Assignment 1 - A\*

Group 7 -Nick Panzarino, Cameron Jones, Joey Perez, Jeff Sirocki

Worcester Polytechnic Institute

CS 4341 – Introduction to Artificial Intelligence

# Running the Program

## A\* Search

Run the jar with the following command:

“java –jar astar.jar <FILENAME> <HEURISTIC> [OPS]”

The filename is the path to the world to be analyzed

The heuristic is a value from 1 to 6 inclusive

The ops value is optional. It is a boolean, whose value determines the search method used. The default value is “1”, which will use a graph search algorithm. If a value of “0” is passed, it will instead use a tree search algorithm.

The algorithm implements both extra credit functionalities: The demolish action and jumping off the edge of the world (if it is the solution with the highest possible score).

## World Builder

We developed a “World Builder” which would generate a rectangle of assorted numbers to act as a world. It does require the user to open the file and manually set the Goal and Start points. It can be run using the included exe or if desired it can be compiled by “gcc worldGenerator.c”

It can be run with the following command:

“./worldGenerator <VERTICAL> <HORIZONTAL> <OUTPUT FILENAME>”

Vertical is the height of the grid

Horizontal is the width of the grid

Output Filename is the name of the file the grid should be written to

# Analysis

## Tree Search vs Graph Search

Initially the search was implemented using a tree search since it was easier to setup. Then looking at the poor performance, it was amended to have the ability to do it as a graph search instead. This yielded better performance and each one of the results and graphs are listed under their given world. The Tree Search Data is in Blue, the Graph Search data is in Orange. The charts below illustrate our observations:

## World 1

|  |  |  |  |
| --- | --- | --- | --- |
| World 1 | | | |
| Heuristic | score | expanded | actions |
| 1 | 96 | 59 | 4 |
| 2 | 96 | 53 | 4 |
| 3 | 96 | 7 | 4 |
| 4 | 96 | 7 | 4 |
| 5 | 96 | 5 | 4 |
| 6 | 96 | 5 | 4 |

|  |  |  |  |
| --- | --- | --- | --- |
| World 1 | | | |
| Heuristic | score | expanded | actions |
| 1 | 96 | 21 | 4 |
| 2 | 96 | 19 | 4 |
| 3 | 96 | 7 | 4 |
| 4 | 96 | 7 | 4 |
| 5 | 96 | 5 | 4 |
| 6 | 96 | 5 | 4 |

## World 2

|  |  |  |  |
| --- | --- | --- | --- |
| World 2 | | | |
| Heuristic | score | expanded | actions |
| 1 | 91 | 3099 | 6 |
| 2 | 91 | 2310 | 6 |
| 3 | 91 | 316 | 6 |
| 4 | 91 | 89 | 6 |
| 5 | 91 | 124 | 6 |
| 6 | 91 | 11 | 6 |

|  |  |  |  |
| --- | --- | --- | --- |
| World 2 | | | |
| Heuristic | score | expanded | actions |
| 1 | 91 | 120 | 6 |
| 2 | 91 | 94 | 6 |
| 3 | 91 | 50 | 6 |
| 4 | 91 | 43 | 6 |
| 5 | 91 | 26 | 6 |
| 6 | 91 | 8 | 6 |

## World 3

|  |  |  |  |
| --- | --- | --- | --- |
| World 3 | | | |
| Heuristic | score | expanded | actions |
| 1 | 89 | 6491 | 6 |
| 2 | 89 | 2233 | 6 |
| 3 | 89 | 1253 | 6 |
| 4 | 89 | 425 | 6 |
| 5 | 89 | 279 | 6 |
| 6 | 89 | 9 | 6 |

|  |  |  |  |
| --- | --- | --- | --- |
| World 3 | | | |
| Heuristic | score | expanded | actions |
| 1 | 89 | 98 | 6 |
| 2 | 89 | 60 | 6 |
| 3 | 89 | 48 | 6 |
| 4 | 89 | 37 | 6 |
| 5 | 89 | 24 | 6 |
| 6 | 89 | 9 | 6 |

