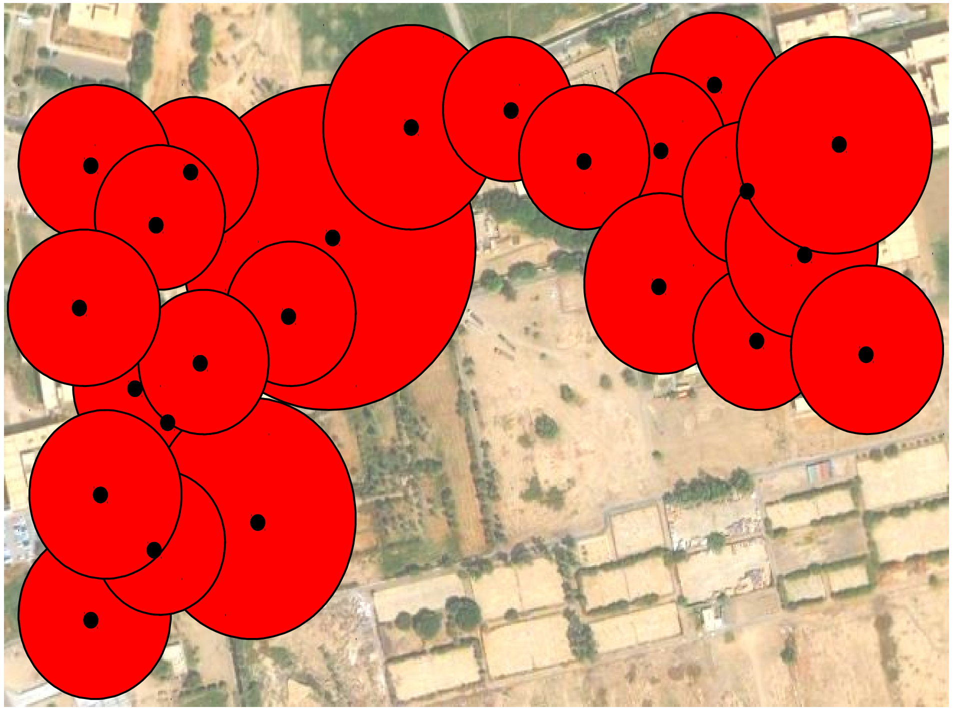
- Global function: sum of local functions

GMs: Exploit problem structure
Efficient, General
Used in many fields:

Constraint Reasoning
Bayesian Network
Error Correcting codes
...

Application: Wide AreaSurveillance

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



<u>Features:</u>

1) Energy Harvesting
2) Energy Neutral Operations
3) Sense/sleep modes

Assumptions:

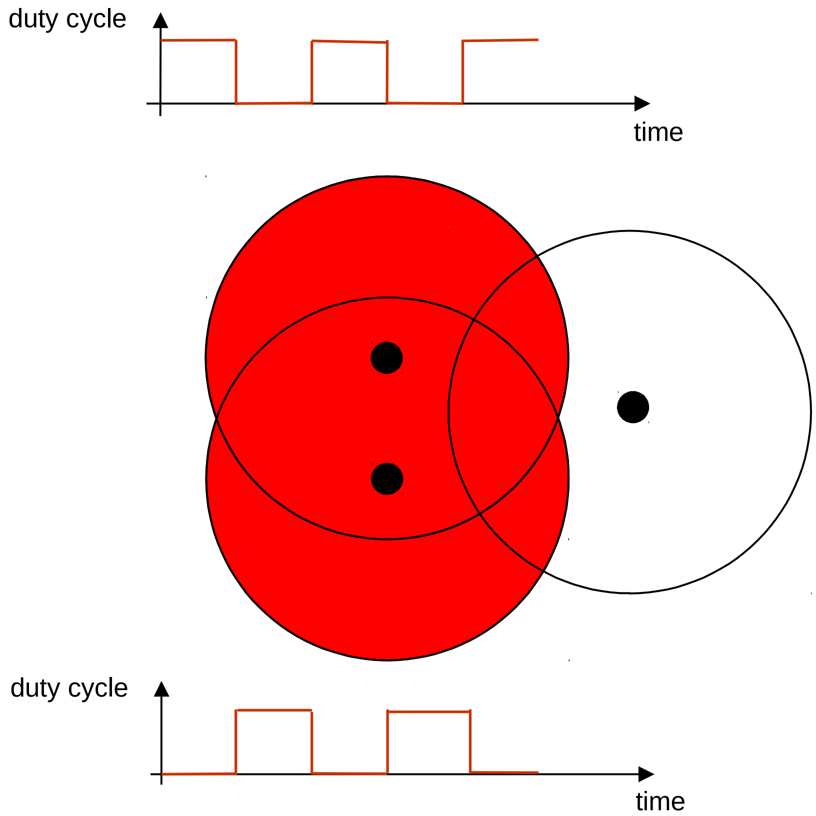
1) Activity can be detected by single sensor
2) Neighbors (i.e., overlapping sensors) can communicate
3) Only Neighbors are aware of each other

