

Ch 4 – artificial intelligence illuminated



problem solving consists of a goal and set of actions that can be taken to lead to the goal.

state of search space represents where you have reached in problem solving

initial state the starting state when solving a problem

goal state the final state that satisfies a goal of a certain problem that was trying to solve

search space problem space

search methods methods of searching nodes

data-driven search starts from initial state and uses actions that are allowed to move forward until a goal is reached (forward chaining)

goal-driven search search starts at goal and works back to a start state by seeing what moves could have led to goal state. (backward chaining)

generate and test search that generates each node in search space and tests it to see if it is a goal node. (simple brute-force method)

exhaustive search brute force search where every node in tree is examined

blind search another term for generate and test search

generator (generate and test) the "spawner" of nodes. should satisfy three properties. 1. must be complete (every possible solution). 2. must be nonredundant 3. must be well informed (should only propose suitable solutions)

depth-first search follows each path to greatest depth before moving to next path. See any algorithm book on implementation. secret is you add nodes to a stack (last in is first out) start from left and work right, goes to leaf, then backtracks a level.

breadth-first search examines all nodes on same level before moving to next level. secret is you add nodes to the front of the queue and remove from the back of queue.

ply all nodes on a level

useful search method properties to consider complexity, completeness, optimality, admissibility, irrevocability

complexity (search methods) how efficient the method is over time and space. time complexity - how long it takes. space complexity - amount of memory search method takes

big-o notation describes complexity of a method $O(x)$ etc.

completeness (search methods) search method is described as complete if guaranteed to find a goal state if one exists. Breadth-first is complete, depth-first search is not complete

optimal (search methods) search method is optimal if it is guaranteed to find the best solution that exists. will take the least number of possible steps to reach goal.

tentative (search methods) methods that use backtracking.

irrevocable (search methods) methods that do not use backtracking (use only one path). often fooled by local optima

humans use depth-first search easy to implement and relates closely to natural way humans search for things

traversing a maze (depth-first) start with hand on left side of maze or right and follow maze around until reach end. this corresponds exactly to depth-first search

depth first search implementation stack. push successors to stack. pop, push successors, pop, etc. if no successors then pop next one repeat until goal found

breadth first search implementation queue. add successors to queue. remove first, add successors, because queue adding successors to back of list rather than front as in stack

optimality of breadth-first vs. depth-first depth-first search is neither optimal nor complete. breadth-first search is both. depth first might never find solution, breadth-first will always find best solution

depth-first iterative deepening (DFID) depth-first search combined with breadth-first search. repeatedly carry out depth-first searches limited to level (1, 2, ..., n). wasteful for small tree, works well for large tree

heuristic (redoux) humans use them to solve problems. Can reduce a hard problem into a relatively simple one.

heuristic evaluation function when applied to a node gives a value that represents a good estimate of the node from the goal. two nodes, m and n, if $f(m) < f(n)$ then it should be that m is more likely to be on an optimal path than n

heuristic search methods, or heuristically informed search methods provides an estimate of the distance from any given node to a goal node.

heuristic is informed if... it uses additional information about nodes that have not yet been explored to decide which nodes to examine next

heuristic is uninformed if... it is not informed or blind

heuristic is more informed than another heuristic if... if $h(\text{node}) \leq j(\text{node})$ for all nodes in the search space. the more informed a search method is the more efficiently it will search

monotone (search method) if it always reaches a given node by shortest path

monotonic heuristic a heuristic that is monotone

admissible heuristic a heuristic that never overestimates the true distance of a node from the goal. a monotonic heuristic is heuristic

hill climbing informed search. check height is higher than current position, move to that location and restart the algorithm. if all directions lead lower than current position, stop and assume summit. - uses stack

steepest ascent hill climbing same as hill climbing, but rather than choosing first higher value, choose the highest out of all possible values (directions) - uses stack, you sort successors before placing on stack

local maxima problem (hill climbing) hill climbing can be fooled by foothills, plateaus, and ridges

best-first search heuristic similar to hill climbing. entire queue is sorted after new paths have been added to it rather than sorting new paths before adding

beam search form of breadth-first that uses heuristic.

optimal path optimal path is one that has lowest cost or involves traveling the shortest distance from start to goal node

British Museum procedure examine every single path through search tree and returning via the best path that was found

A* algorithm similar to best-first but uses more complex heuristic

ONCH 4 - ARTIFICIAL INTELLIGENCE ILLUMINATED SPECIFICALLY FOR YOU FOR ONLY \$13.90/PAGE Order Now