Juan Barrera

Professor Mark Voortman

Introduction to AI

Module 2 Assignment (Part B)

Sep 22 20205

Questions:

How local search works and what are the benefits over the search methods:

The local search algorithm starts from one single current state and explores its neighbors, instead of systematically constructing the full search tree as in the case of breadth-first, depth-first, or A*. An evaluation function enables the algorithm to determine a good move: it might shift to a neighboring state or stay in the current state, making the algorithm highly space efficient since only the current state must be stored in memory and not all paths. This technique is especially useful when the size of the state space is vast or even infinite, where the original search quickly becomes impractical. Another advantage is that local search does not need a well defined goal state; instead, it can optimize for the best solution where one wants to maximize or minimize an objective function. Hill climbing moves to the neighbor with the best improvement but risks being stalled at local optima; simulated annealing allows occasional "bad" moves to escape local optima while reducing randomness; and genetic algorithms work with a population of states where they undergo crossover and mutation operations to evolve better solutions over time. These ways provide local search with possibilities for optimization such as scheduling, design, or machine learning.

3 different local search approaches

Hill climbing makes improvements by selecting the best neighbors to move to, provided this movement stops in a local optimum. Therefore, it is okay for simulated annealing to allow the occurrence of downhill movement for escaping bad local solutions and for a broad exploration whereby the downhill movement has a decreasing probability over time. Genetic algorithms are very dominant, where candidates in the population evolve while undergoing incidents, based on crossover and mutation, working towards refining and finding promising solutions. In all, they are space-efficient and scale well to large or continuous state spaces and are especially useful in optimization problems where traditional search methods become impractical.

1.

Each world is a particular pattern of which of the internal adjacency edges are open or blocked. The agent's initial belief state is the set of all worlds consistent with no knowledge of the internal walls. Searching offline in belief-state space means planning a policy that maps belief states to actions so that for every world in the initial belief the policy reaches the goal.

How many worlds? Then in the 3x3 grid there are 12 internal adjacency edges (6 horizontal + 6 vertical and each one of this edges can be open or blocked, so the number of possible worlds is 2^12 = 4096. Thus the initial belief contains 4096 worlds.

How large is the belief-state space?

Power-set bound (every belief is any subset of worlds): 2^4096 (i.e. 2^(2^12)) which is astronomically huge.

Compact representation bound: if for each of the 12 edges we store "known open / known blocked / unknown" (3 possibilities) and also the agent's current cell (9 possibilities), an upper bound is 9 * 3^12 = 9 * 531,441 = 4,782,969. This is much smaller than 2^4096 but still very large.

2.

The agent starts at corner (1,1). The only possible local moves (within the grid) are Up and Right; each may be legal or blocked. So the possible sets of legal actions the agent can perceive at the start are:
{} (both blocked), {Up}, {Right}, {Up,Right}.

Therefore there are 4 distinct percepts possible at the initial state (ignoring the trivial "visited?" bit which is simply "not yet visited").

3.

First steps (decision tree top): pick an action to try from (1,1), say Try Right. Two immediate perceptual outcomes/branches:

Right succeeds (legal) -> agent moves to (2,1); follow the subplan from (2,1).

Right blocked -> agent remains at (1,1) and learns that right-edge is blocked; branch to an alternative, e.g. Try Up. That splits into Up succeeds (go to (1,2)) or Up blocked (both blocked at start).

Continue expanding each branch similarly: every attempted action yields different percepts (success vs blocked), producing further branches; when you move to a new cell you branch on that cell's percept set.

Size of the complete contingency plan (rough magnitude):

Lower bound: must distinguish every possible world that leads to different required behaviors, so at least one leaf per world, which is at least 2^12 = 4096 leaves.

Compact upper bound (edge-knowledge representation): could require on the order of up to about 9 * 3^12 which is about 4.8 million belief nodes in the worst case.

Absolute worst (power-set beliefs): up to 2^4096 belief nodes, completely infeasible.