The Coordination Problem

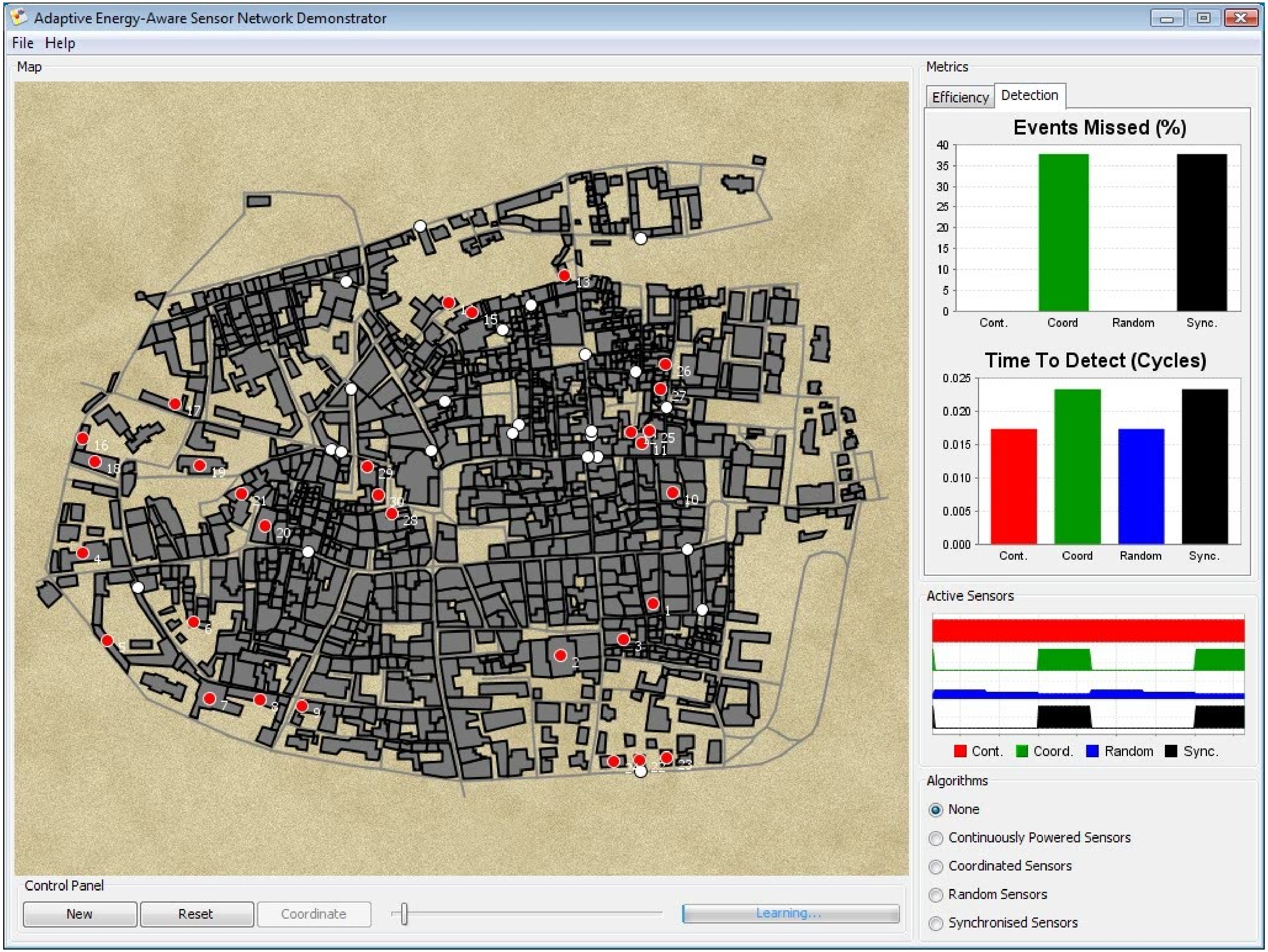
Energy neutral operation: Constraints on sense/sleep schedules

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

Similar to Graph Coloring but:
   Overlapping Areas -> weights
   Non binary relationships



W. A. S. Demo



Adaptive Energy-Aware Sensor Network Demonstrator
File Help
Map
Metrics
Efficiency
Detection
Events Missed (%)
Cont.
Coord
Random
Sync.
Time To Detect (Cycles)
Active Sensors
Cont. Coord. Random Sync.
Algorithms
None
Continuously Powered Sensors
Coordinated Sensors
Random Sensors
Synchronised Sensors
Control Panel
New
Reset
Coordinate
Learning...

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 Programing)
Information Theory (error correcting codes)
Agents and Multi Agent Systems (coordination)

CP: what can we express with constraints

<u>All problems that can be formulated as follow:</u>

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

<u>Graph colouring:</u>

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

<u>N-Queens problem:</u>

- 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: x1,y1,x2,y2,x3,y3,x4,y4

– No two queens on same row: x1 != x2, x1 != x3, ...

– No two queens on same column: y1 != y2, y1 != y3, ...

– No two queens on same diagonal: |x1-x2| != |y1–y2| ...

