Artificial Intelligence

CSP: Constraint Satisfaction Problems

Jendrik Seipp

Linköping University

based on slides by Thomas Keller and Malte Helmert (University of Basel)

Questions?

post feedback and ask questions anonymously at

`https://padlet.com/jendrikseipp/tddc17`

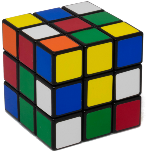
Intended Learning Outcomes

- explain what “constraint satisfaction problems” (CSPs) are
- model and solve simple CSPs

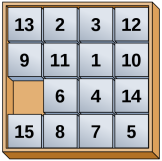
Constraint Satisfaction Problems

Constraint Satisfaction Problems

(heuristic) search algorithms considered so far:

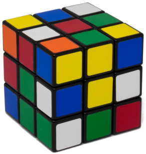


- wide variety of problems
- problem-specific heuristics
- no general solver

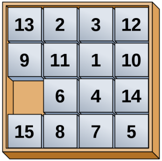


Constraint Satisfaction Problems

(heuristic) search algorithms considered so far:

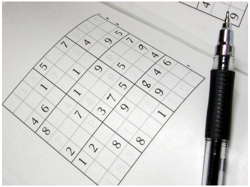


- wide variety of problems
- problem-specific heuristics
- no general solver

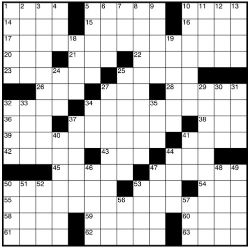


constraint satisfaction problems (CSP) considered today:

- problem scope more restricted
- problem-independent methods
- general solver

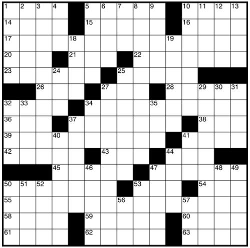


Informal Description



- a constraint is a condition that every solution to a problem must satisfy
- a CSP is defined by
  - a finite set of variables
  - a domain for each variable
  - a set of constraints
- a solution for a CSP is an assignment to all variables that satisfies all constraints

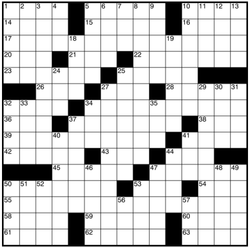
Informal Description



- a constraint is a condition that every solution to a problem must satisfy
- a CSP is defined by
  - a finite set of variables
  - a domain for each variable
  - a set of constraints
- a solution for a CSP is an assignment to all variables that satisfies all constraints

- variables and domains in cross-word puzzle? constraints?

Informal Description

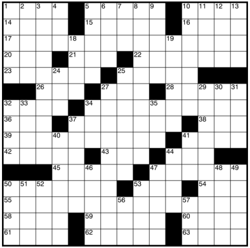


- a constraint is a condition that every solution to a problem must satisfy
- a CSP is defined by
  - a finite set of variables
  - a domain for each variable
  - a set of constraints
- a solution for a CSP is an assignment to all variables that satisfies all constraints

- variables and domains in cross-word puzzle? constraints?

$\longrightarrow$ one variable for each cell with domain $\{A, \ldots, Z\}$

Informal Description



- a constraint is a condition that every solution to a problem must satisfy
- a CSP is defined by
  - a finite set of variables
  - a domain for each variable
  - a set of constraints
- a solution for a CSP is an assignment to all variables that satisfies all constraints

- variables and domains in cross-word puzzle? constraints?

$\longrightarrow$ one variable for each cell with domain $\{A, \dots, Z\}$

$\longrightarrow$ constraints: length of “boxes”, horizontal and vertical words must not contradict each other

Running Example: Simple Math Puzzle

informal description:

- assign a value from $\{1, 2, 3, 4\}$ to the variables w and y
- and from $\{1, 2, 3\}$ to x and z
- such that
  - $w = 2x$,
  - $w < z$ and
  - $y > z$.

Constraint Satisfaction Problems

Definition

Definition (binary constraint satisfaction problem)

A (binary) constraint satisfaction problem (or constraint network) is a 3-tuple $\mathcal{C} = \langle V, dom, (C_{u,v}) \rangle$, where

- V is a non-empty and finite set of variables,
- dom is a function that assigns a non-empty and finite domain to each variable $v \in V$, and
- $(C_{u,v})_{u,v \in V, u \neq v}$ is an indexed family of binary relations over V, the constraints $C_{u,v} \subseteq dom(u) \times dom(v)$.

