CSPs

Constraint Satisfaction Problems (CSPs)

- Assigning valid values to a set of variables
  - “Valid” → all constraints are met
- “Identification” problem, not search problem
  - Don’t care about path to goal
  - Caveat: this is just a naming technicality; we often formulate CSPs as search problems

Formulation

- **Variables:** X_1, ..., X_N, each able to take on one single value
- **Domain:** set of all possible values a variable can take on, $\{x_1, ..., x_d\}$
- **Constraints:** restrictions on which values a variable can take
  - Can be restrictions on a single variable (unary constraint)
  - Can be restrictions on relationships between multiple variables (binary / higher-order)

How to solve?

1. **Backtracking:**
    a. Pick an arbitrary variable order (ex. A → B → C)
    b. At each iteration, pick the next unassigned variable and attempt to assign it a value
    c. If no values exist that don't violate constraints, backtrack previously assigned variables (i.e. try new assignments)
    d. Akin to DFS

Performance Improvements

- **Filtering:** pruning domains of unassigned variables ahead of time
  - **Forward checking:** whenever X is assigned, prune domains of all variables Y that have constraints with X
  - **Arc consistency:** make "arcs" (two directed edges) between any two variables that have constraints with each other
    - For each arc Y → Z, if an assignment Y = *y* results in no possible assignments for Z, prune *y* from Y's domain
    - IF we prune something from Y, add to the queue all arcs of the form X → Y (if needed)
    - IF pruning something from Y leaves an empty domain, backtrack

Performance Improvements

- **Heuristics:** determining optimal order of assignment
  - **Minimum remaining values (MRV):** choose variable with smallest domain to assign next
  - **Least constraining value (LCV):** choose value that will result in fewest prunes
  - **Min-conflicts:** choose value that violates the fewest constraints