X2 = 2 y2=1

X1 = 1 y1=2

	Q2		
Q1			Q4
		Q3	

4-Queens alternative formulation

A (better) Formulation

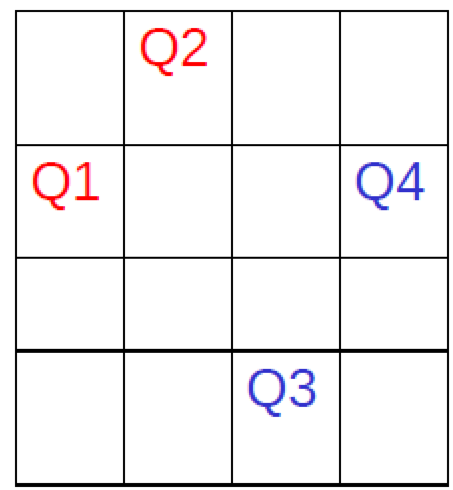
In every valid solution one column for each queen

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

Constraints:

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

r1 = 2 r2 = 1



Formalization and Representation

Formal Definition

Representing Constraint Networks

Examples

Constraints encode information

Constraint as information:

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

We can exploit constraints 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 = {x1,...,xn}
D set of domains D = {D1,...,Dn} Di = {v1,...,vk(i)}
C set of constraints (Si,Ri) [Si ⊆ X]

  scope: variables involved in Ri
  Ri subset of cartesian product of variables in Si
  Ri expresses allowed tuples over Si

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,...,4]

	r1	r2	r3	r4
1	Q	no		
2		no		
3		ok		
4		ok		

C1 = (S1,R1)
S1 = {r1,r2}
R1 = {(1,3)(1,4)(2,4)(3,1)(4,1)(4,2)} = R1-2
...
C4 = (S4,R4)
S4 = {r2,r3}
R4 = {(1,3)(1,4)(2,4)(3,1)(4,1)(4,2)} = R2-3
...

Solution and partial consistent solutions

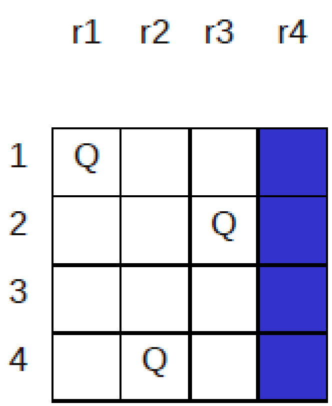
Partial Solution

– Assignment of a subset of variables

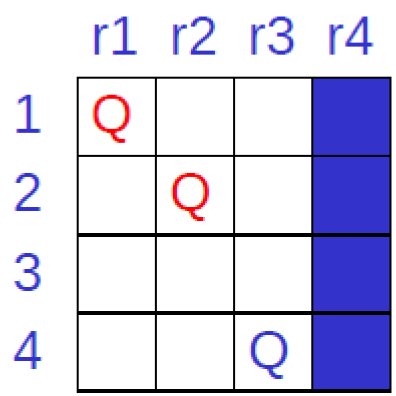
Consistent partial solution:

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

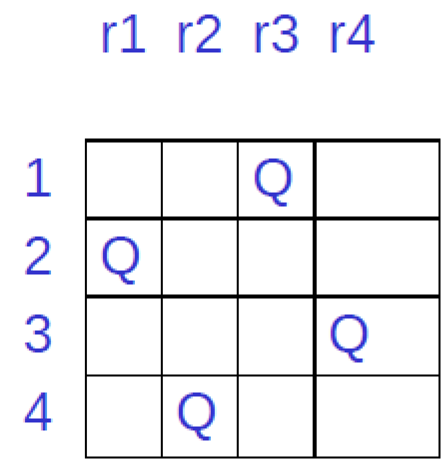
– A consistent partial solution may not be a subset of a solution



