Constraint Satisfaction Problems

Introduction

Bernhard Nebel and **Stefan Wölfl**

based on a slideset by

Malte Helmert and Stefan Wölfl

(summer term 2007)

Albert-Ludwigs-Universität Freiburg

October 19, 2009

Constraints

What is a constraint?

1 a: the act of constraining **b:** the state of being checked, restricted, or compelled to avoid or perform some action . . .
c: a constraining condition, agency, or force . . .
2 a: repression of one's own feelings, behavior, or actions
b: a sense of being constrained . . .

(from *Merriam-Webster's Online Dictionary*)

Usage

- In programming languages, constraints are often used to restrict the domains of variables.
- In databases, constraints can be used to specify integrity conditions.
- In mathematics, a constraint is a requirement on solutions of optimization problems.

Examples

Examples:

- Latin squares
- Eight queens problem
- Sudoku
- Map coloring problem
- Boolean satisfiability

Latin Square

Problem:

- How can one fill an $n \times n$ table with n different symbols
- … such that each symbol occurs exactly once in each row and and each column?

$$\begin{bmatrix}1\end{bmatrix} \quad \begin{bmatrix}1 & 2\\2 & 1\end{bmatrix} \quad \begin{bmatrix}1 & 2 & 3\\2 & 3 & 1\\3 & 1 & 2\end{bmatrix} \quad \begin{bmatrix}1 & 2 & 4 & 3\\2 & 3 & 1 & 4\\3 & 4 & 2 & 1\\4 & 1 & 3 & 2\end{bmatrix}$$

There are essentially 56 different Latin squares of size 5, 9408 squares of size 6, 16.942.080 squares of size 7, 535.281.401.856 squares of size 8, …

Latin Square

Problem:

- How can one fill an $n \times n$ table with n different symbols
- ... such that each symbol occurs exactly once in each row and and each column?

$$\begin{bmatrix}1\end{bmatrix} \quad \begin{bmatrix}1 & 2\\2 & 1\end{bmatrix} \quad \begin{bmatrix}1 & 2 & 3\\2 & 3 & 1\\3 & 1 & 2\end{bmatrix} \quad \begin{bmatrix}1 & 2 & 3 & 4\\2 & 3 & 4 & 1\\3 & 4 & 1 & 2\\4 & 1 & 2 & 3\end{bmatrix}$$

There are essentially 56 different Latin squares of size 5, 9408 squares of size 6, 16.942.080 squares of size 7, 535.281.401.856 squares of size 8, ...

Eight Queens Puzzle

Problem:

- How can one put 8 queens on a standard chess board (8×8-board)
- ...such that no queen can attack any other queen?

Solutions:

- The puzzle has 12 unique solutions (up to rotations and reflections)

- Old problem proposed in 1848.
- Various variants
  - knights (instead of queens)
  - 3D
  - n queens on an $n \times n$-board

A Solution . . .

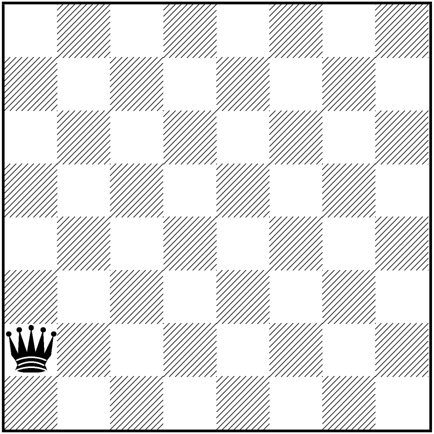


Figure: A solution of the 8-queens problem

A Solution . . .

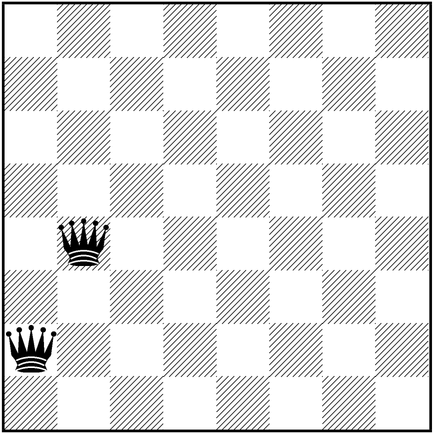


Figure: A solution of the 8-queens problem

A Solution . . .

