

Constraint Programming

the user states the problem, the computer solve it

Nadjib Lazaar

Ing - Phd - Assistant Professor - University of Montpellier - COCONUT Team
<http://www.lirmm.fr/~lazaar/>

27/09/2021

Artificial Intelligence

Received ideas

AI is 100% artificial

► AI needs our help

AI would overall surpass human intelligence

► Living being is far from a machine being

AI is dangerous

► Not about AI, but about data sovereignty

An algorithm = an IA

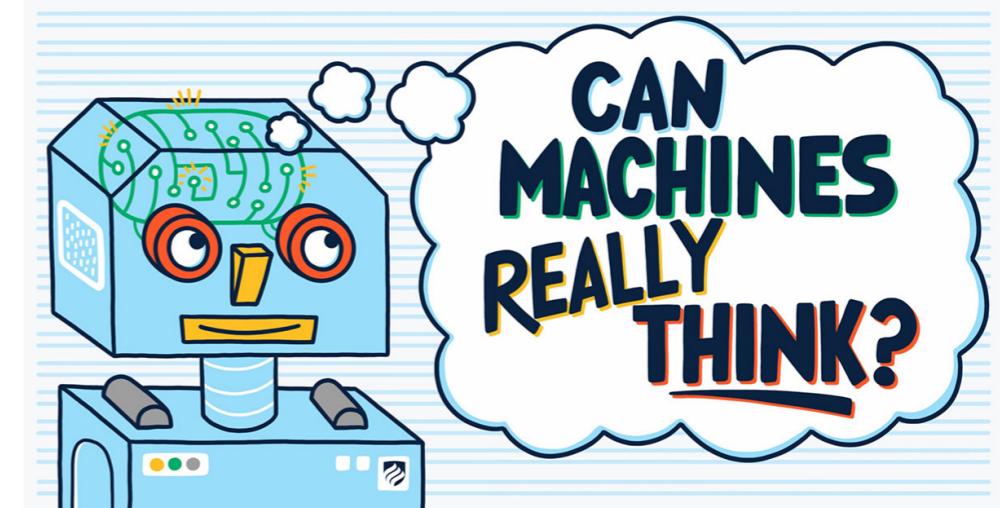
► 40 % of European AI start-ups are not using IA



Blind men and an elephant "盲人摸象"

Artificial Intelligence

Definition



A. Turing 1950

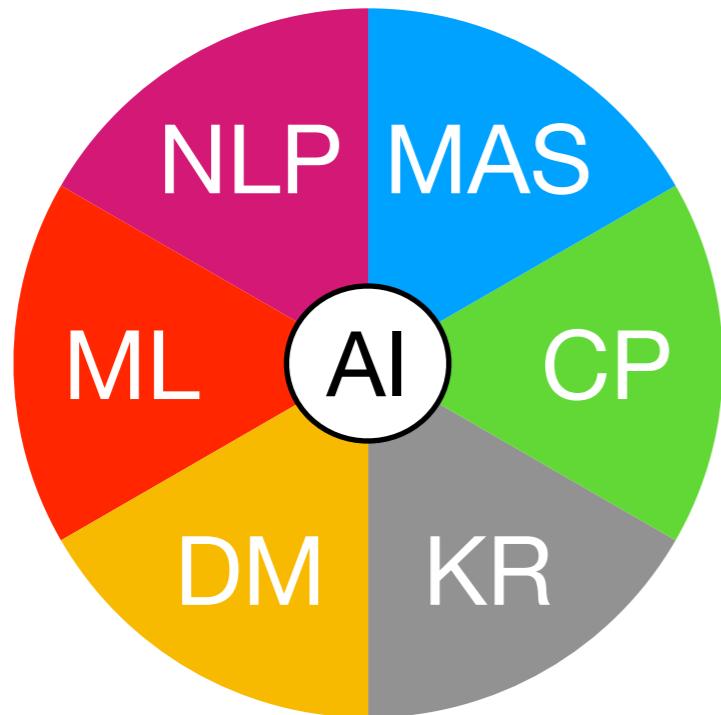
« *The science of making machines do things that would require intelligence if done by men* » Marvin Minsky

That is, be able to:

- use a language
- make abstractions and to express concepts
- solve different kind of human problems,
- improve themselves

Artificial Intelligence

AI Topics

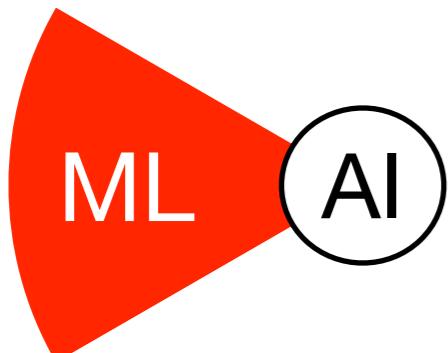


- NLP: Natural Language Processing
- MAS: Multi-Agent Systems
- KR: Knowledge Representation
- DM: Data Mining
- ML: Machine Learning
- CP: Constraint Programming

COCONUT

Artificial Intelligence

ML Topics

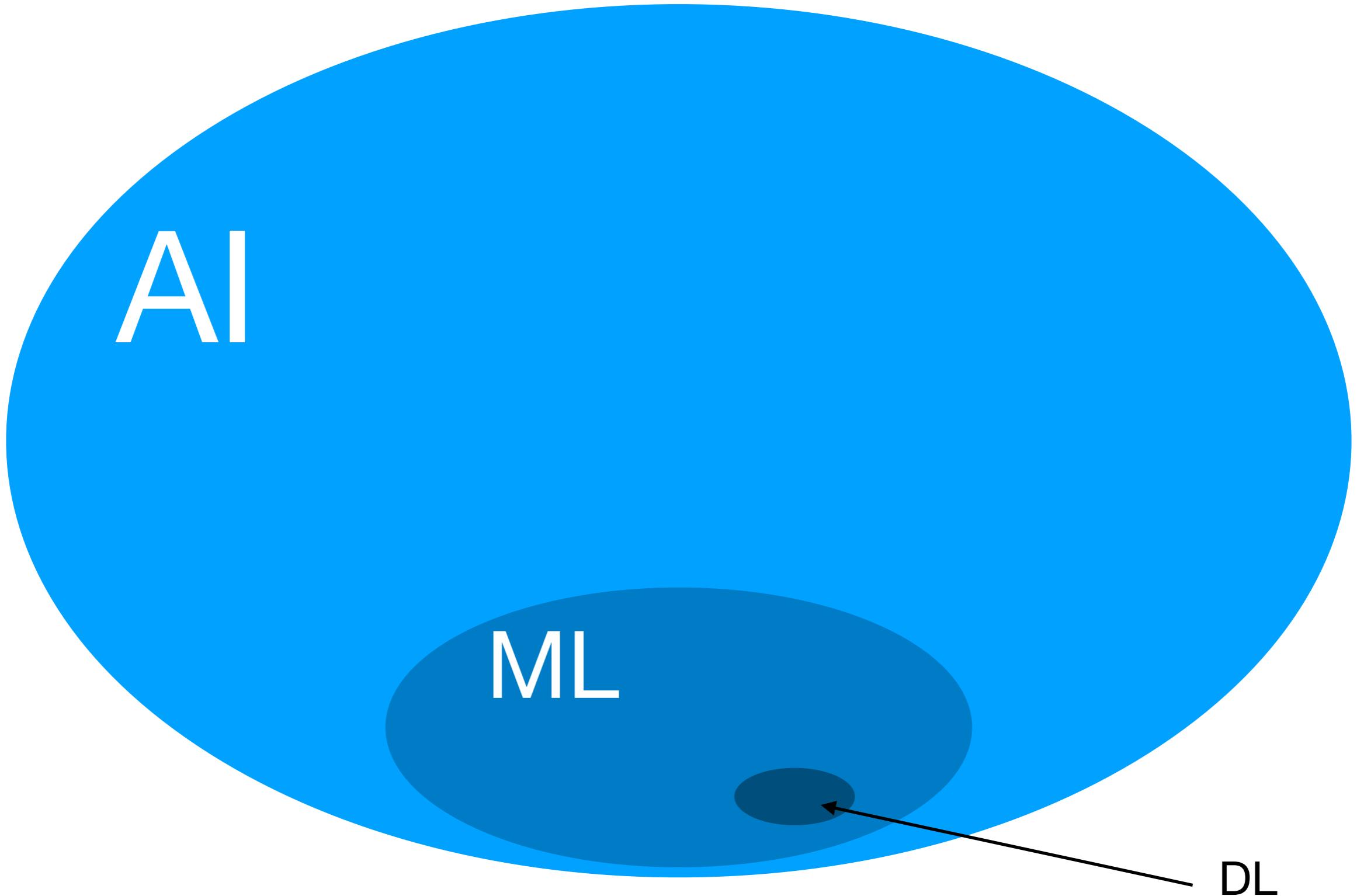


1. Active Learning
2. Adversarial Machine Learning
3. Bayesian Optimization
4. Classification
5. Clustering
6. Cost-Sensitive Learning
7. Deep Generative Models
9. Developmental Learning
10. Dimensionality Reduction and Manifold Learning
11. Ensemble Methods
12. Explainable Machine Learning
13. Feature Selection
14. Learning Sparse Models
15. Federated Learning
16. Interpretability
17. Kernel Methods
18. Knowledge-based Learning
19. Learning Generative Models
20. Learning Graphical Models
21. Learning Preferences or Rankings
22. Learning Theory
23. Multi-instance; Multi-label; Multi-view learning
24. Neuro-Symbolic Methods
25. Online Learning
26. Probabilistic Machine Learning
27. Recommender Systems
28. Reinforcement Learning
29. Relational Learning
30. Semi-Supervised Learning
31. Structured Prediction
32. Tensor and Matrix Methods
33. Time-series; Data Streams
34. Transfer, Adaptation, Multi-task Learning
35. Trusted Machine Learning
36. Unsupervised Learning