consistent



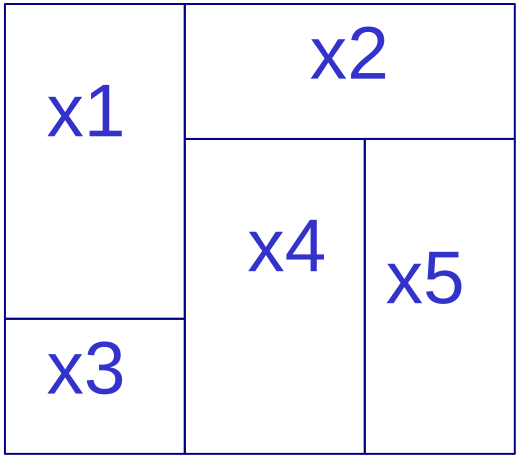
inconsistent

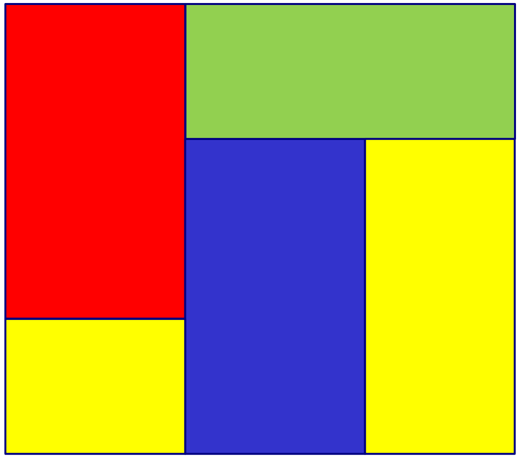


solution

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





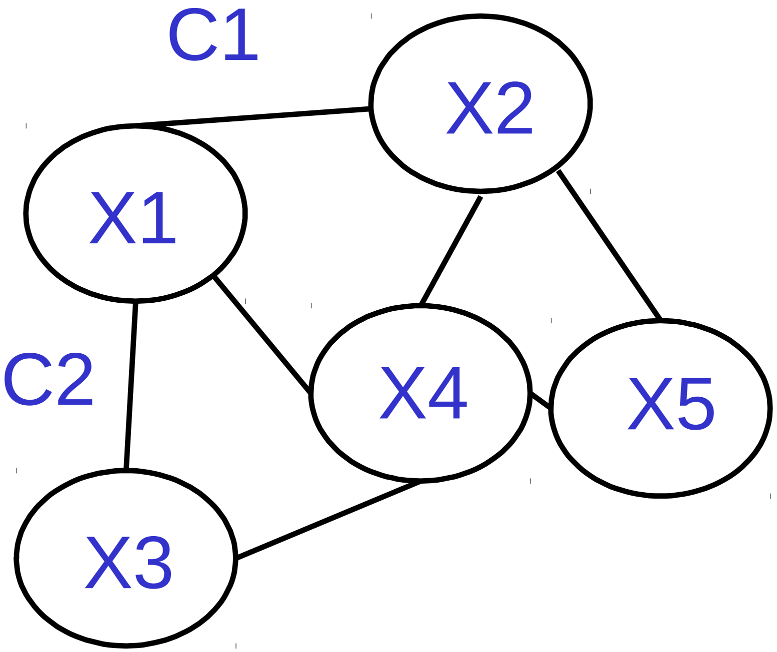
C1 = ({x1,x2}, x1 != x2)

C2 = ({x1,x3}, x1 != x3)

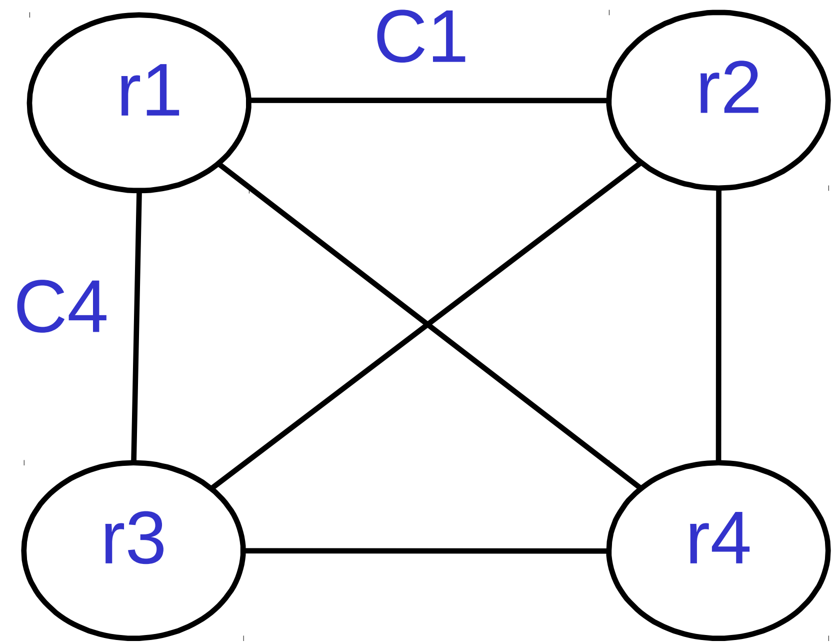
...