Definition

Definition (binary constraint satisfaction problem)

A (binary) constraint satisfaction problem (or constraint network) is a 3-tuple $\mathcal{C} = \langle V, dom, (C_{u,v}) \rangle$, where

- V is a non-empty and finite set of variables,
- dom is a function that assigns a non-empty and finite domain to each variable $v \in V$, and
- $(C_{u,v})_{u,v \in V, u \neq v}$ is an indexed family of binary relations over V, the constraints $C_{u,v} \subseteq dom(u) \times dom(v)$.

possible generalizations:

- infinite domains (e.g., $dom(v) = \mathbb{Z}$)
- constraints of higher arity (e.g., satisfiability in propositional logic)

Definition

Definition (binary constraint satisfaction problem)

A (binary) constraint satisfaction problem (or constraint network) is a 3-tuple $\mathcal{C} = \langle V, dom, (C_{u,v}) \rangle$, where

- V is a non-empty and finite set of variables,
- dom is a function that assigns a non-empty and finite domain to each variable $v \in V$, and
- $(C_{u,v})_{u,v \in V, u \neq v}$ is an indexed family of binary relations over V, the constraints $C_{u,v} \subseteq dom(u) \times dom(v)$.

Example (Simple math puzzle)

(partial) informal description:

- assign a value from $\{1, 2, 3, 4\}$ to the variables w and y
- and from $\{1, 2, 3\}$ to x and z
- such that ...

(partial) formal model:

- $V = \{w, x, y, z\}$
- $dom(w) = dom(y) = \{1, 2, 3, 4\}$
  $dom(x) = dom(z) = \{1, 2, 3\}$
- ...

Binary Constraints

a binary constraint $C_{u,v}$ with $u, v \in V$

- expresses which joint assignments to u and v are allowed in solutions

simple math puzzle:

(partial) informal description:

- ... such that
  - $w = 2x$,
  - $w < z$ and
  - $y > z$.

some constraints:

$$C_{w,z} = \{\langle 1, 2\rangle, \langle 1, 3\rangle, \langle 2, 3\rangle\}$$

Binary Constraints

a binary constraint $C_{u,v}$ with $u, v \in V$

- expresses which joint assignments to u and v are allowed in solutions
- is trivial if $C_{u,v} = dom(u) \times dom(v)$
  $\rightsquigarrow$ there is no restriction on the joint assignment to u and v and $C_{u,v}$ is usually not given explicitly (but exists formally!)

simple math puzzle:

(partial) informal description:

- ... such that
  - $w = 2x$,
  - $w < z$ and
  - $y > z$.

some constraints:

$$C_{w,z} = \{\langle 1, 2\rangle, \langle 1, 3\rangle, \langle 2, 3\rangle\}$$

$$C_{x,z} = \{\langle 1, 1\rangle, \langle 1, 2\rangle, \langle 1, 3\rangle, \langle 2, 1\rangle, \langle 2, 2\rangle, \langle 2, 3\rangle, \langle 3, 1\rangle, \langle 3, 2\rangle, \langle 3, 3\rangle\}$$

Binary Constraints

a binary constraint $C_{u,v}$ with $u, v \in V$

- expresses which joint assignments to u and v are allowed in solutions
- is trivial if $C_{u,v} = dom(u) \times dom(v)$
  $\rightsquigarrow$ there is no restriction on the joint assignment to u and v and $C_{u,v}$ is usually not given explicitly (but exists formally!)
- refers to the same variables as constraint $C_{v,u}$
  $\rightsquigarrow$ usually, only one of them is given explicitly

simple math puzzle:

(partial) informal description:

- . . . such that
  - $w = 2x$,
  - $w < z$ and
  - $y > z$.

some constraints:

$$
\begin{aligned}
C_{w,z} &= \{\langle 1,2\rangle, \langle 1,3\rangle, \langle 2,3\rangle\} \\
C_{x,z} &= \{\langle 1,1\rangle, \langle 1,2\rangle, \langle 1,3\rangle, \\
&\quad \langle 2,1\rangle, \langle 2,2\rangle, \langle 2,3\rangle, \\
&\quad \langle 3,1\rangle, \langle 3,2\rangle, \langle 3,3\rangle\} \\
C_{z,w} &= \{\langle 2,1\rangle, \langle 3,1\rangle, \langle 3,2\rangle\}
\end{aligned}
$$

Simple Math Puzzle: Formal Model

informal description:

- assign a value from $\{1, 2, 3, 4\}$ to the variables w and y
- and from $\{1, 2, 3\}$ to x and z
- such that
  - $w = 2x$,
  - $w < z$ and
  - $y > z$.

