Heuristics

- 0 heuristic $\triangleq$ UCS
- Heuristics can be a `max` of multiple heuristics

Graph Search

- Tree search expands out all states including repeats but graph search does not
- Tree search but never expand a state twice
  - Make a list of things we've seen before
- Not always optimal (with inconsistent $h(x)$)

Consistency

- You must underestimate the cost of every action
- $h(A) - h(B) \leq \text{cost}(A, B)$
- Consistency $\implies$ admissibility

Constraint Satisfaction Problems (CSPs)

- Identification problem $\triangleq$ the goal is important, not the path
- Subset of search problems
- State defined by $X \coloneqq \{X_i \ \forall \ i\}$ with values from a domain D
  - D is sometimes dependent on i
- Goal test is a set of constraints specifying allowable combinations of values for subsets of variables
- eg: Coloring of states in Australia
  - Variables $\triangleq \{\text{WA}, \text{NT}, Q, \text{NSW}, V, \text{SA}, T\}$
  - $D \coloneqq \{\text{red}, \text{gren}, \text{blue}\}$
  - Constraints:
    * Implicit: $\text{WA} \neq \text{NT}$
    * Explicit: $(\text{WA}, \text{NT}) \in \{(\text{red}, \text{green}), (\text{red}, \text{blue}), ...\}$
  - Solutions $\triangleq \{\text{WA} \mathrel{\hat{=}} \text{red}, \text{NT} \mathrel{\hat{=}} \text{green}, Q \mathrel{\hat{=}} \text{red}, \text{NSW} \mathrel{\hat{=}} \text{green}, V \mathrel{\hat{=}} \text{red}, \text{SA} \mathrel{\hat{=}} \text{blue}, T \mathrel{\hat{=}} \text{green}\}$
- eg2: Queens chess
  - Variables $\triangleq \{X_{ij} \forall \{i, j\} \in N\}$
  - $D \coloneqq \{0, 1\}$
  - Constraints:

$$\left\{\begin{matrix} (X_{ij}, X_{ik}) \\ (X_{ij}, X_{kj}) \\ (X_{ij}, X_{(i+k)(j+k)}) \\ (X_{ij}, X_{(i+k)(j-k)}) \end{matrix} \quad \middle| \quad \{i, j, k\} \in N \right\} \in \{(0,0), (0,1), (1,0)\}$$

$$\sum_{i,j} X_{ij} = N$$

- Better approach:
- Variables $\triangleq \{Q_k \forall k \in N\}$
- $D \coloneqq 1, 2, 3, ..., N$
- Constraints $\triangleq \forall i, j \mid Q_i$ doesnt threaten Q_j

Constraint Graph

- Graph where edges are constraints and nodes are the variables
- We make intermediary nodes (depicted as squares) if a constraint applies to multiple nodes
  - The square then becomes the constraint

Solving

- CSVs are a type of search problem, so can use search functions
  - Generally not efficient because there isn't a concept of the rules
- Backtracking Search
  - Type of DFS
  - Evaluate one variable at a time (fixed ordering)
    * Tree expands only possible evaluations of one variable instead of all possibilities of all variables
  - Check constraints as you go (consider only values which do not conflict with previous assignments)

Filtering

- Filtering := keep track of domains for unassigned variables and cross off bad options
- Forward checking := cross off values that violate a constraint when added to the existing assignment
- Reason about consequences of actions