Solution

Constraint Network



Map Colouring

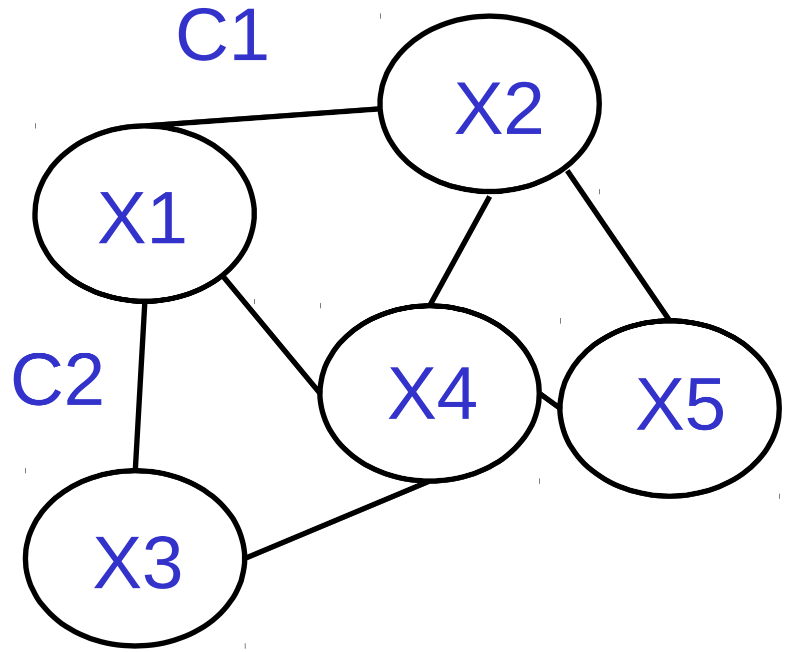


4-Queens

Constraint Graph

<u>Primal graph</u>

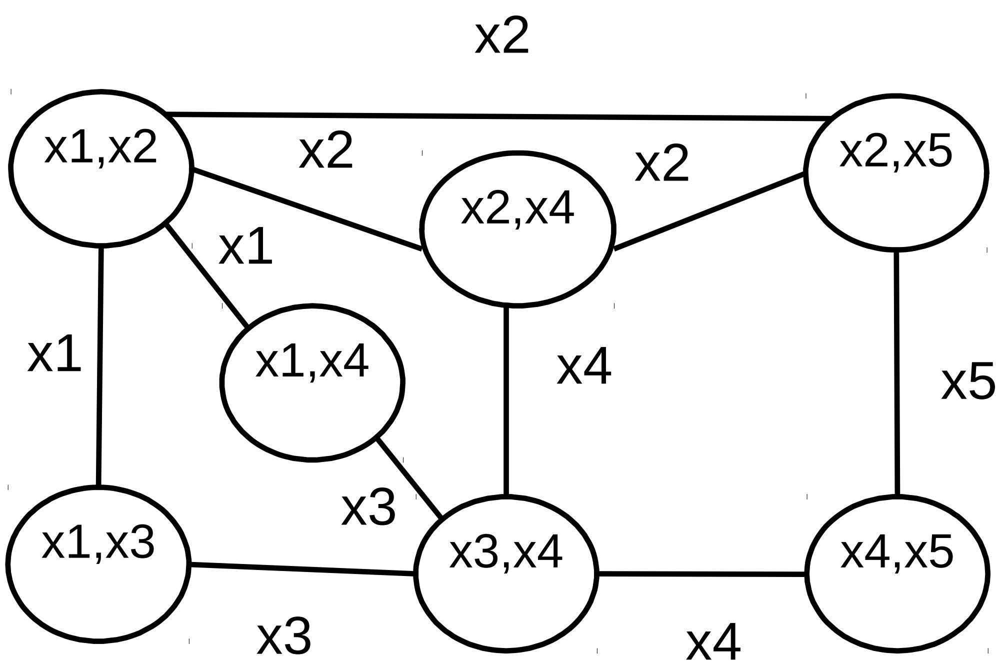
– Node: variable
– Arc: constraint holding between variables