formal model:

- $V = \{w, x, y, z\}$
- $dom(w) = dom(y) = \{1, 2, 3, 4\}$
  $dom(x) = dom(z) = \{1, 2, 3\}$
- $C_{wx} = \{\langle 2, 1\rangle, \langle 4, 2\rangle\}$
  $C_{wz} = \{\langle 1, 2\rangle, \langle 1, 3\rangle, \langle 2, 3\rangle\}$
  $C_{yz} = \{\langle 2, 1\rangle, \langle 3, 1\rangle, \langle 3, 2\rangle, \langle 4, 1\rangle, \langle 4, 2\rangle, \langle 4, 3\rangle\}$

Compact Encodings and General CSP Solvers

CSPs allow for compact encodings of large sets of assignments:

- consider a CSP with n variables with domains of size k

$\rightsquigarrow$ k^n assignments

Compact Encodings and General CSP Solvers

CSPs allow for compact encodings of large sets of assignments:

- consider a CSP with n variables with domains of size k

$\rightsquigarrow$ k^n assignments

- for the description as a CSP, at most $\binom{n}{2}$, i.e., $O(n^2)$ constraints have to be provided
- every (binary) constraint consists of at most $O(k^2)$ pairs

$\rightsquigarrow$ encoding size $O(n^2 k^2)$

$\rightsquigarrow$ the number of assignments is exponentially larger than the description of the CSP

Compact Encodings and General CSP Solvers

CSPs allow for compact encodings of large sets of assignments:

- consider a CSP with n variables with domains of size k

$\rightsquigarrow$ k^n assignments

- for the description as a CSP, at most $\binom{n}{2}$, i.e., $O(n^2)$ constraints have to be provided
- every (binary) constraint consists of at most $O(k^2)$ pairs

$\rightsquigarrow$ encoding size $O(n^2k^2)$

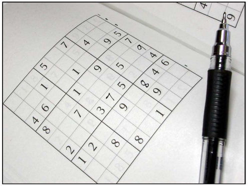
$\rightsquigarrow$ the number of assignments is exponentially larger than the description of the CSP

- as a consequence, such descriptions can be used as inputs of general constraint solvers

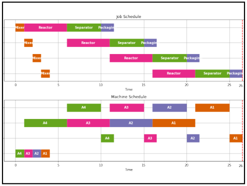
Examples

Constraint Satisfaction Problem Applications

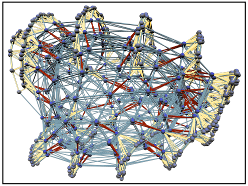
puzzles

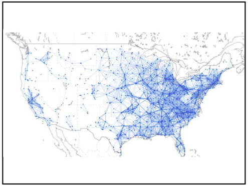


scheduling

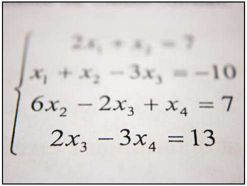


SAT solver

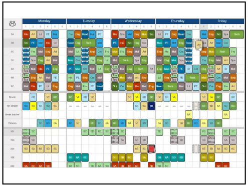




FCC spectrum auction



linear equations

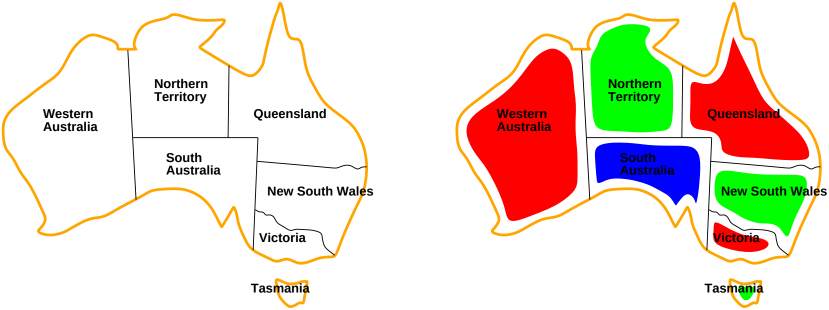


timetable generation

Example: Graph Coloring

Given a graph and $k \in \mathbb{N}$, how can we

- color the vertices using k colors
- such that two neighboring vertices never have the same color?

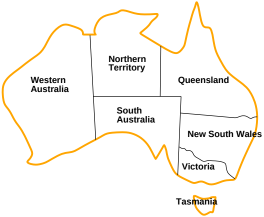


How to formalize this CSP?