8. Deep Learning

Artificial Intelligence

ML Topics



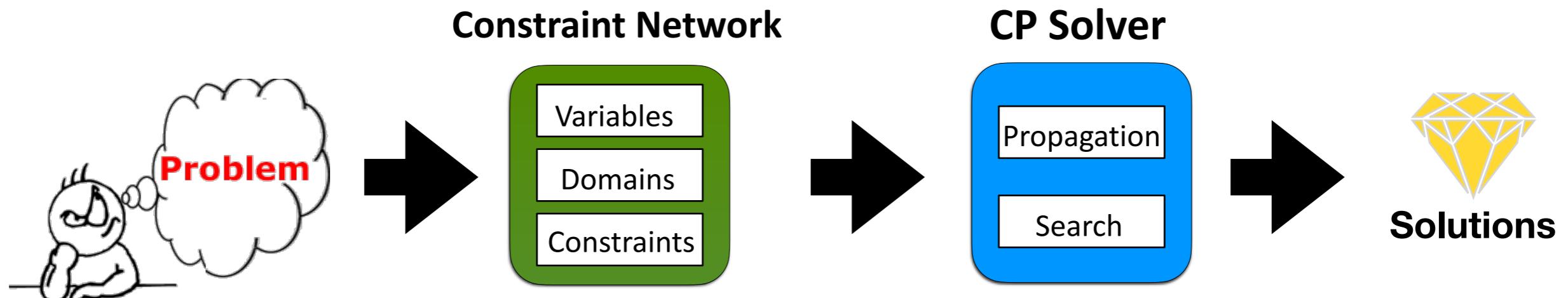
Constraint Programming

Definition

A formalism to model (**constraint network**) and to solve (**constraint solver**) combinatorial problems (**scheduling, planning,...**).

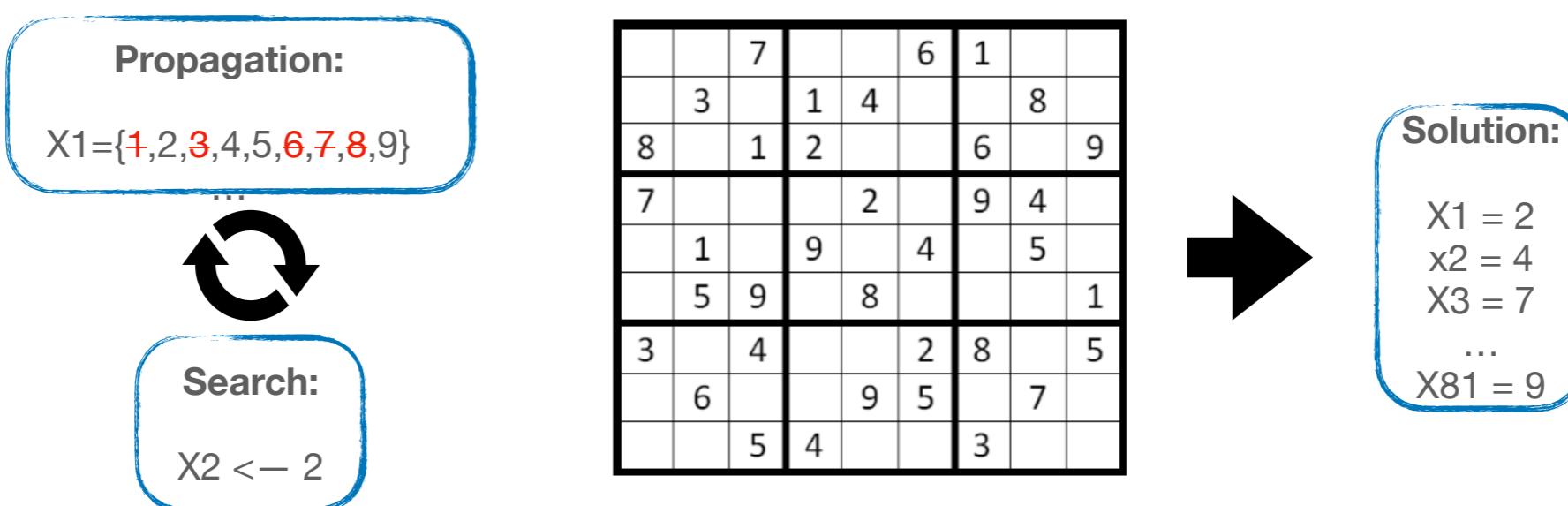
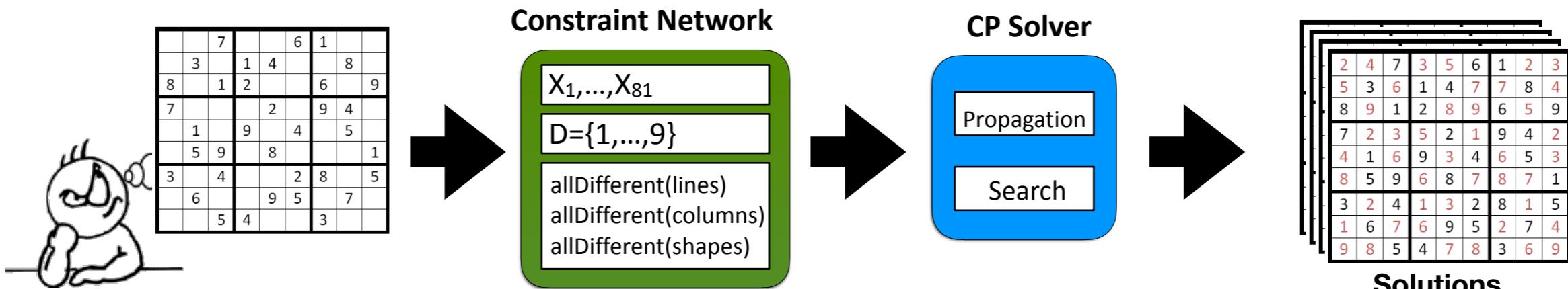
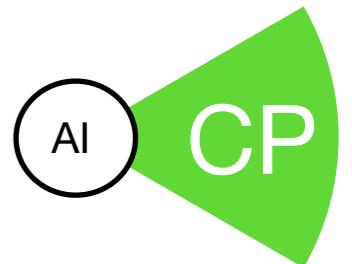
Examples of constraints :

- $X_1 + X_2 + X_3 = 5$
- `allDifferent(X1,..Xn)`
- $c(X_i, X_j) = \{(1,1), (2,1), (2,3), (4,4)\}$



Constraint Programming

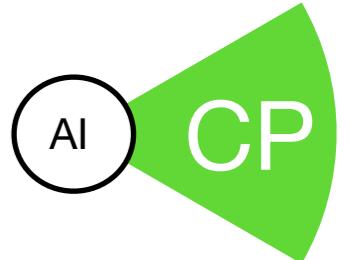
Example



Switch

Constraint Programming

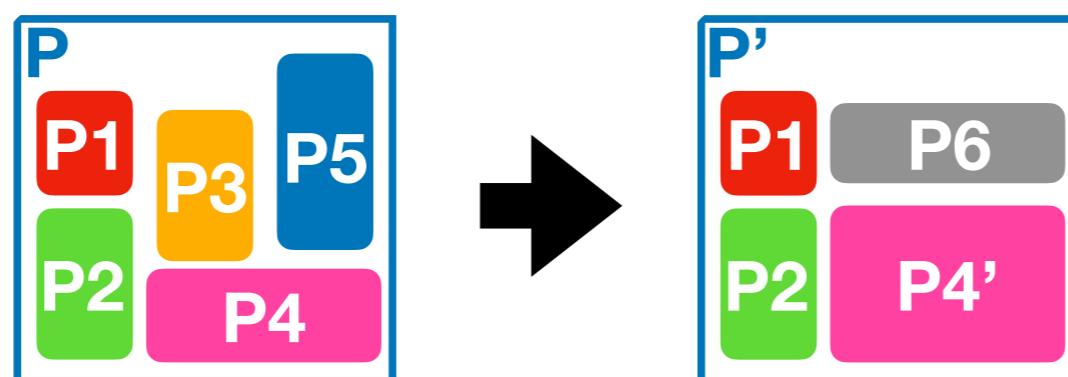
Why CP?



