Tutorials

Graph Searching

Tutorial Four: Searching Options

-Algorithm Options:

There are six different algorithms that can be used to solve a graph.

You can change what algorithm to select by clicking on the algorithm of you choice in the 'Graph Options' menu. They differ in how they select the next node from the frontier. Below is a brief description of the algorithms.

Uninformed - do not use any node heuristic value or edge cost information:

- **Depth-first search:** tries to search paths to their completion before trying new ones. When a path fails, the algorithm backtracks to the last point at which it could have chosen a different path, and then tries that one. In other words, when a node is expanded, the next node to be chosen is the first child of the previous node.
- **Breadth-first search:** treats the frontier like a queue, first-in-first-out. It chooses the leftmost element in the frontier. If that element is not the goal node, its children are placed at the end of the queue and then the next leftmost element is checked.
- **User defined:** user chooses whichever node of their choice on the frontier.

Use edge costs:

- **Lowest cost search:** chooses the element of the frontier with the lowest cost. It treats the frontier like a priority queue, ordered by g(n), which is a function that gives the cost of the path from the start node to the goal node.

Use node heuristic values:

- **Best first search:** chooses the element of the frontier with the lowest value of h(n), since it tries to choose the element that appears to be closest to the goal. It treats the frontier like a priority queue, ordered by the heuristic function.
- **Heuristic depth search:** a version of depth first search that uses heuristic knowledge to guide the search. It makes the local best choice, searching the most promising child of the previous node, instead of the leftmost child. Which child is most promising is determined by the value of h(n).

Use both node heuristic values and edge costs:

- **A*:** for each path on the frontier, this search uses an estimate, f(n), of the total path length from the start node to the goal node by adding the length of the path from the start node to a frontier node, g(n), plus the heuristic value of the node, h(n). Then the node with the lowest f(n) is chosen from the frontier.

-Pruning Options:

There are three different methods that you can choose when there are cycles in a graph.

You can change the pruning options by clicking on the option of your choice in the 'Graph Options' menu. Below is a description of the different options:

- **None:** no checking.
- **Multiple-Path pruning:** if the node is already on the path from the start node to that node, check to see if the path to the goal node is shorter and choose the shortest path.
- **Loop detection:** prune a node that already appears on the path from the start node to that node.