Example: Graph Coloring

Given a graph and $k \in \mathbb{N}$, how can we

- color the vertices using k colors
- such that two neighboring vertices never have the same color?



- variables: $V = \{WA, NT, Q, NSW, VI, SA, T\}$
- domains: $dom(v) = \{r, g, b\}$ for all $v \in V$
- constraints: $\{c_{uv} \mid \text{for all connected } u, v \in V\}$, where

  $c_{uv} = \langle (u, v), \{(k, \ell) \in \{r, g, b\} \times \{r, g, b\} \mid k \neq \ell\} \rangle$

e.g., $c_{WA,NT} = \langle (WA, NT), \{(r, g), (r, b), (g, r), (g, b), (b, r), (b, g)\} \rangle$

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

Four Color Problem

famous problem in mathematics: Four Color Problem

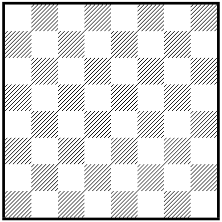
- Is it always possible to color a planar graph with 4 colors?
- conjectured by Francis Guthrie (1852)
- 1890 first proof that 5 colors suffice
- several wrong proofs surviving for over 10 years
- solved by Appel and Haken in 1976: 4 colors suffice
- Appel and Haken reduced the problem to 1936 cases,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

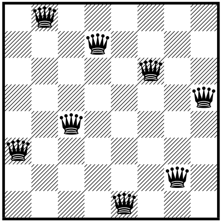
numberphile video:
`https://www.youtube.com/watch?v=NgbK43jB4rQ`

Example: 8 Queens Problem

How can we

- place 8 queens on a chess board
- such that no two queens threaten each other?



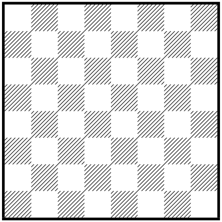


- originally proposed in 1848
- variants: board size; other pieces; higher dimension
- there are 92 solutions (12 non-symmetric ones)

Example: 8 Queens Problem

How can we

- place 8 queens on a chess board
- such that no two queens threaten each other?



- variables: $V = \{v_1, \ldots, v_8\}$
- domains: $dom(v) = \{1, \ldots, 8\}$ for all $v \in V$
- constraints: $\{c_{ij} \mid \text{for all } 1 \leq i < j \leq 8\}$, where

$$c_{i,j} = \langle (v_i, v_j), \{(k, \ell) \in \{1, \ldots, 8\} \times \{1, \ldots, 8\} \mid k \neq \ell \land |k - \ell| \neq |i - j|\}\rangle$$

e.g., $c_{1,3} = \langle (v_1, v_3), \{(1,2), (1,4), (1,5), (1,6), (1,7), (1,8),$
$(2,1), (2,3), (2,5), (2,6), (2,7), (2,8),$
$\ldots,$
$(8,1), (8,2), (8,3), (8,4), (8,5), (8,7)\}\rangle$

Example: Sudoku

How can we

- completely fill an already partially filled 9×9 matrix with numbers from {1,2,...,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

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

Example: Sudoku

How can we

- completely fill an already partially filled 9×9 matrix with numbers from {1,2,...,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

- variables: $V = \{v_{ij} \mid 1 \leq i, j \leq 9\}$
- domains: for all $v_{ij} \in V$
  - $dom(v_{ij}) = \{1, \ldots, 9\}$ if cell $\langle i, j \rangle$ is empty
  - $dom(v_{ij}) = \{k\}$ if cell $\langle i, j \rangle$ has predefined value k
- constraints:

  $C_{v_{ij}, v_{i'j'}} = \{(a, b) \in \{1, \ldots, 9\}^2 \mid a \neq b\}$
  for all $v_{ij}, v_{i'j'} \in V$ with
  - $i = i'$ (same row), or
  - $j = j'$ (same column), or
  - $\langle \lceil \frac{i}{3} \rceil, \lceil \frac{j}{3} \rceil \rangle = \langle \lceil \frac{i'}{3} \rceil, \lceil \frac{j'}{3} \rceil \rangle$ (same block)

Solutions

Assignments

Definition (assignment, partial assignment)

Let $\mathcal{C} = \langle V, dom, C \rangle$ be a CSP.

A partial assignment of $\mathcal{C}$ (or of V) is a function

$$\alpha : V' \to \bigcup_{v \in V} dom(v)$$

with $V' \subseteq V$ and $\alpha(v) \in dom(v)$ for all $v \in V'$.

If $V' = V$, then α is also called total assignment (or assignment).