► Combinatorial problems can be solved by Integer Linear Programming (ILP) or by propositional satisfiability (SAT)

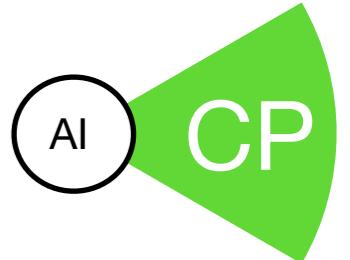
► Advantages of CP :

- Compactness
- Expressiveness
- And (often) efficiency



Constraint Programming

Why CP?



- ▶ Sudoku 9x9:
 - ▶ SAT = 6 500 clauses
 - ▶ CP: [sudoku.mod](#) with n=9

Variables $X = \{X_{1,1}, \dots, X_{n,n}\}$

Domaine des variables : $\{1, \dots, n\}$

Contraintes :

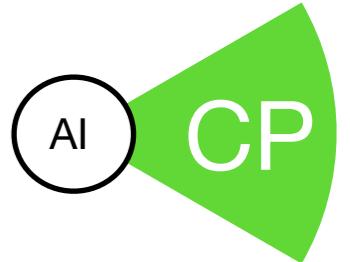
$\text{allDifferent}(X_{i,1}, \dots, X_{i,n}), \forall i$ (rows)
 $\text{allDifferent}(X_{1,j}, \dots, X_{n,j}), \forall j$ (columns)
 $\text{allDifferent}(X_{i,j}), \forall i, j$ (squares)

[sudoku.mod](#)

Sudoku 36x36

Constraint Programming

Why CP?



► Sudoku 36x36:

- SAT ~ 2 millions of clauses : cnf file of 160Gb
- CP: [sudoku.mod](#) with n=36

Variables $X = \{X_{1,1}, \dots, X_{n,n}\}$

Domaine des variables : $\{1, \dots, n\}$

Contraintes :

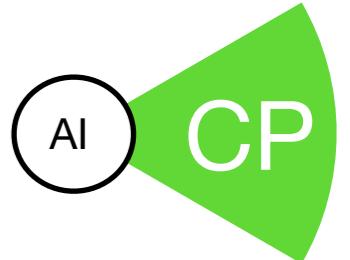
$\text{allDifferent}(X_{i,1}, \dots, X_{i,n}), \forall i$ (rows)
 $\text{allDifferent}(X_{1,j}, \dots, X_{n,j}), \forall j$ (columns)
 $\text{allDifferent}(X_{i,j}), \forall i, j$ (squares)

[sudoku.mod](#)

Switch

Constraint Programming

Why CP?

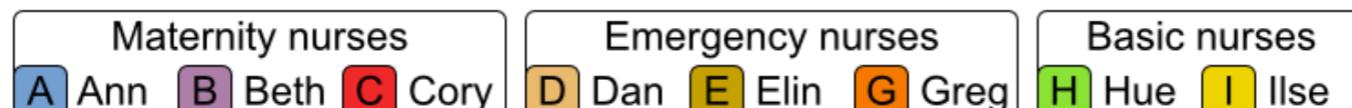


► Nurse Rostering Problem

► Best Linear Program: more than 10 000 lines

Employee shift rostering

Populate each work shift with a nurse.

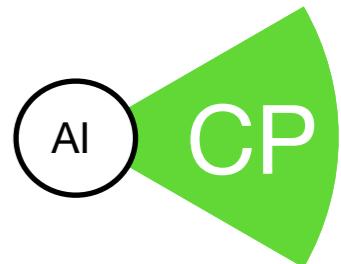


			Largest staff first			Drools Planner		
			Sat	Sun	Mon	Sat	Sun	Mon
			6 14 22	6 14 22	6 14 22	6 14 22	6 14 22	6 14 22
Maternity nurses	1 2 C A B	1 1 C A B	2 1 A C B	1 2 C A B	1 1 C A B	2 1 C A B	1 1 C A B	2 1 C A B
Emergency nurses	2 1 D G E	2 1 D G E	1 1 D E	2 1 D G E	2 1 D G E	1 1 D G	1 1 D G E	1 1 D G
Any nurses	1 1 H I	1 1 H I G H I	1 1 1 1 1	1 1 H I G H I	1 1 1 1 1	1 1 H I E H I	1 1 H I E H I	1 1 H I E H I

Switch

Constraint Programming

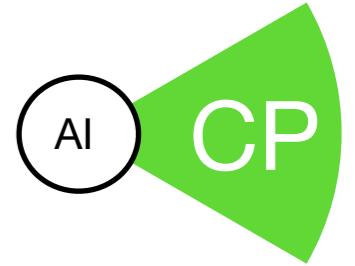
Solvers and CP platforms



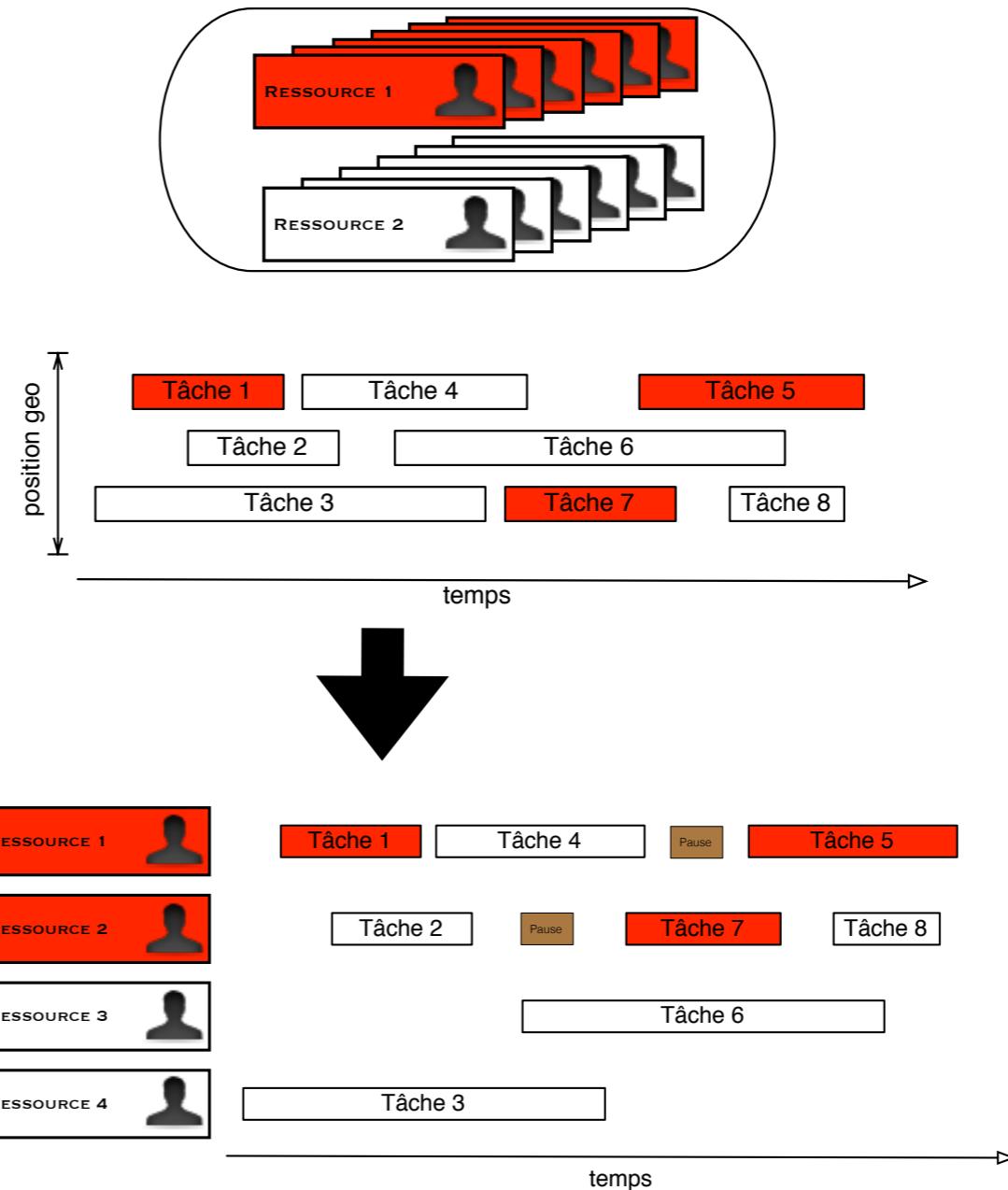
IBM ILOG CP Optimizer (Java, C++, Python, .NET)	The IBM logo, consisting of the word "IBM" in its signature blue striped font.
Google OR-Tools (C++, Java, C#, Python)	The Google OR-Tools logo, which is a red, yellow, and blue geometric triangle icon next to the text "Google OR-Tools".
Artelys Kalis (Java, C++, Python)	The Artelys Kalis logo, featuring the words "ARTELYS KALIS™" in blue and orange, with "Constraint Programming Library" in smaller blue text below it.
SICStus Prolog (CLPFD bib in prolog)	The SICStus Prolog logo, with the word "SICStus" in a bold black font and a large red number "4" to its right.
Gecode (C++)	The Gecode logo, which consists of three colored arrows (red, blue, and green) pointing towards a white letter "G".
Choco (Java)	The Choco logo, featuring a blue square with a white octopus icon and the word "CHOCO" in white.
Minizinc (high-level, solver-independent)	The Minizinc logo, which consists of the letters "Zn" in a stylized font inside a blue rounded rectangle, with the word "mini" in smaller letters above "Zn".