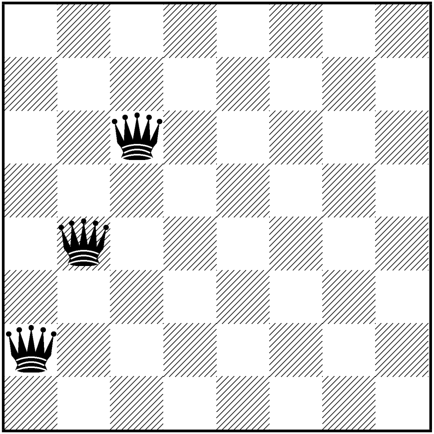


Figure: A solution of the 8-queens problem

A Solution . . .

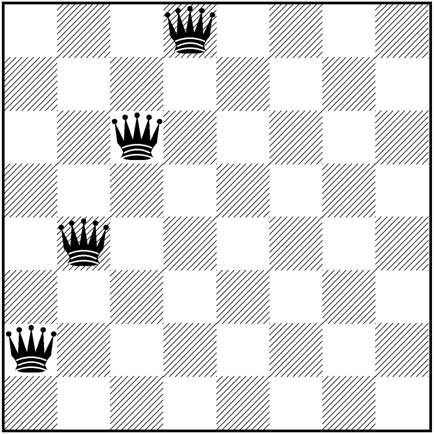


Figure: A solution of the 8-queens problem

A Solution . . .

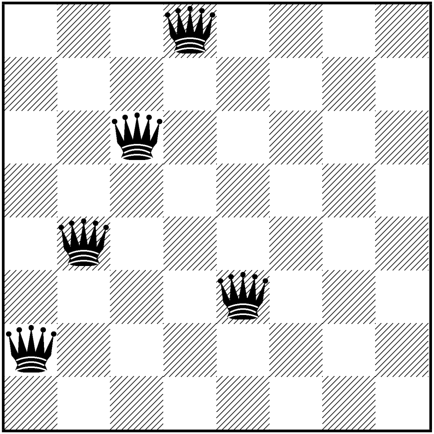


Figure: A solution of the 8-queens problem

A Solution ...

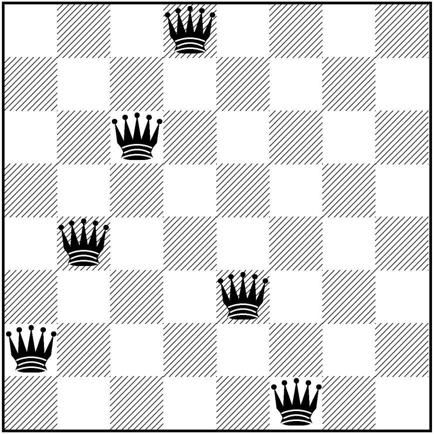


Figure: A solution of the 8-queens problem

A Solution . . .

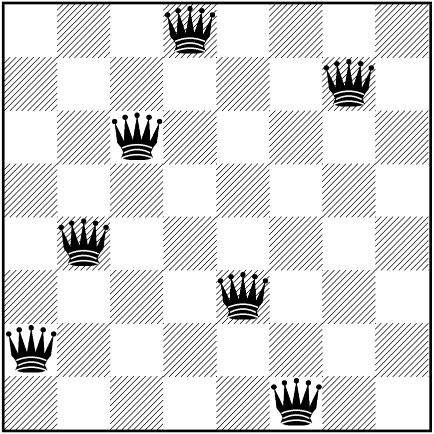


Figure: A solution of the 8-queens problem

A Solution . . .

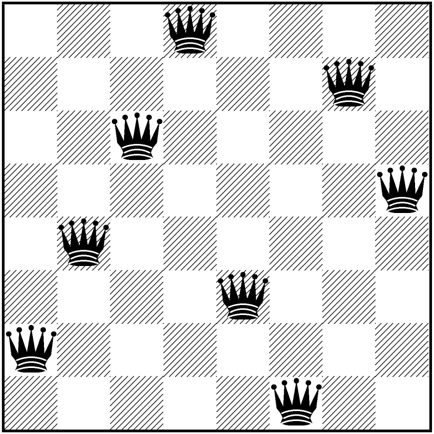


Figure: A solution of the 8-queens problem

Sudoku

Problem:

- Fill a partially completed 9×9 grid such that
- . . . each row, each column, and each of the nine 3×3 boxes contains the numbers from 1 to 9.

2	5			3		9		1
	1				4			
4		7				2		8
		5	2					
				9	8	1		
	4				3			
			3	6			7	2
	7							3
9		3				6		4

Sudoku

Problem:

- Fill a partially completed 9×9 grid such that
- … each row, each column, and each of the nine 3×3 boxes contains the numbers from 1 to 9.