Map Colouring

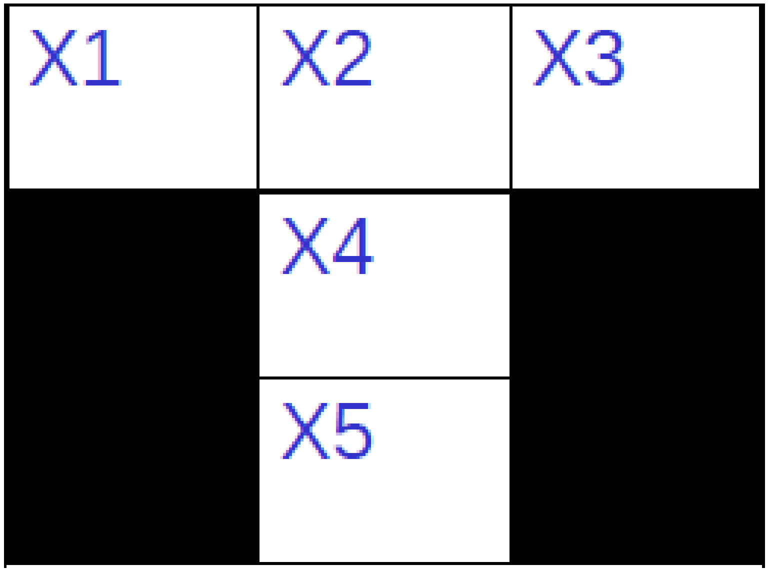
Dual Graph

Nodes: constraints' scopes
Arcs: shared variables

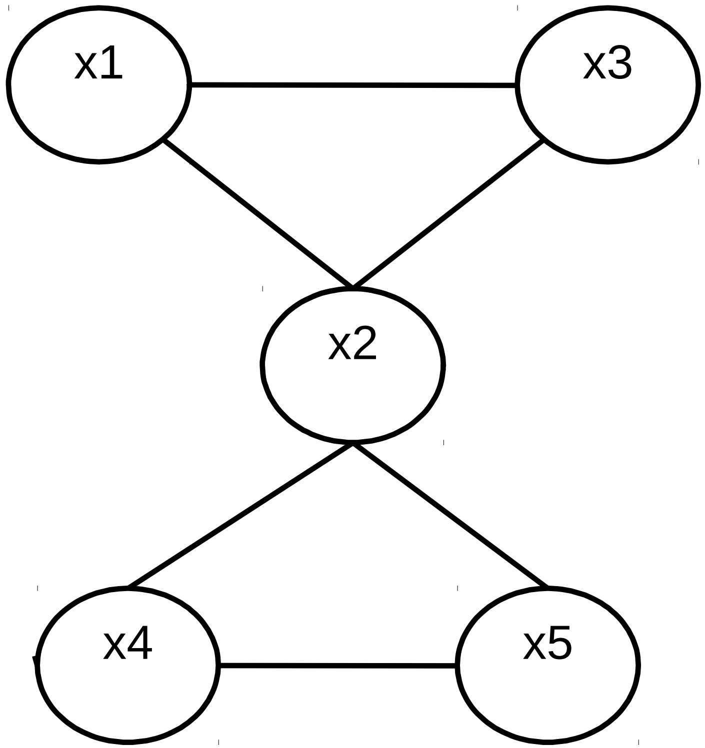


Map Colouring

Crossword Puzzle: Primal graph



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



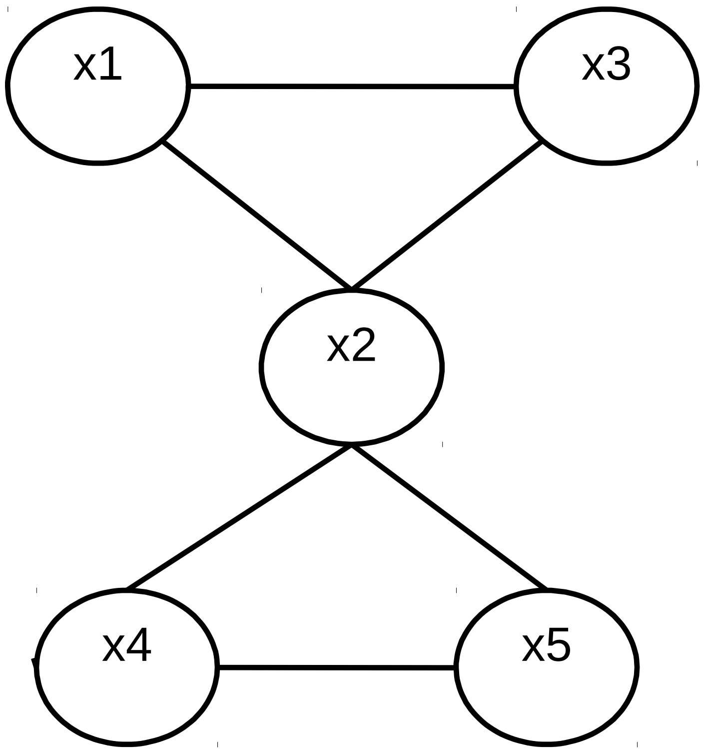
Di : letters of the alphabet
C1 [{x1,x2,x3},(MAP)(ARC)]
C2 [{x2,x4,x5},(MAP)(ARC)]

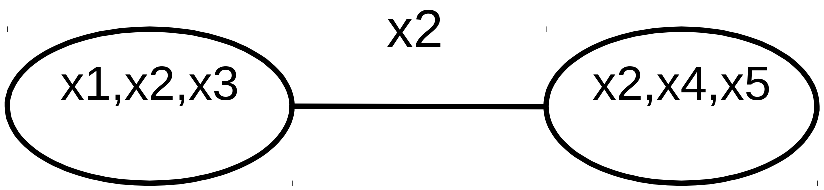
Crossword Puzzle: dual graph

Di : letters of the alphabet

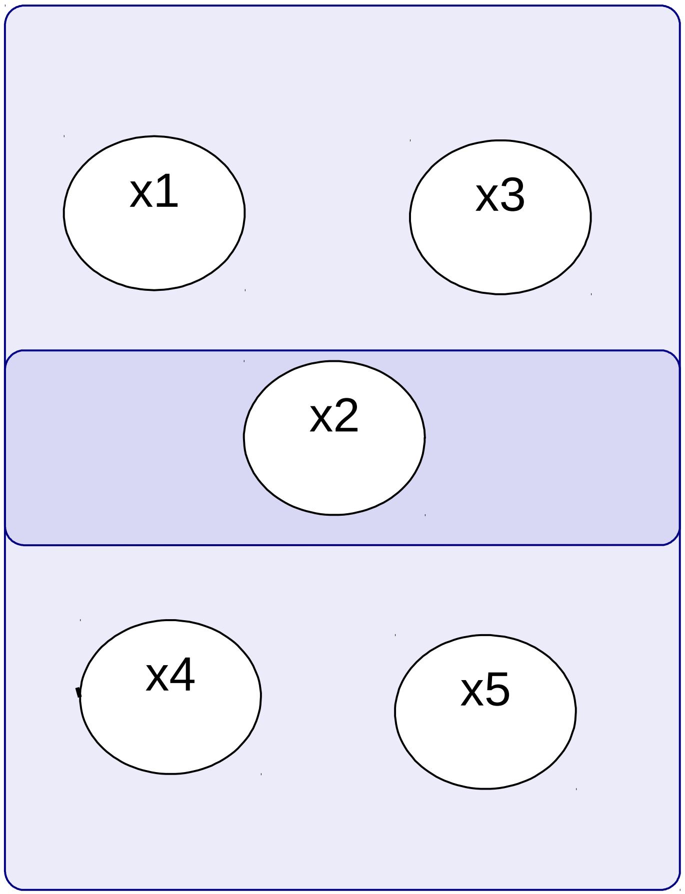
C1 [{x1,x2,x3},(MAP)(ARC)]