Constraint Programming

SNCF train driver planning using CP

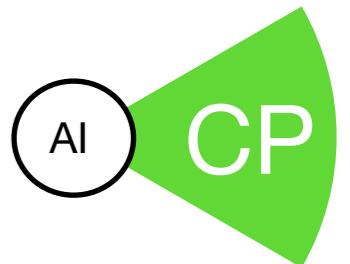


- Legal rules constraints:
 - daily working time
 - daily rest periods
 - ...
- Preferences:
 - Type of lines
 - Day of rest
 - Place of rest
 - ...
- Solution:
 - Best in terms of ressources
 - Robust one
 - Cheapest one
 - ...

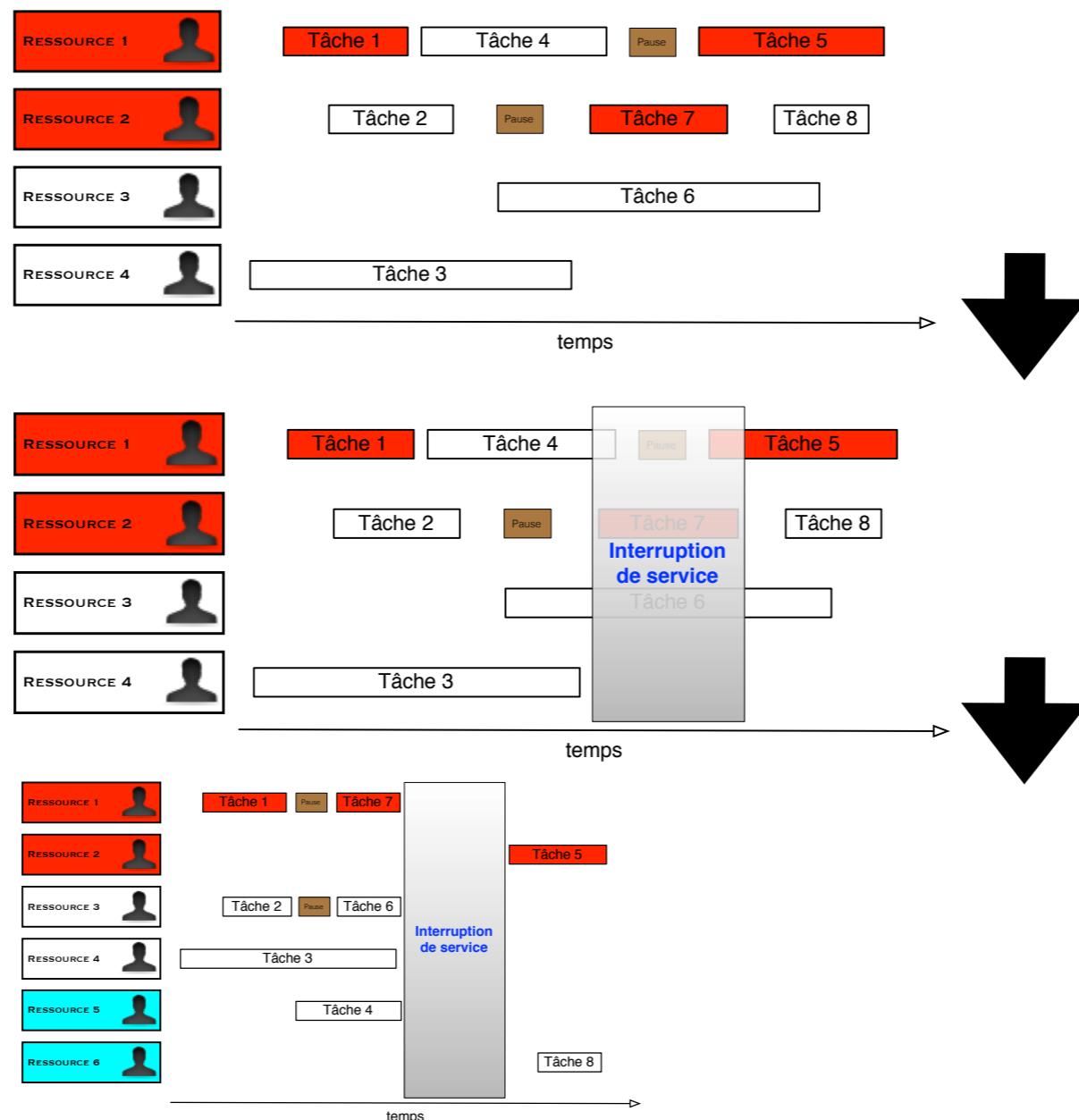


Constraint Programming

SNCF train driver planning using CP

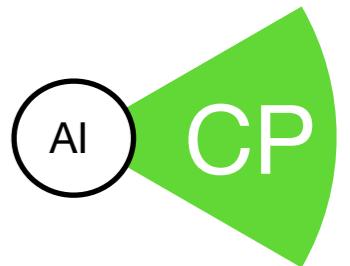


- Maintaining an existing planning



Constraint Programming

ABB Robotics partner projects



► SWMOD: Test Case Execution Scheduling with CP

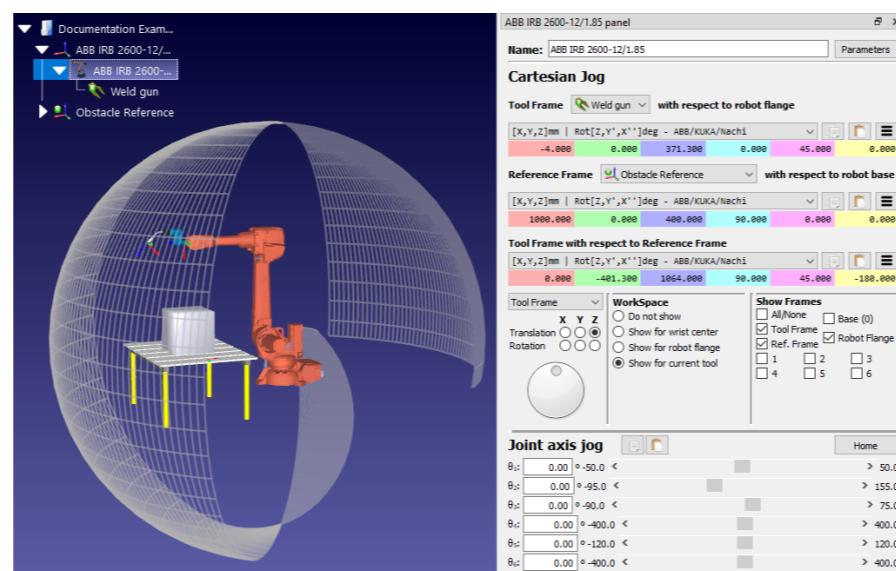


"SWMOD deployed at ABB Robotics and used every day to schedule tests throughout several ABB centers in the world (Norway, Sweden, India, China)"