- $\rightsquigarrow$ partial assignments assign values to some or to all variables
- $\rightsquigarrow$ (total) assignments are defined on all variables

Consistency

Definition (inconsistent, consistent, violated)

A partial assignment α of a CSP $\mathcal{C}$ is called inconsistent if there are variables u, v such that α is defined for both u and v, and there is $c_{u,v} \in C$ s.t. $(\alpha(u), \alpha(v)) \notin \mathrm{rel}(c_{u,v})$.

In this case, we say α violates the constraint $c_{u,v}$.

A partial assignment is called consistent if it is not inconsistent.

Solution

Definition (solution, solvable)

Let C be a CSP.

A consistent and total assignment of C is called a solution of C.

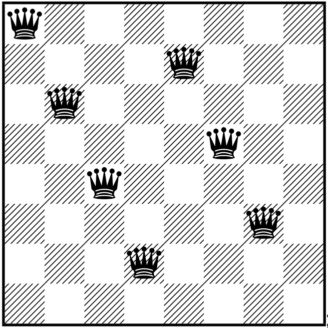
If a solution of C exists, C is called solvable.

If no solution exists, C is called inconsistent.

Consistency vs. Solvability

Note: Consistent partial assignments α cannot necessarily be extended to a solution.

It only means that so far (i.e., on the variables where α is defined) no constraint is violated.

;

$$\alpha = \{v_1 \mapsto 1, v_2 \mapsto 3, v_3 \mapsto 5, v_4 \mapsto 7, v_5 \mapsto 2, v_6 \mapsto 4, v_7 \mapsto 6\}$$

Complexity

Complexity of Constraint Satisfaction Problems

Proposition (CSPs are NP-complete)

Deciding whether a given CSP is solvable is NP-complete.

Proof

Membership in NP:

Guess and check: guess a solution and check it for validity.
This can be done in polynomial time in the size of the input.

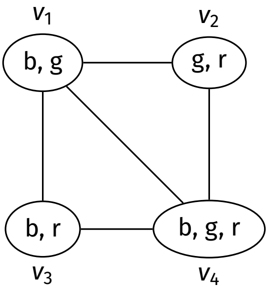
NP-hardness:

The graph coloring problem is a special case of CSPs
and is already known to be NP-complete.

Exercise

CSP Exercise

Consider a variant of the graph coloring problem where each vertex has a set of allowed colors.



1. Formalize the example as a binary constraint network.
2. Is the constraint network solvable? If yes, provide a solution of the constraint network. If not, justify your answer.
3. Provide a <u>minimal</u> consistent partial assignment that <u>cannot</u> be extended to a solution.
4. Provide an inconsistent partial assignment.

CSP Exercise – Solution

1. $\mathcal{C} = \langle V, dom, (R_{uv}) \rangle$ with
   - $V = \{v_1, \ldots, v_4\}$,
   - $dom(v_1) = \{b, g\}$, $dom(v_2) = \{g, r\}$, $dom(v_3) = \{b, r\}$, $dom(v_4) = \{b, g, r\}$, and
   - (binary) constraints:

$$\begin{aligned}
R_{v_1,v_2} &= \{\langle b, g\rangle, \langle b, r\rangle, \langle g, r\rangle\} \\
R_{v_1,v_3} &= \{\langle b, r\rangle, \langle g, b\rangle, \langle g, r\rangle\} \\
R_{v_1,v_4} &= \{\langle b, g\rangle, \langle b, r\rangle, \langle g, b\rangle, \langle g, r\rangle\} \\
R_{v_2,v_4} &= \{\langle g, b\rangle, \langle g, r\rangle, \langle r, b\rangle, \langle r, g\rangle\} \\
R_{v_3,v_4} &= \{\langle b, g\rangle, \langle b, r\rangle, \langle r, b\rangle, \langle r, g\rangle\}
\end{aligned}$$

2. Solvable. Solution: $\alpha_1 = \{v_1 \mapsto b, v_2 \mapsto r, v_3 \mapsto r, v_4 \mapsto g\}$

   (There is second solution: $\alpha_2 = \{v_1 \mapsto g, v_2 \mapsto r, v_3 \mapsto r, v_4 \mapsto b\}$)
3. $\alpha_1 = \{v_2 \mapsto g\}$ or $\alpha_2 = \{v_4 \mapsto r\}$ or $\alpha_3 = \{v_3 \mapsto b\}$
4. Inconsistent partial assignment: $\alpha = \{v_1 \mapsto b, v_3 \mapsto b\}$