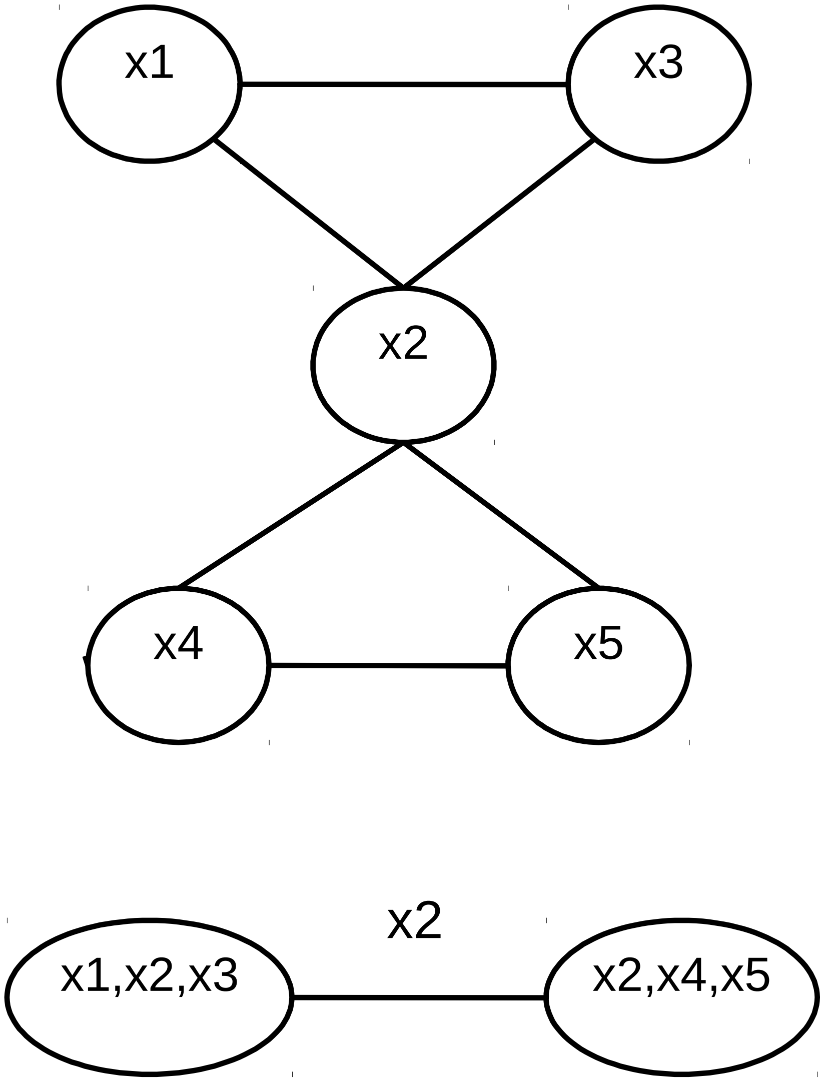
C2 [{x2,x4,x5},(MAP)(ARC)]





Hypergraphs and Dual Graphs





Hypergraph 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 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
  - *X1 != X2 in the n-queen problem*

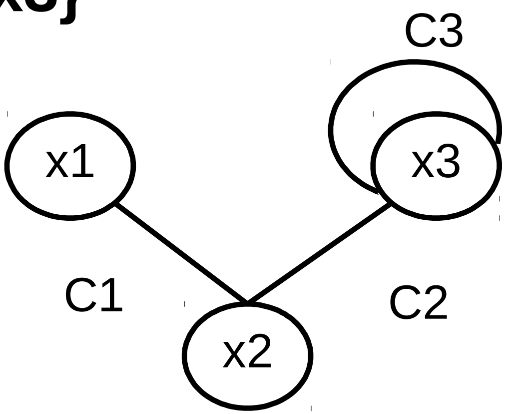
Propositional formula

- Boolean values of variables
  - *Boolean values that satisfy the formula*
  - **(a or b) = {(0,1)(1,0)(1,1)}**

Propositional CNF

Consider the set of clauses:

- **{x1 or not x2, not x2 or not x3, not x3}**
- Constraint formulation for SAT
  - C1 ({x1,x2},(0,0)(1,0)(1,1))
  - C2 ({x2,x3},(0,0)(1,0)(0,1))
  - C3 ({x3},(0)) Unary constraint
- Ex: Compute dual graph



Ex: Consider the set of clauses:

- **{not C, A or B or C, not A or B or E, not B 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

R

x_1	x_2	x_3
b	b	c
c	b	c
c	n	n

R'

x_2	x_3	x_4
a	a	1
b	c	2
b	c	3

$\sigma_{x_3=c}(R)$

x_1	x_2	x_3
b	b	c
c	b	c

$\pi_{x_2,x_3}(R)$

x_2	x_3
b	c
n	n

$R \bowtie R'$

x_1	x_2	x_3	x_4
b	b	c	2
b	b	c	3
c	b	c	2
c	b	c	3

Binary Constraint Network

- Constraint Inference
- Projection Network
- Minimal Network
- Binary Decomposable Networks