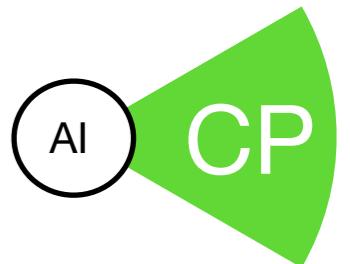
<https://github.com/Makouno44/Robtest>

► Robtest: Optimal Stress Test Trajectories for Robots with CP

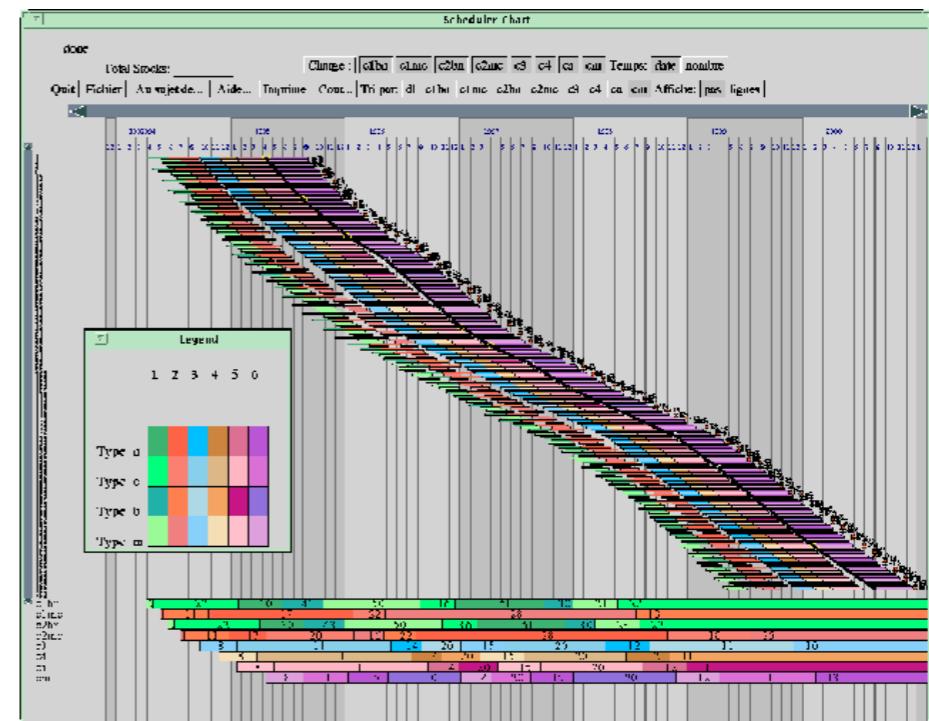


Constraint Programming

Aircraft Industry - Dassault Aviation

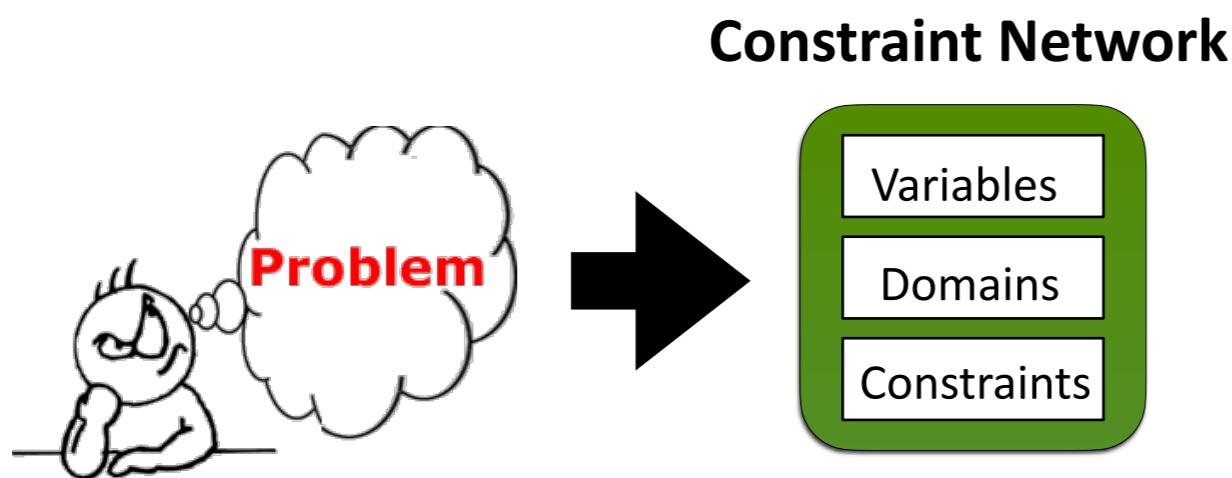


Assembly of Mirage aircraft



Constraint Programming

Modeling



Constraint Network $N=(X, D, C)$

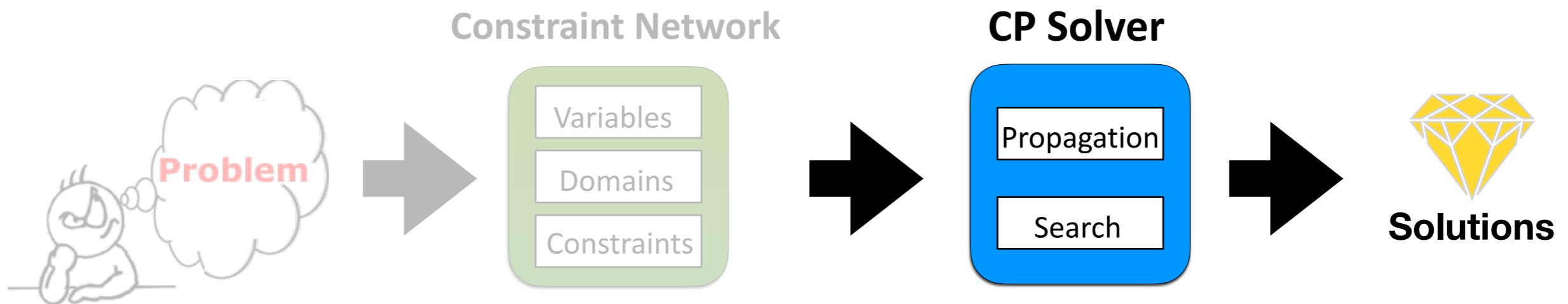
X: finite set of variables

D: domain on X, where $D(X_i)$ is a finite set of values for x_i

C: a set of constraints

Constraint Programming

Solving



Instantiation I

- Complete: assignment on X
- Partial: assignment on $Y \subset X$
- Valid: values in D
- Violating c : assignment on $\text{var}(c)$ is not in c
- Locally consistent: assignment not violating any constraint on Y
- Solution: locally consistent on X

Constraint Programming

Solving - search

Backtracking(<X,D,C>, I):

If I is complete then return true

Select a variable X_i not in I

ForEach v in $D(X_i)$ do

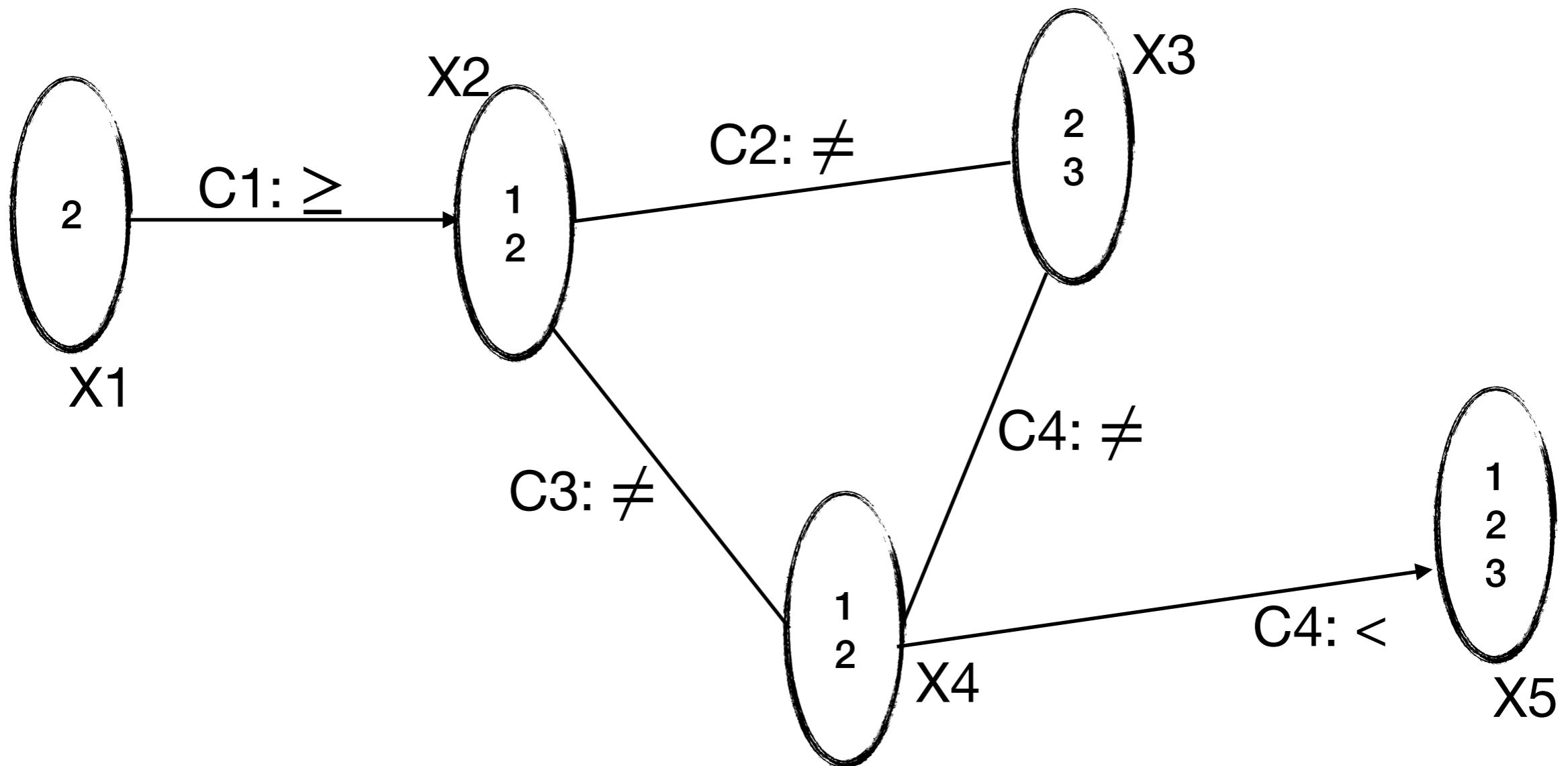
If $I \cup \langle X_i, v \rangle$ is locally consistent then

