

Aritificial Intelligence Cheat Sheet Cheat Sheet by [deleted] via cheatography.com/31421/cs/9528/

Search Methods	
Tree Search	Expand nodes using gradients
Graph Search	Avoids revisiting nodes and ensure efficiency

Graph Search	Avoids revisiting nodes and ensure efficiency
Uninformed sea	rch
Uniform cost search	aka Cheapest-first Add visited node to Explored and add its neighbors to the frontier Visit cheapest node in the frontier Move to next cheapest if all neighbors are
	explored
Iterative deepening	Iteratively calls depth-limited search Initialize frontier with root If not goal, remove from frontier and expand If at the end of depth, terminate and start over Repeat until goal is found Guaranteed to find optimal path More efficient than DFS Time complexity: O(b^d) Space complexity: O(d)
Bidirectional	Finds shortest path of a grpah

Informed Search	
Best-first search	Choose unvisited node with the best heuristic value to visit next Same as lowest cost BFS
Greedy best-first search	Uses heuristic to get the node closest to the goal Bad performance if heuristic is bad Does NOT consider cost of getting to the node

Informed Search (cont)	
A* search	Always expand to node with minimum f Evaluate cost of getting to goal using heuristics $f(n) = g(n) + h(n) \text{ where g is cost to get to n}$ Uses priority queue
Heuristics	Cost to get to the goal
Admissible herustic	Optimistic model for estimating cost to reach the goal Never overestimates $h(n) \le c(n)$ where c is actual cost
Consistent heuristic	$h(n) \le c(n, a, n') + h(n')$ Immediate path costs less than longer path Consistent \Longrightarrow Admissible

Consistent	$\Pi(\Pi) = G(\Pi, a, \Pi) + \Pi(\Pi)$
heuristic	Immediate path costs less than longer path
	Consistent → Admissible
Consistent h	euristic
	Consistent heuristics • A heuristic is consistent if for every node n , every successor n' of n generated by any action a , $h(n) \leq c(n,a,n') + h(n')$ • If h is consistent, we have $f(n') = g(n) + h(n')$ $= g(n) + h(n)$ $= g(n) + h(n)$ $= f(n)$ • i.e., $f(n)$ is non-decreasing along any path. • Theorem: If $h(n)$ is consistent, A^* using GRAPH-SEARCH is optimal

Adversarial Search	
Hill climbing	Method of local search Only move to neighbors to find the max Does NOT guarantee to find optimal
Simulated annealing	Method of local search Combine hill climbing and random walk Always find global max



By [deleted]

cheatography.com/deleted-31421/ Published 19th October, 2016. Last updated 19th October, 2016. Page 1 of 2.

Yours! https://apollopad.com

Sponsored by ApolloPad.com

Everyone has a novel in them. Finish



Aritificial Intelligence Cheat Sheet Cheat Sheet by [deleted] via cheatography.com/31421/cs/9528/

SAT

Adversarial Search (cont) Local beam Generate k random states Generate successors of all k states If goal stop; else, pick k best successors and repeat Different from hill-climbing since information is shared between k points Genetic Cross-Over and mutation algorithm Decomposes strands of DNA and permute

Produces children by: Selection, Cross-over, Mutation

Propositional SAT: Graph coloring At lest 1 of K per i (Ci,1 ∨ Ci,2 ∨ ... ∨ Ci,k) O(n) clauses 1 ≥ color per i ∀ k≠k' (¬Ci,k ∨ ¬Ci,k') O(n^2) If node i and j share an edge ∀ x∈k, (¬Ci,x ∨ ¬Cj, x) assign different colors

α-β Pruning		
function alphabeta(node, depth, α , β ,		
maximizingPlayer)		
if depth = or node is a terminal node		
return the heuristic value of node		
if maximi zin gPlayer		
∨ := -∞		
for each child of node		
v := max(v, alphab eta (child,		
depth - 1, α , β , FALSE))		
$\alpha := \max(\alpha, v)$		
if $\beta \leq \alpha$		
β cutbreak)(
return v		
else		
∀ := ∞		
for each child of node		
v := min(v, alphab eta (child,		
depth - 1, α , β , TRUE))		
$\beta := \min(\beta, v)$		
if $\beta \leq \alpha$		
α cuthpeak)(
return v		



By [deleted]

31421/

cheatography.com/deleted-