A* Variants

https://movingai.com/astar-var.html

U of Alberta

A* Variants

There are many simple variants of the A* algorithm. A*'s performance can be varied by changing the heuristic used, but many other variants arise when changing the weighting on the g-cost and h-cost of a node. Suppose we are sorting nodes in the OPEN list by:

$$f = w1 * g + w2 * h$$

By varying w1 and w2 we can get different behavior. This page has videos for the following variants:

- w1 = 1; w2 = 0: Dijkstra's Algorithm. This is traditional graph search without a heuristic
- w1 = 0; w2 = 1: Pure Heuristic Search. This is also known as greedy best-first search. It focuses search efforts on states with low heuristic values.
- w1 = 1; w2 = 1: A*. This is the regular A* algorithm.
- w1 = 1; w2 = 10: Weighted A*. Weighted A* gives up optimality to try to find a solution more quickly.
- w1 = 1; w2 = 1; differential heuristic: The proceeding videos use the octile distance heuristic. This video uses the max of 10 differential heuristics at each state.

Note that these videos speed up over time. If not, some algorithms would take quite a long time to run.

Dijkstra's Algorithm

Dijkstra's algorithm searches uniformly without regard to the heuristic. Thus, when the search starts, you can see it moves outward from the start state in a uniform circle. Dijkstra's algorithm must expand almost every state in the entire map to solve this problem. Dijkstra's algorithm is guaranteed to find the optimal path.

Pure Heuristic Search

Pure heuristic search only looks at heuristic values, and the heuristic values start relatively low, so the search effort first focuses on states near the goal. It eventually makes its way around to the goal. In this example Pure Heuristic Search expands the same states as A*, it just expands them in a different order. Additionally, Pure Heuristic Search will likely not find the optimal path to the goal.

A*

A* is guaranteed to find the optimal path to the goal. Given a consistent heuristic, it is also optimal, in that another algorithm given the same information (heuristic and successor function) cannot do better than A*. In practice many approaches outperform A* because they use better heuristics, do not return optimal solutions, or take advantage of other state-space specific properties.

Weighted A*

Weighted A* focuses its search effort on states with low heuristic values, but does not completely ignore the g-costs like Pure Heuristic Search does. As a result, Weighted A* begins with a much "flatter" search than other algorithms. Once the search proceeds around the map, it quickly finds the goal. Weighted A* does not find optimal paths.

A* with Differential Heuristics

Differential heuristics (also called ALT) can be combined to greatly reduce the work performed by A*. In this example we see that far fewer states are expanded, and that the search is fairly well focused from the start to the goal. The cost is that the heuristics have a larger overhead and may not be valid if the map changes.

Downloads

Dijkstra
Pure Heuristic Search
A*
Weighted A* (weight = 10)
A* with 10 differential heurisitcs