

Graph Search

Problem defined by: **states** S and **initial state**, **actions** A , **state transition model** $T(s_1 \in S, a \in A) \mapsto s_2 \in S$, **goal test**, **cost function** $C(a): A \mapsto \mathbb{R}$.

Consistency: $\forall v_i, v_j, h(v_i) \leq h(v_j) + c(v_i, v_j)$ where v_j is a successor of v_i and $c(v_i, v_j)$ is the cost of moving from v_i to v_j , i.e. for each step we move backwards, the value of the heuristic at the further node \leq the value at the closer node plus the cost of the step.

Admissibility: $\forall v_i, h(v_i) \leq C^*(v_i) = g(w) - g(v_i)$ where $C^*(v)$ is the true optimal cost to reach the goal from v , i.e. the value of the heuristic is always less than or equal to the true cost of reaching the goal.

A^* is optimal for consistent h , or admissible h with tree search. Consistency and $h(w) = 0 \implies$ admissibility. Admissible heuristics can be created by relaxing problem constraints.

For cycle checking, we keep track of all previously seen nodes (expanded or in frontier) *and* their cost. When adding a new node to the frontier, if it matches a previously seen node and has higher (or equal) cost, don't add it. If it matches but has lower cost, add it and prune the higher cost node from the frontier. Note for A^* the cost is taken as $g(n)$ and not $f(n)$.

Local Search

Problem defined by: **states** S , **neighbours** $N(s \in S)$, **value** $\text{Val}(s)$ (lower values more desirable)

Cooling schedule $T(t)$ is typically decreasing, application dependent. Simulated annealing terminates when $T = 0$ or $\text{Val}(C) = 0$. Both local search algorithms are *incomplete*.

Constraint Satisfaction

Problem defined by: **variables** $X = \{x_1, x_2, \dots, x_n\}$, **domains** $D = \{D_{x_1}, D_{x_2}, \dots, D_{x_n}\}$, **constraints** C relating to domains. NP-Hard.

Default inference is expensive. Methods to limit depth: **forward checking:** only check the effect of assigning the current variable (don't add x to queue); or add to queue only if the variable has fewer than N valid values left, $|T| \leq N$.

For Infer, only operate on the variables that haven't been assigned. Heuristics: **Minimum Remaining Value:** PickUnassignedVar choose the unassigned variable with the smallest effective domain size (allows quick backtracking); **Least Constraining Value:** OrderDomainValue choose the value that rules out the fewest domain values for other variables (so we maximize the chances of success).

Variants: boolean satisfiability (B-SAT), where $D_{x_i} = \{0, 1\}$; binary CSP, where every constraint $C(x_i, x_j)$ is over 2 variables. 2-SAT: combination of the two, is the only one with polynomial time solutions.