If **Backtracking(<X,D,C>, $I \cup \langle X_i, v \rangle$)** then
return true

return false

Constraint Programming

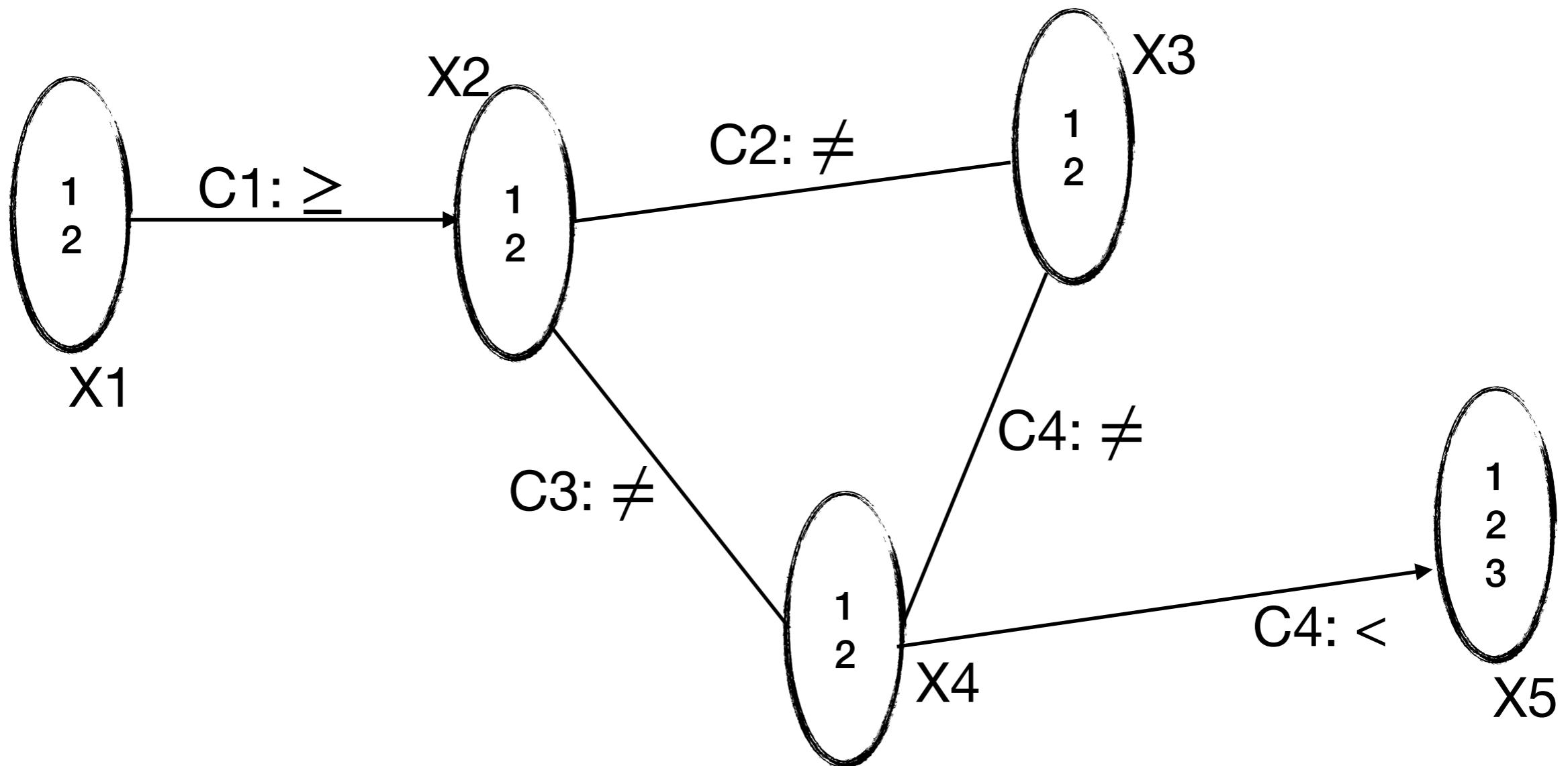
Solving - propagation (local consistency)



Situation 1

Constraint Programming

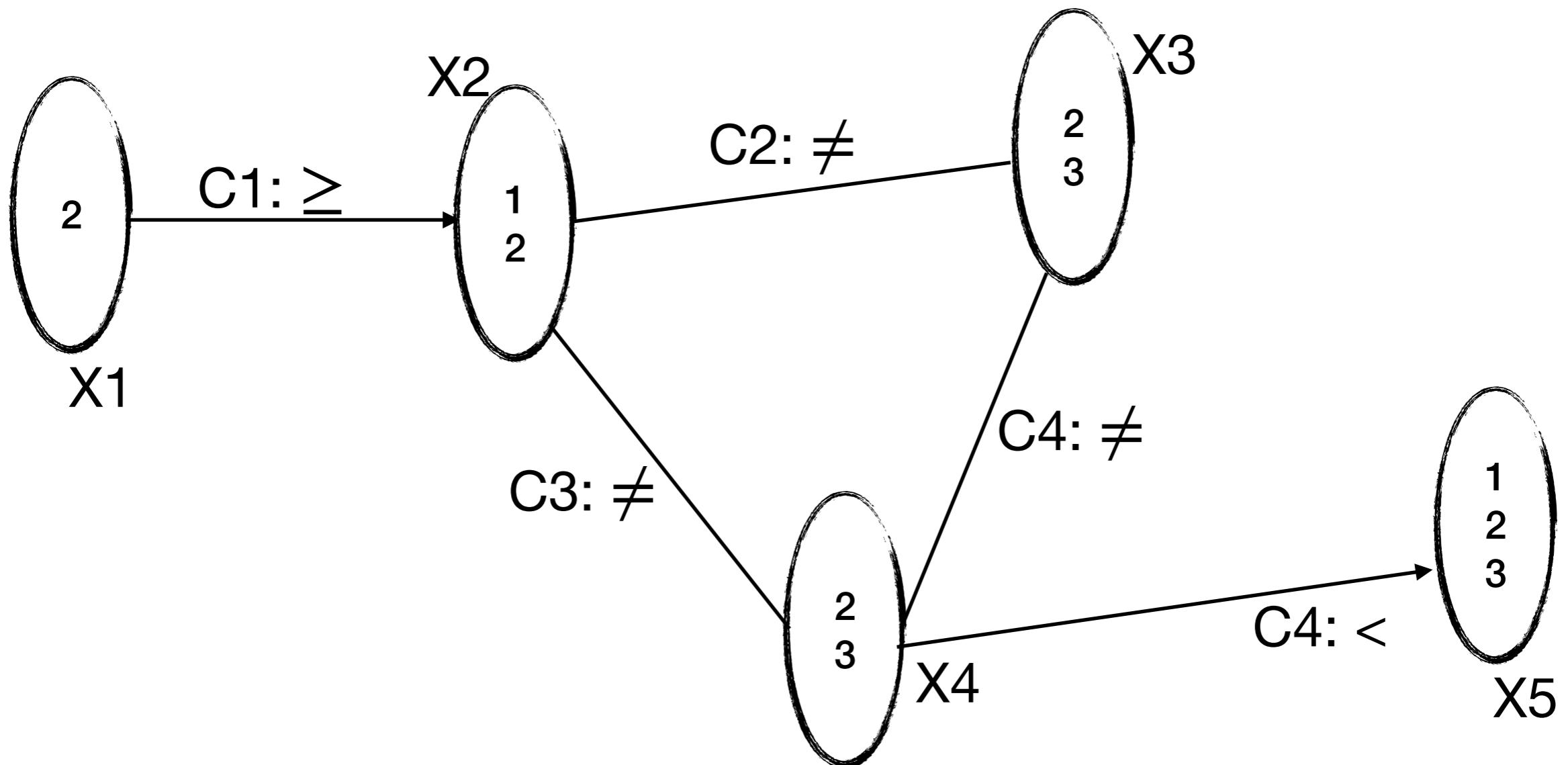
Solving - propagation (local consistency)