2	5	8	7	3	6	9	4	1
6	1	9	8	2	4	3	5	7
4	3	7	9	1	5	2	6	8
3	9	5	2	7	1	4	8	6
7	6	2	4	9	8	1	3	5
8	4	1	6	5	3	7	2	9
1	8	4	3	6	9	5	7	2
5	7	6	1	4	2	8	9	3
9	2	3	5	8	7	6	1	4

Constraint Satisfaction Problem

Definition

A constraint network is defined by:

- a finite set of variables
- a (finite) domain of values for each variable
- a finite set of constraints (i.e., binary, ternary, ... relations defined between the variables)

Problem

Is there a solution of the network, i.e., an assignment of values to the variables such that all constraints are satisfied?

k-Colorability

Problem:

- Can one color the nodes of a given graph with k colors
- . . . such that all nodes connected by an edge have different colors?

Reformulated as a constraint network:

- Variables: the nodes of the graph;
- Domains: "colors" $\{1, \ldots, k\}$ for each variable;
- Constraints: nodes connected by an edge must have different values.

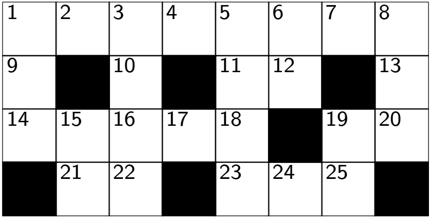
This constraint network has a particular restricted form:

- Only binary constraints.
- The domains are finite.

Crossword Puzzle

Problem instance:

- Variables: empty squares in a crossword puzzle;
- Domains: letters $\{A, B, C, \ldots, Z\}$ for each variable;
- Constraints: relations defined by a given set of words that need (or are allowed) to occur in the completed puzzle.



Fill-in words: EIER, HOLZ, IE, IM, IT, NZ, ON, RAM, RE, ROLLE, ROT, ZAR, ZUHOERER

Crossword Puzzle

Problem instance:

- Variables: empty squares in a crossword puzzle;
- Domains: letters $\{A, B, C, \ldots, Z\}$ for each variable;
- Constraints: relations defined by a given set of words that need (or are allowed) to occur in the completed puzzle.

Z	U	H	O	E	R	E	R
A		O		I	E		A
R	O	L	L	E		I	M
	N	Z		R	O	T	

Fill-in words: EIER, HOLZ, IE, IM, IT, NZ, ON, RAM, RE, ROLLE, ROT, ZAR, ZUHOERER

Boolean Satisfiability

Problem instance (Boolean constraint network):

- Variables: (propositional) variables;
- Domains: truth values $\{0, 1\}$ for each variable;
- Constraints: relations defined by a propositional formulae in these variables.

Example: $(x_1 \vee \neg x_2 \vee \neg x_3), (x_1 \vee x_2 \vee x_4)$

SAT as a constraint satisfaction problem:

Given an arbitrary Boolean constraint network, is the network solvable?

3SAT as a constraint satisfaction problem:

Given an arbitrary Boolean constraint network defined by clauses that contain exactly three literals, is the network solvable?

Real World Applications

CSP solving techniques can be used in

- *civil engineering* (design of power plants, water and energy supply, transportation and traffic infrastructure)
- *mechanical engineering* (design of machines, robots, vehicles)
- digital circuit *verification*
- automated timetabling
- air traffic control
- finance

Computational Complexity

Theorem

It is NP-hard to decide solvability of CSPs.

Since k-colorability (SAT, 3SAT) is NP-complete, solvability of CSPs in general must be NP-hard.

Question: Is CSP solvability *in* NP?

Computational Complexity

Theorem

It is NP-hard to decide solvability of CSPs.

Since k-colorability (SAT, 3SAT) is NP-complete, solvability of CSPs in general must be NP-hard.

Question: Is CSP solvability *in* NP?

Computational Complexity

Theorem

It is NP-hard to decide solvability of CSPs.

Since k-colorability (SAT, 3SAT) is NP-complete, solvability of CSPs in general must be NP-hard.

Question: Is CSP solvability *in* NP?

Solving CSPs

- Enumeration of all assignments and testing

⇝ ... too costly

- Backtracking search

⇝ numerous different strategies, often "dead" search paths are explored extensively

- Constraint propagation: elimination of obviously impossible values
- Interleaving backtracking and constraint propagation: constraint propagation at each generated search node
- Many other search methods, e. g., local search, stochastic search, etc.

