

1. When OR-SEARCH cycles back to a state on path it returns a token loop which means to loop back to the most recent time this state was reached along the path to it. Since path is implicitly stored in the returned plan, there is sufficient information for later processing, or a modified implementation, to replace these with labels.

The plan representation is implicitly augmented to keep track of whether the plan is cyclic (i.e., contains a loop) so that OR-SEARCH can prefer acyclic solutions.

AND-SEARCH returns failure if all branches lead directly to a loop, as in this case the plan will always loop forever. This is the only case it needs to check as if all branches in a finite plan loop there must be some And-node whose children all immediately loop.

2. A sequence of actions is a solution to a belief state problem if it takes every initial physical state to a goal state. We can relax this problem by requiring it take only some initial physical state to a goal state. To make this well defined, we’ll require that it finds a solution for the physical state with the most costly solution. If h∗(s) is the optimal cost of solution starting from the physical state s, then



is the heuristic estimate given by this relaxed problem. This heuristic assumes any solution to the most difficult state the agent things possible will solve all states.

On the sensorless vacuum cleaner problem, h correctly determines the optimal cost for all states except the central three states (those reached by [suck], [suck, left] and [suck, right]) and the root, for which h estimates to be 1 unit cheaper than they really are. This means A∗ will expand these three central nodes, before marching towards the solution.

3.

(a) An action sequence is a solution for belief state b if performing it starting in any states ∈ b reaches a goal state. Since any state in a subset of b is in b, the result is immediate.

Any action sequence which is not a solution for belief state b is also not a solution for any superset; this is the contrapositive of what we’ve just proved. One cannot, in general, say anything about arbitrary supersets, as the action sequence need not lead to a goal on the states outside of b. One can say, for example, that if an action sequence solves a belief state b and a belief state b′ then it solves the union belief state b ∪ b′.

(b) On expansion of a node, do not add to the frontier any child belief state which is a  
superset of a previously explored belief state.

(c) If you keep a record of previously solved belief states, add a check to the start of OR search to check whether the belief state passed in is a subset of a previously solved belief state, returning the previous solution in case it is.

4. Consider a very simple example: an initial belief state {S1, S2}, actions a and b both leading to goal state G from either initial state, and  
c(S1, a, G) = 3 ; c(S2, a, G) = 5 ;  
c(S1, b, G) = 2 ; c(S2, b, G) = 6 .

In this case, the solution [a] costs 3 or 5, the solution [b] costs 2 or 6. Neither is “optimal” in any obvious sense.

The cost for [a] is {S1:3, S2:5} and the cost for [b] is {S1:2, S2:6}. We can say that plan p1 *weakly dominates* p2 if, for each initial state, the cost for p1 is no higher than the cost for p2. (Moreover, p1 *dominates* p2 if it weakly dominates it *and* has a lower cost for some state.) If a plan p weakly dominates all others, it is optimal. Notice that this definition reduces to ordinary optimality in the observable case where every belief state is a singleton. As the preceding example shows, however, a problem may have no optimal solution in this sense. A perhaps acceptable version of A∗ would be one that returns any solution that is not dominated by another.

In particular, if we define the cost of a plan in belief-state space as the minimum cost of any physical realization, we violate Bellman’s principle. Modifying and extending the previous example, suppose that a and b reach S3 from S1 and S4 from S2, and then reach G from there:

c(S1, a, S3) = 6 ; c(S2, a, S4) = 2 ;

c(S1, b, S3) = 6 ; c(S2, b, S4) = 1 .c(S3, a, G) = 2 ; c(S4, a, G) = 2 ;

c(S3, b, G) = 1 ; c(S4, b, G) = 9 .

In the belief state {S3, S4}, the minimum cost of [a] is min{2, 2} = 2 and the minimum cost of [b] is min{1, 9} = 1, so the optimal plan is [b]. In the initial belief state {S1, S2}, the four possible plans have the following costs:

[a, a] : min{8, 4} = 4 ; [a, b] : min{7, 11} = 7 ; [b, a] : min{8, 3} = 3 ; [b, b] : min{7, 10} = 7 .

Hence, the optimal plan in {S1, S2} is [b, a], which does not choose b in {S3, S4} even though that is the optimal plan at that point. This counterintuitive behavior is a direct consequence of choosing the minimum of the possible path costs as the performance measure.

5. No solution is possible because no path leads to a belief state all of whose elements satisfy the goal. If the problem is fully observable, the agent reaches a goal state by executing a sequence such that Suck is performed only in a dirty square. This ensures deterministic behavior and every state is obviously solvable.

6.

(a) Online search is equivalent to offline search in belief-state space where each action in a belief-state can have multiple successor belief-states: one for each percept the agent could observe after the action. A successor belief-state is constructed by taking the previous belief-state, itself a set of states, replacing each state in this belief-state by the successor state under the action, and removing all successor states which are inconsistent with the percept. The initial belief state has 210 = 1024 states in it, as we know whether two edges have walls or not but nothing more. There are  possible belief states, one for each set of environment configurations.

After each action and percept, the agent learns whether or not an internal wall exists between the current square and each neighboring square. Hence, each reachable belief state can be represented exactly by a list of status values (present, absent, unknown) for each wall separately. That is, the belief state is completely decomposable and there are exactly 312 reachable belief states. The maximum number of possible wall-percepts in each state is 16, so each belief state has four actions, each with up to 16 nondeterministic successors.

(b) Assuming the external walls are known, there are two internal walls and hence 4 possible percepts.

(c) The initial null action leads to four possible belief states, from each belief state, the agent chooses a single action which can lead to up to 8 belief states, given the possibility of having to retrace its steps at a dead end, the agent can explore the entire maze in no more than 18 steps, so the complete plan has no more than 818 nodes.

7. Since we can observe successor states, we always know how to backtrack from to a previous state. This means we can adapt iterative deepening search to solve this problem. The only difference is backtracking must be explicit, following the action which the agent can see leads to the previous state.

The algorithm expands the following nodes:

Depth 1: (0, 0), (1, 0), (0, 0), (−1, 0), (0, 0)

Depth 2: (0, 1), (0, 0), (0, −1), (0, 0), (1, 0), (2, 0), (1, 0), (0, 0), (1, 0), (1, 1), (1, 0), (1, −1)