Situation 2

Constraint Programming

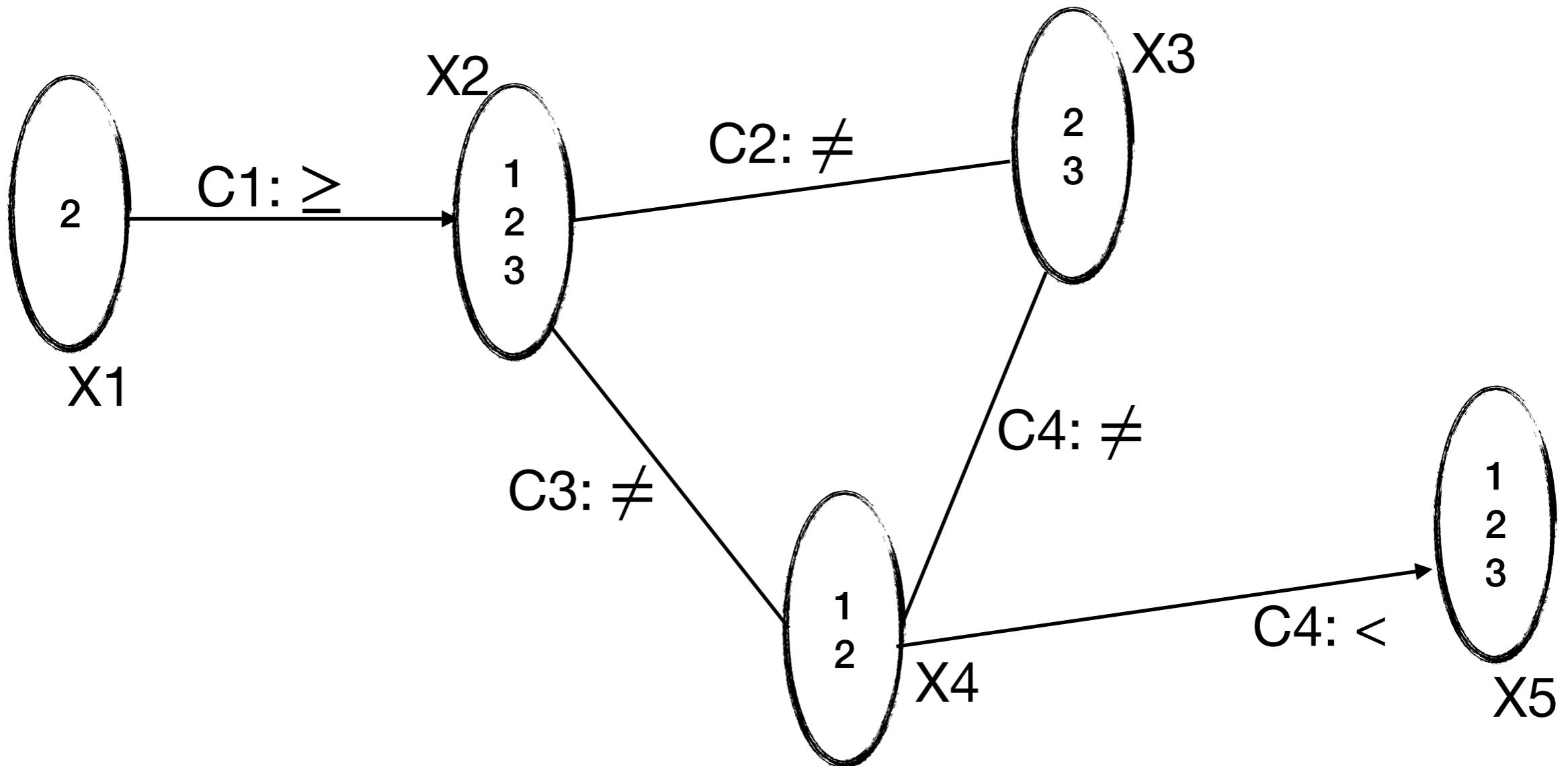
Solving - propagation (local consistency)



Situation 3

Constraint Programming

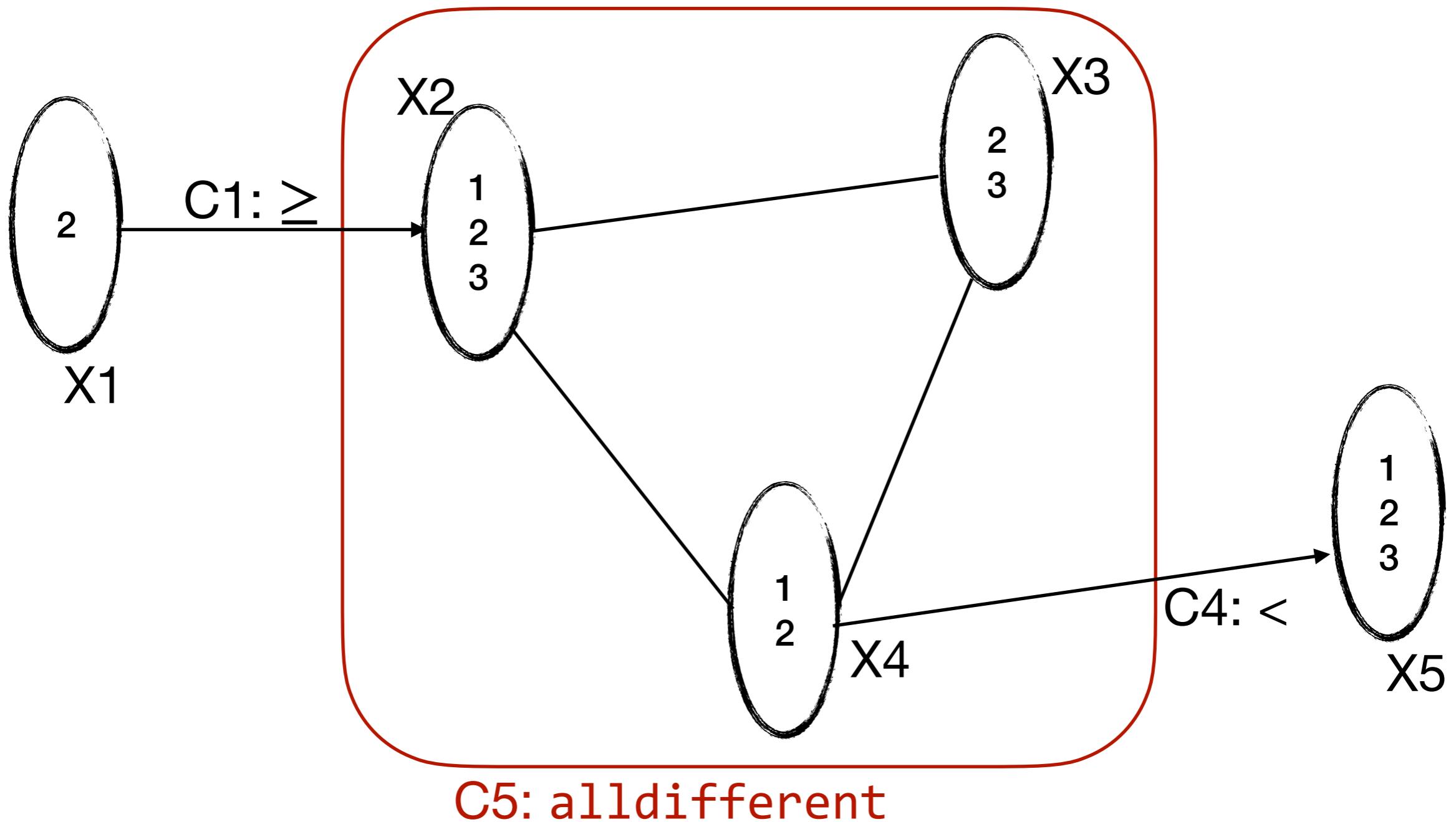
Solving - propagation (local consistency)



Situation 4

Constraint Programming

Solving - propagation (local consistency)

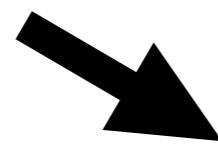
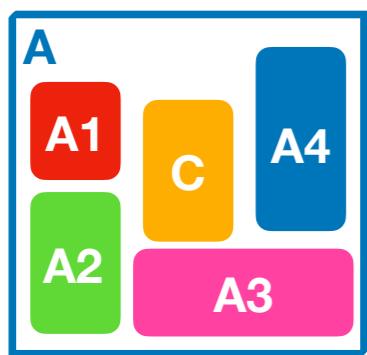
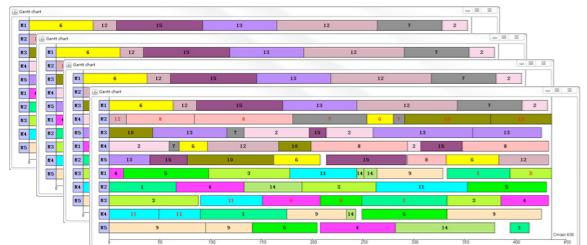