Solving CSPs

- Enumeration of all assignments and testing

⇝ . . . too costly

- Backtracking search

⇝ numerous different strategies, often "dead" search paths are explored extensively

- Constraint propagation: elimination of obviously impossible values
- Interleaving backtracking and constraint propagation: constraint propagation at each generated search node
- Many other search methods, e. g., local search, stochastic search, etc.

Solving CSPs

- Enumeration of all assignments and testing

⇝ . . . too costly

- Backtracking search

⇝ numerous different strategies, often “dead” search paths are explored extensively

- Constraint propagation: elimination of obviously impossible values
- Interleaving backtracking and constraint propagation: constraint propagation at each generated search node
- Many other search methods, e. g., local search, stochastic search, etc.

Solving CSPs

- Enumeration of all assignments and testing

⇝ . . . too costly

- Backtracking search

⇝ numerous different strategies, often “dead” search paths are explored extensively

- Constraint propagation: elimination of obviously impossible values
- Interleaving backtracking and constraint propagation: constraint propagation at each generated search node
- Many other search methods, e. g., local search, stochastic search, etc.

Solving CSPs

- Enumeration of all assignments and testing

⇝ . . . too costly

- Backtracking search

⇝ numerous different strategies, often “dead” search paths are explored extensively

- Constraint propagation: elimination of obviously impossible values
- Interleaving backtracking and constraint propagation: constraint propagation at each generated search node
- Many other search methods, e. g., local search, stochastic search, etc.

Contents

Contents

- *Introduction and mathematical background*
  - Sets, relations, graphs
  - Constraint networks and satisfiability
  - Binary Constraint Networks
  - Simple solution methods (backtracking, etc.)
- *Inference-based methods*
  - Arc and path consistency
  - k-consistency and global consistency
- *Search methods*
  - Backtracking
  - Backjumping
  - Comparing different methods
  - Stochastic local search
- *Selected advanced topics*
  - Expressiveness vs complexity of constraint formalisms
  - Qualitative constraint networks

Contents

- *Introduction and mathematical background*
  - Sets, relations, graphs
  - Constraint networks and satisfiability
  - Binary Constraint Networks
  - Simple solution methods (backtracking, etc.)
- *Inference-based methods*
  - Arc and path consistency
  - k-consistency and global consistency
- *Search methods*
  - Backtracking
  - Backjumping
  - Comparing different methods
  - Stochastic local search
- *Selected advanced topics*
  - Expressiveness vs complexity of constraint formalisms
  - Qualitative constraint networks

Contents

- *Introduction and mathematical background*
  - Sets, relations, graphs
  - Constraint networks and satisfiability
  - Binary Constraint Networks
  - Simple solution methods (backtracking, etc.)
- *Inference-based methods*
  - Arc and path consistency
  - k-consistency and global consistency
- *Search methods*
  - Backtracking
  - Backjumping
  - Comparing different methods
  - Stochastic local search
- *Selected advanced topics*
  - Expressiveness vs complexity of constraint formalisms
  - Qualitative constraint networks

Lectures: Where, When, Web Page

Where

101, Room 00-010/14

When

Monday: 11:15–13:00

Wednesday: 11:15–12:00 (+ exercises: 12:15–13:00)

Winter Term Holidays

December 24, 2009 – January 6, 2010

Web Page

`http://www.informatik.uni-freiburg.de/~ki/teaching/ws0910/csp/`

Lecturers

Prof. Bernhard Nebel

Room 52-00-029
Consultation: Wednesday, 13-14
Phone: 0761/203-8221
Email: nebel@informatik.uni-freiburg.de

Dr. Stefan Wölfl

Room 52-00-043
Consultation: by appointment
Phone: 0761/203-8228
Email: woelfl@informatik.uni-freiburg.de

Exercises

Who

Robert Mattmüller
Room 52-00-045
Consultation: by appointment
Phone: 0761/203-8229
Email: mattmuel@informatik.uni-freiburg.de

Matthias Westphal
Room 52-00-045
Consultation: by appointment
Phone: 0761/203-8227
Email: westpham@informatik.uni-freiburg.de

Where

101, Room 00-010/14

Examination

Exams

- Oral exams in February/April 2010

Literature

- Rina Dechter, *Constraint Processing*,
  Morgan Kaufmann, 2003.

- Further readings will be given during the lecture.

Course Prerequisites & Goals

Goals

- Acquiring skills in constraint processing
- Understanding the principles behind different solving techniques
- Being able to read and understand research literature in the area of constraint satisfaction
- Being able to complete a project (thesis) in this research area

Prerequisites

- Basic knowledge in the area of AI
- Basic knowledge in formal logic
- Basic knowledge in theoretical computer science