Constraint Satisfaction Problem

Constraint Satisfaction Problem (CSP)

- A problem is solved when each variable has a value that satisfies all the constraints on the variable. A problem described this way is called a **constraint satisfaction problem**, or CSP.

- The main idea is to eliminate large portions of the search space all at once by identifying variable/value combinations that violate the constraints.

Definition of CSP

- Constraint satisfaction problem consists of three components, X, D, and C:
  - X is a set of variables, {X1,...,X,}.
  - D is a set of domains, {D,...,D,}, one for each variable.
  - C is a set of constraints that specify allowable combinations of values.

- A domain, D, consists of a set of allowable values, {v1,..., vn}, for variable X;.
  - For example, a Boolean variable would have the domain {true,false}.

- Each constraint C_j consists of a pair (scope, rel),
  - where scope is a tuple of variables that participate in the constraint
  - rel is a relation that defines the values that those variables can take on.

- A relation can be represented as an explicit set of all tuples of values that satisfy the constraint, or as a function that can compute whether a tuple is a member of the relation.

  - For example, if X; and X, both have the domain {1,2,3}, then the constraint saying that X1; must be greater than X2 can be written as,

((X1,X2),{(3,1),(3,2),(2,1)}) or as ((X1, X2), X1 > X2).

- CSPs deal with assignments of values to variables, {Xi = vi,Xj =vj,...}.

- An assignment that does not violate any constraints is called a **consistent or legal assignment**.

- A **complete assignment** is one in which every variable is assigned a value, and a **solution** to a CSP is a consistent, complete assignment.

- A **partial assignment** is one that leaves some variables unassigned, and a partial solution is a partial assignment that is **consistent**.

Example problem: Map coloring

- A map of Australia showing each of its states and territories.
- We are given the task of coloring each region either red, green, or blue in such a way that no two neighboring regions have the same color.

Example problem: Map coloring

- We define the variables to be the regions:

$$X = \{WA, NT, Q, NSW, V, SA, T\}.$$

- The domain of every variable is the set

$$Di = \{red, green, blue\}.$$

- The constraints require neighboring regions to have distinct colors.
- Since there are nine places where regions border, there are nine constraints:

$$C = \{SA \neq WA,\ SA \neq NT,\ SA \neq Q,\ SA \neq NSW,\ SA \neq V,$$
$$WA \neq NT,\ NT \neq Q,\ Q \neq NSW,\ NSW \neq V\}.$$

Example problem: Map coloring

- Here we arc using abbreviations; SA $\neq$ WA is a shortcut for ((SA,WA),SA $\neq$ WA),

- Where SA $\neq$ WA can be fully enumerated in turn as

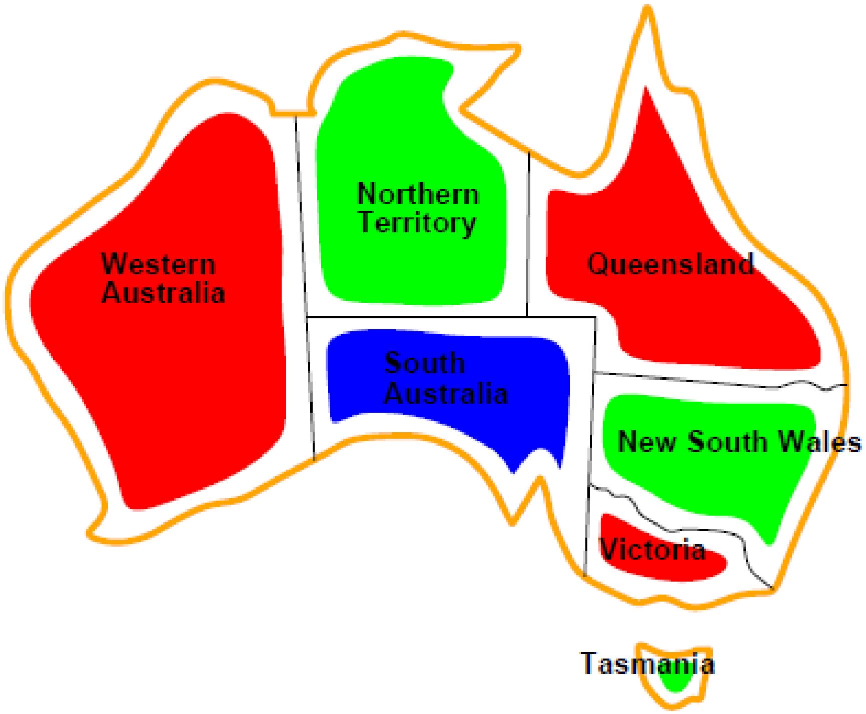
{(red, green), (red. blue), (green, red), (green, blue), (blue, red), (blue, green) }.