Situation 5

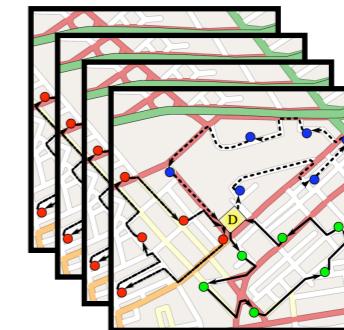
Constraint Programming

Global constraints

Scheduling instances



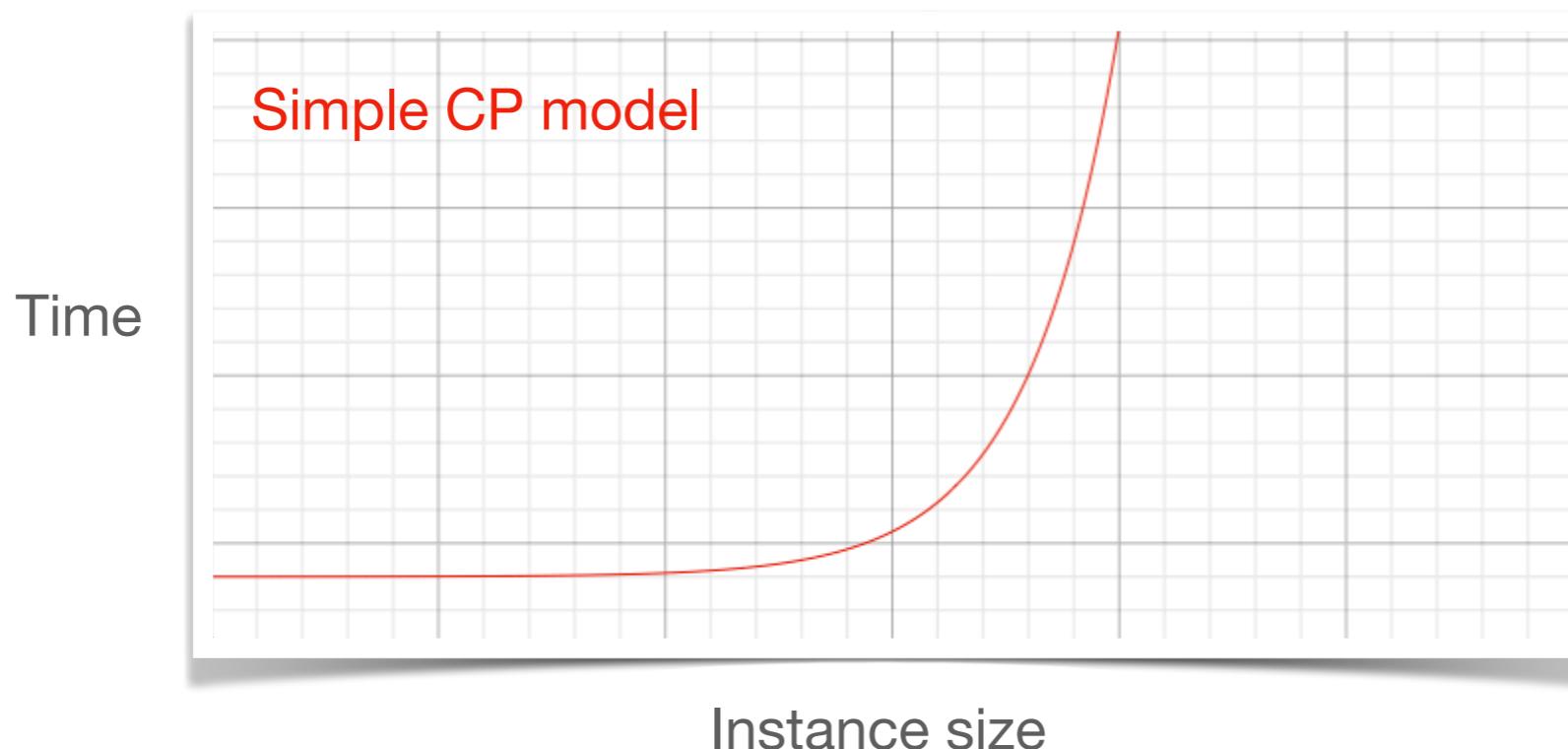
Vehicule routing instances



•
•
•

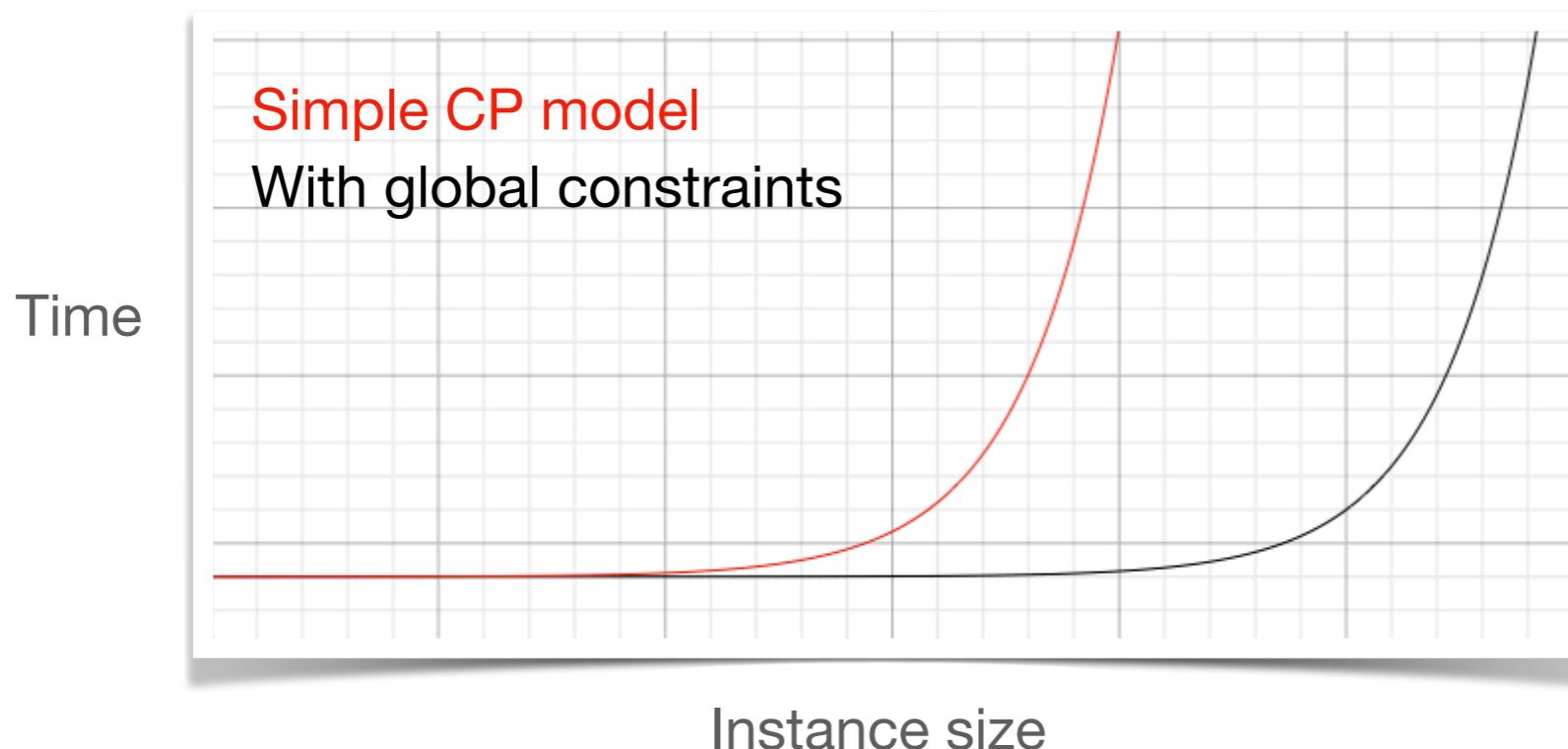
Constraint Programming

Global constraints



Constraint Programming

Global constraints



Constraint Programming

References & links

- **Handbook of Constraint Programming.** Francesca Rossi
Peter van Beek Toby Walsh. Elsevier Science 2006



- ACP: Association for Constraint Programming



- Guide to Constraint Programming

- CP conference Series

- Global Constraint Catalog

- COCONUT Team, LIRMM