

Foundations of Artificial Intelligence

22. Constraint Satisfaction Problems: Introduction and Examples

Malte Helmert

University of Basel

April 6, 2022

Foundations of Artificial Intelligence

April 6, 2022 — 22. Constraint Satisfaction Problems: Introduction and Examples

22.1 Introduction

22.2 Examples

22.3 Summary

Classification

Classification:

Constraint Satisfaction Problems environment:

- ▶ **static** vs. dynamic
- ▶ **deterministic** vs. non-deterministic vs. stochastic
- ▶ **fully** vs. partially vs. not **observable**
- ▶ **discrete** vs. continuous
- ▶ **single-agent** vs. multi-agent

problem solving method:

- ▶ problem-specific vs. **general** vs. learning

Constraint Satisfaction Problems: Overview

Chapter overview: constraint satisfaction problems

- ▶ 22.–23. Introduction
 - ▶ **22. Introduction and Examples**
 - ▶ 23. Constraint Networks
- ▶ 24.–26. Basic Algorithms
- ▶ 27.–28. Problem Structure

22.1 Introduction

Constraints

What is a Constraint?

a condition that every solution to a problem must satisfy

German: Einschränkung, Nebenbedingung (math.)

Examples: Where do constraints occur?

- ▶ **mathematics**: requirements on solutions of optimization problems (e.g., equations, inequalities)
- ▶ **software testing**: specification of invariants to check data consistency (e.g., assertions)
- ▶ **databases**: integrity constraints

Constraint Satisfaction Problems: Informally

Given:

- ▶ set of **variables** with corresponding domains
- ▶ set of **constraints** that the variables must satisfy
 - ▶ most commonly **binary**, i.e., every constraint refers to **two** variables

Solution:

- ▶ **assignment** to the variables that satisfies all constraints

German: Variablen, Constraints, binär, Belegung

22.2 Examples

Examples

Examples

- ▶ 8 queens problem
- ▶ Latin squares
- ▶ Sudoku
- ▶ graph coloring
- ▶ satisfiability in propositional logic

German: 8-Damen-Problem, lateinische Quadrate, Sudoku, Graphfärbung, Erfüllbarkeitsproblem der Aussagenlogik

more complex examples:

- ▶ systems of equations and inequalities
- ▶ database queries

Example: 8 Queens Problem (Reminder)

(reminder from previous two chapters)

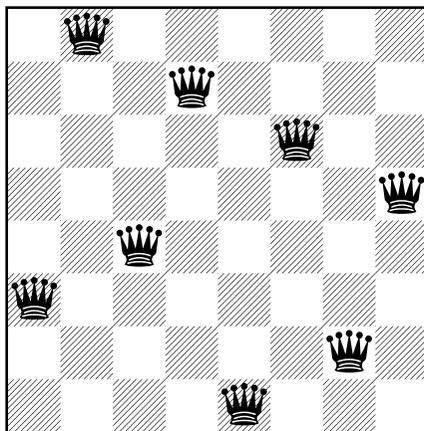
8 Queens Problem

How can we

- ▶ place **8 queens** on a chess board
 - ▶ such that **no two queens threaten each other?**
- ▶ originally proposed in 1848
 - ▶ **variants:** board size; other pieces; higher dimension

There are **92 solutions**, or **12 solutions** if we do not count symmetric solutions (under rotation or reflection) as distinct.

8 Queens Problem: Example Solution



example solution for the 8 queens problem

Example: Latin Squares

Latin Squares

How can we

- ▶ build an $n \times n$ matrix with n symbols
- ▶ such that **every symbol occurs exactly once** in every **row** and every **column**?

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

There exist 12 different Latin squares of size 3,
576 of size 4, 161 280 of size 5, . . . ,
5 524 751 496 156 892 842 531 225 600 of size 9.

Example: Sudoku

Sudoku

How can we

- ▶ completely fill an already partially filled 9×9 matrix with numbers between 1–9
- ▶ such that each row, each column, and each of the nine 3×3 blocks contains every number exactly once?

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			

relationship to Latin squares?

Sudoku: Trivia

- ▶ well-formed Sudokus have **exactly one** solution
- ▶ to achieve well-formedness, ≥ 17 cells must be filled already (McGuire et al., 2012)
- ▶ 6 670 903 752 021 072 936 960 solutions
- ▶ only 5 472 730 538 “non-symmetrical” solutions

Example: Graph Coloring

Graph Coloring

How can we

- ▶ **color the vertices** of a given graph using k colors
- ▶ such that two neighboring vertices **never have the same** color?

(The graph and k are problem parameters.)

NP-complete problem

- ▶ even for the special case of planar graphs and $k = 3$
- ▶ easy for $k = 2$ (also for general graphs)

Relationship to Sudoku?

Four Color Problem

famous problem in mathematics: **Four Color Problem**

- ▶ Is it always possible to color a **planar** graph with 4 colors?
- ▶ conjectured by Francis Guthrie (1852)
- ▶ 1890 first proof that 5 colors suffice
- ▶ several wrong proofs surviving for over 10 years
- ▶ solved by Appel and Haken in 1976: 4 colors suffice
- ▶ Appel and Haken reduced the problem to 1936 cases, which were then checked by computers
- ▶ first famous mathematical problem solved (partially) by computers
 - ↔ led to controversy: is this a mathematical proof?

Numberphile video:

<https://www.youtube.com/watch?v=NgBK43jB4rQ>

Satisfiability in Propositional Logic

Satisfiability in Propositional Logic

How can we

- ▶ assign **truth values** (true/false) to a set of propositional variables
- ▶ such that a given set of **clauses** (formulas of the form $X \vee \neg Y \vee Z$) is satisfied (true)?

remarks:

- ▶ NP-complete (Cook 1971; Levin 1973)
- ▶ formulas expressed as clauses (instead of arbitrary propositional formulas) is no restriction
- ▶ clause length bounded by 3 would not be a restriction

relationship to previous problems (e.g., Sudoku)?

Practical Applications

- ▶ There are **thousands** of practical applications of constraint satisfaction problems.
- ▶ This statement is true already for the satisfiability problem of propositional logic.

some examples:

- ▶ verification of hardware and software
- ▶ timetabling (e.g., generating time schedules, room assignments for university courses)
- ▶ assignment of frequency spectra (e.g., broadcasting, mobile phones)

22.3 Summary

Summary

- ▶ **constraint satisfaction:**
 - ▶ find **assignment** for a set of **variables**
 - ▶ with given **variable domains**
 - ▶ that satisfies a given set of **constraints**.
- ▶ **examples:**
 - ▶ 8 queens problem
 - ▶ Latin squares
 - ▶ Sudoku
 - ▶ graph coloring
 - ▶ satisfiability in propositional logic
 - ▶ many practical applications