- Solutions to this problem?

Example problem: Map coloring

- There are many possible solutions to this problem, such as

{WA =red, NT=green, Q= red, NSW = green, V = red, SA=blue, T = red }.

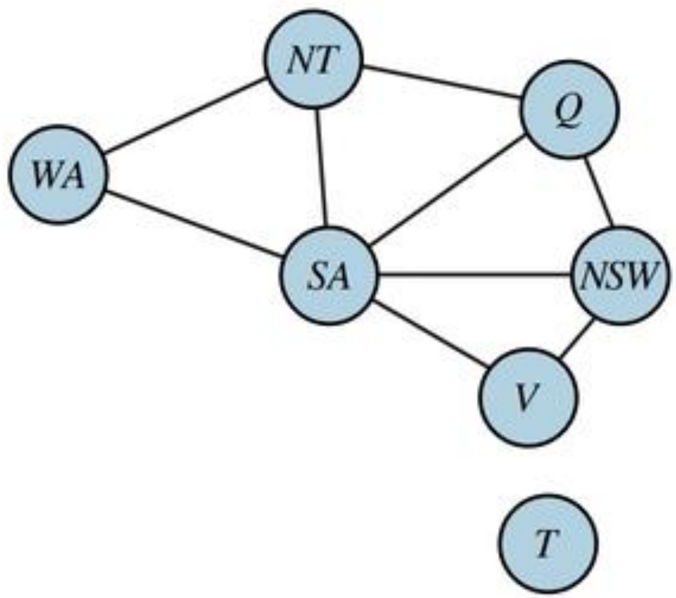


Solutions are assignments satisfying all constraints, e.g.,

$\{WA = red, NT = green, Q = red, NSW = green, V = red, SA = blue, T = green\}$

Example problem: Map coloring

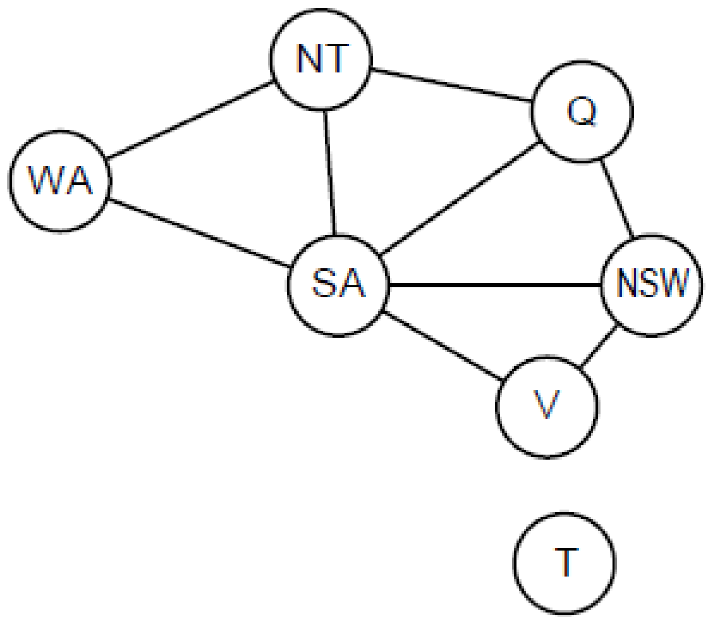
- It can be helpful to visualize a CSP as a constraint graph, as shown in Figure.
- The nodes of the graph correspond to variables of the problem, and an edge connects any two variables that participate in a constraint.



Constraint graph

Binary CSP: each constraint relates at most two variables

Constraint graph: nodes are variables, arcs show constraints



General-purpose CSP algorithms use the graph structure
to speed up search. E.g., Tasmania is an independent subproblem!

Types of CSP

Discrete variables

finite domains; size $d \Rightarrow O(d^n)$ complete assignments

- ◇ e.g., Boolean CSPs, incl. Boolean satisfiability (NP-complete)

infinite domains (integers, strings, etc.)

- ◇ e.g., job scheduling, variables are start/end days for each job
- ◇ need a constraint language, e.g., $StartJob_1 + 5 \leq StartJob_3$
- ◇ linear constraints solvable, nonlinear undecidable

Continuous variables

- ◇ e.g., start/end times for Hubble Telescope observations
- ◇ linear constraints solvable in poly time by LP methods