

A* Project Write-Up

Ben Cohen and Ben Brittain

March 28, 2013

Problem Definition

We define this search problem as attempting to find the optimal path to roll a die through a maze from a pre-defined start to a pre-defined end.

States

We define a state description as a representation of the board with indication of where the start node is, the end node is, and where the die is, as well as information about the orientation of the die. For simplicity, our program outputs state descriptions as a $n \times m$ grid of "."s (empty passable nodes), "*"s (empty, unpassable nodes), one "S" signifying the start state, one "G" signifying the goal state, and one number, signifying the die, as well as the number currently on top of it.

Initial State

Our initial state is marked with an "S" in the maze.

Actions

Our set of possible actions is simply defined as rotating our die 90 degrees in each of the cardinal directions.

Transition Function

Our transition function is defined as follows: $\text{board}[\text{row}][\text{col}] \times \text{rollEast}() = \text{board}[\text{row}][\text{col}+1]$ (provided col is not equal to the number of columns in the board [ie. the rightmost row], in which case rollEast() is an invalid action)

$\text{board}[\text{row}][\text{col}] \times \text{rollNorth}() = \text{board}[\text{row}+1][\text{col}]$ (provided row is not equal to zero [ie. the top row], in which case rollNorth() is an invalid action)

$\text{board}[\text{row}][\text{col}] \times \text{rollSouth}() = \text{board}[\text{row}-1][\text{col}]$ (provided row is not equal to the number of rows in the board [ie. the bottom row], in which case rollSouth() is an invalid action)

$\text{board}[\text{row}][\text{col}] \times \text{rollWest}() = \text{board}[\text{row}][\text{col}-1]$ (provided col is not equal to zero [ie. the leftmost row], in which case rollWest() is an invalid action)

Goal State

Our Goal state is simply defined as the node marked with a "G" on our board.

Heuristics

For our project we implemented three different heuristics: Manhattan Distance, Direct Cost, and Diagonal Manhattan Distance. Below are descriptions of, as well as arguments for the validity of these three heuristics.

Manhattan Distance

The Manhattan distance is simply the minimum number of moves from the current state to the goal state, disregarding all rules regarding the orientation, as well as walls. This number is guaranteed to be less than or equal to the actual minimum number of moves (ie. following the rules). Consider the below cases:

Case 1: The moves considered without paying attention to the rules are the same moves that would be made while considering the rules. In this case, the two numbers are equal, meaning the heuristic is valid.

Case 2: The number of moves estimated considers an "illegal" move, meaning that in reality more than one legal move would have to be made to get to that state. This means that our estimate will be lower than the actual number, making it an admissible heuristic.

As we have seen, this heuristic essentially makes it an easier problem, meaning that the estimate will always be less than or equal to the actual number.

Direct Cost (Euclidian Distance)

This is perhaps our simplest heuristic. All this does is examine the straight line distance between two points using the pythagorean theorem. This is valid because the shortest distance between two points is a straight line, so this heuristic always provides an optimistic estimation, as the best our die can do is go in a straight line.

Performance

Puzzle	Nodes Generated	Nodes Visited
Puzzle1	11C	22C
Puzzle2	9C	19C
Puzzle3	10C	21C
Puzzle4	10C	21C
Puzzle5	10C	21C