Uninformed Search

Depth First Search

A depth first search (DFS) will begin at the top of a tree and explore as deep as possible within each branch before moving on to the next branch. This search will move from left to right across a search tree and could potentially process the entire tree to find a solution.

Breadth First Search

A breadth first search (BFS) will begin at the top of the tree and explore from left to right, processing all the nodes above the shallowest solution. It will explore all of the nodes of the tree at the current depth before searching the next depth level.

Uniform Cost Search

Uniform Cost Search (UCS) considers the cost associated with exploring the different nodes that it visits. It expands the cheapest node first, and then moves on to find the solution with the lowest cumulative cost. It does this while not having knowledge of where the goal location is.

Informed Search

Heuristics

Heuristics are functions that will try to determine how close a given state is to the goal. These are generally crafted around a specific application.

Greedy Search

A greedy search is a heuristic where the node that seems the closest to the goal is expanded and searched first. The heuristic applied here would be to estimate the distance to the goal for a given state.

A* Search

An A* search combines elements of UCS and Greedy search. When calculating a move, the UCS cost of a path is combined with the Greedy search proximity to the goal. This will move towards the goal more readily than the UCS but will also consider the cost of making those moves more than the Greedy search.

Adversarial Search

Minimax

The minimax algorithm is used to make decisions by finding an optimal move in a given situation. There are 2 agents involved, the minimizer and the maximizer. The maximizer will try to find the highest result, while the minimizer tries to find the opposite.

Alpha-Beta Pruning

Alpha-Beta pruning is based on the Minimax algorithm. It expands upon it by eliminating moves that are demonstrated to be worse than a move that has previously been examined. These moves are pruned, which results in not needing to use resources on exploring branches that cannot be in the solution.

Expectimax Search

Expectimax is also like the Minimax algorithm, but instead of using a minimizer, the expected outcomes are calculated and averaged. By not using the worst-case scenario outcomes with the minimizer, the average-case is now being returned. This is useful when an opponent is acting randomly, unlike minimax which assumes the opponent is behaving optimally.

Markov Decision Process

A Markov Decision Process is a tuple containing 4 elements: A set of states, a set of actions, the probability that an action in a state will lead to another state, and the reward for moving from the first state to the next.

Epsilon Greedy

Epsilon greedy is a method for choosing between different options when probability distributions for those options are unknown. A player would choose which option appears to have the highest payout, which is the greedy choice. The epsilon choice would be to choose a different option some of the time to continue exploring all of the available options.

Cross Entropy Method

The cross-entropy method is a two-step procedure that is iterated through repeatedly. First a random data sample is selected. Then, the parameters of the mechanism whereby the data is selected are update so that a better sample is picked next.