Binary constraint Network

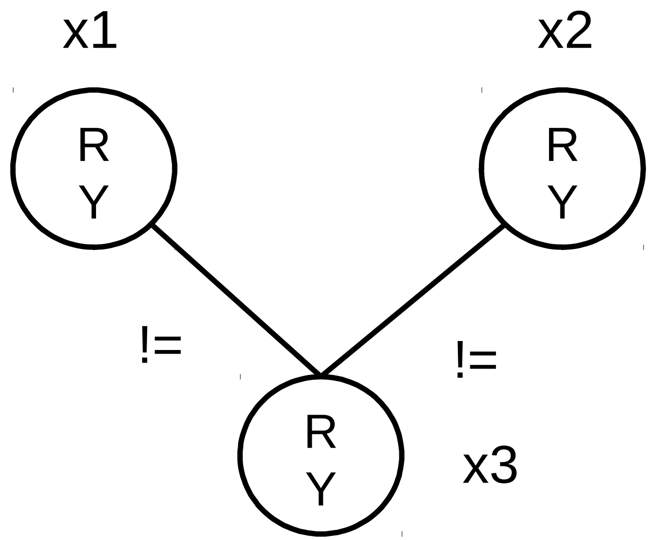
Constraint Inference

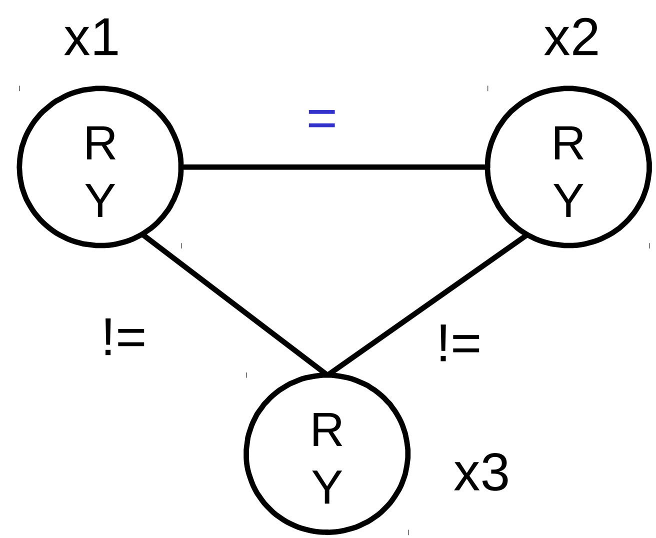
Given R13 and R23
- R13=R23 = (R,Y)(Y,R)

We can infer R12
- R12 = (R,R)(Y,Y)

Composition

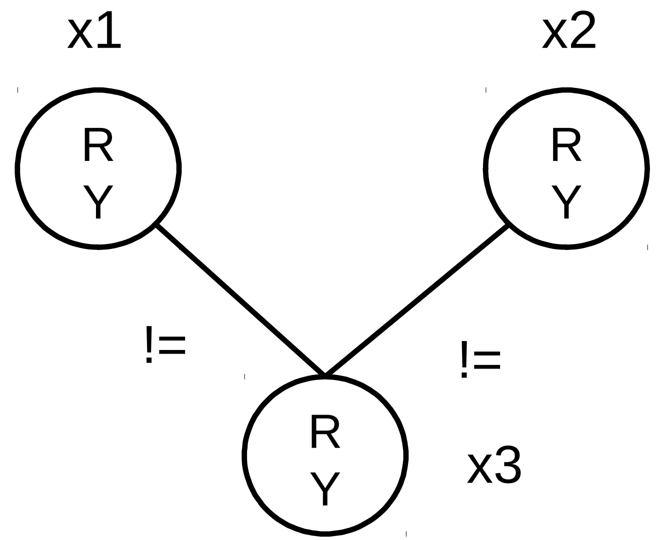
$$\pi_{x_1,x_2}(R_{1,3} \bowtie R_{2,3})$$





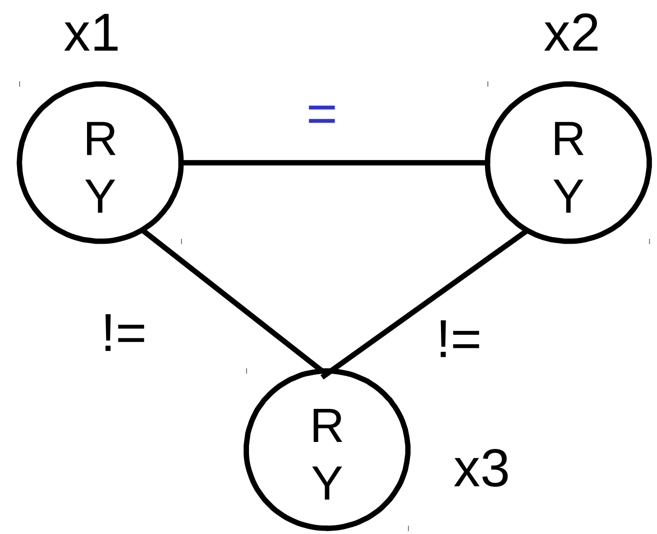
Binary constraint Network

R12 is redundant
- Every deduced constraint is



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):

Given n variables with domain size k

– # of relations (subsets of joint tuples) 2^{k^n}

– # of binary networks (k^2 tuples for each couple, n^2 couples at most) $2^{k^2 n^2}$

Representing general rel. : 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]: P12 = {(1,1)(1,2)}*
    P13 = {(1,2)(1,1)}
    P23 = {(1,2)(2,2)(2,1)}
- 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)}
- *Sol(P[R]) = {**(1,1,2)(1,2,2)**<u>(2,1,2)</u>**(2,1,3)**<u>(2,2,2)</u>}*

Sol(P[R]) != R but...

If N is a projection network of R this is always true

$$R \subseteq Sol(P[R])$$

The projection network N is the tightest upper bound for R

$$\neg\exists R' \; R \subseteq R' \subset Sol(P[R])$$

“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 Sol(N’) are included in Sol(N)

$$Sol(N') \subseteq Sol(N)$$

– The intersection of two network is the pair-wise intersection of their constraints

$$N' \cap N \Rightarrow \forall i,j \; R'_{i,j} \cap R_{i,j}$$

– If N and N’ are two equivalent networks then N 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

$$\{N_1, \cdots, N_l\} \text{ equivalent to } N_0$$

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

$$\rho = Sol(N) \; M(\rho) = P[\rho]$$

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 ⊆ X
- Ex: r = {(a,a,a,a)(a,b,b,b)(b,b,a,c)} binary dec. ?

Summary

Constraint Networks --> efficient way of representing and solving combinatorial problem

CN have several representations

- Structure: primal, dual graph
- Constraints: logic, arithmetic, tables

Binary Network --> special types of CN
Can not represent all relations but can approximate them (projection network)

Easy Network are the Binary Decomposable ones