## World 4

|  |  |  |  |
| --- | --- | --- | --- |
| World 4 | | | |
| Heuristic | score | expanded | actions |
| 1 | 86 | 85670 | 8 |
| 2 | 86 | 20766 | 8 |
| 3 | 86 | 6055 | 8 |
| 4 | 86 | 1420 | 8 |
| 5 | 86 | 1144 | 8 |
| 6 | 86 | 10 | 8 |

|  |  |  |  |
| --- | --- | --- | --- |
| World 4 | | | |
| Heuristic | score | expanded | actions |
| 1 | 86 | 214 | 8 |
| 2 | 86 | 156 | 8 |
| 3 | 86 | 90 | 8 |
| 4 | 86 | 48 | 8 |
| 5 | 86 | 49 | 8 |
| 6 | 86 | 10 | 8 |

## World 5

|  |  |  |  |
| --- | --- | --- | --- |
| World 5 | | | |
| Heuristic | score | expanded | actions |
| 1 | 66 | 62018 | 16 |
| 2 | 66 | 30983 | 16 |
| 3 | 66 | 13943 | 16 |
| 4 | 66 | 6269 | 16 |
| 5 | 66 | 4610 | 16 |
| 6 | 65 | 25 | 15 |

We couldn’t run a full search on map 5 when using the tree search because it was too inefficient.

## Cliff Jumper

This map was setup to showcase our ability to handle the case where jumping off the ledge is a better idea then trying to get to the goal because it has a lower cost than trying to reach the goal, or its impossible to reach the goal. To test the agent’s ability to jump off the edge of the world (if it is the best option), run the application on the cliffJumper.txt file.

# Questions

## How Do the 5 Heuristics Vary in Effectiveness

Heuristic 1: The first heuristic was effective in finding the best solution, but expanded far more nodes than any other heuristic.

Heuristic 2: The second heuristic was effective in finding the best solution, but likewise expanded far more nodes than the higher heuristics.

Heuristic 3: The third heuristic was effective in finding the best solution and made a drastic improvement off of heuristics one and two by expanding much fewer nodes.

Heuristic 4: The fourth heuristic was effective in finding the best solution and made a drastic improvement off of heuristic three by expanding much fewer nodes.

Heuristic 5: The fifth heuristic was effective in finding the best solution, but was often tied with heuristic 4 on simpler maps or ahead on more complicated maps with world 4 being the exception where heuristic 4 outperformed heuristic 5.

Heuristic 6: The sixth heuristic was exceptionally effective in finding a *good enough* solution and made a drastic improvement to number of nodes expanded. It performed optimally in worlds 1-4 and sub optimally in world 5 by 1 point.

## How much gain is there to using *any* heuristic (#1 vs. #2)

When comparing the six heuristics, there are significant gains made. In our case the higher number heuristics generally performed better than the lower number heuristics except for a few exceptions. The charts and tables above illustrate this trend.

When comparing heuristic 1 vs. heuristic 2 specifically we observed several things. In each world, both heuristics found optimal solutions, but heuristic 2 always outperformed heuristic 1. Depending on the world, heuristic 2 could have been up to 3 times more optimal. This advantage over heuristic 1 made heuristic 2 far better for the agent to use to search the map.

## Is #5 noticeably more effective than the other heuristics

When comparing heuristic 5 to the other heuristics we observed several things. In each world heuristic 5 found optimal solutions and was more efficient than heuristics 1-4 with the exception of heuristic 4 in world 4. As the maps got more complicated like in world 5, we saw that Heuristic 5 was up to 15 times better than heuristic 1, 7 times better than heuristic 2, 3 times better than heuristic 3, and 1.5 times better than heuristic 4. These gains in efficiency made heuristic 5 the best heuristic for the robot A\* search that would always be optimal. When comparing heuristic 5 to heuristic 6, and because they are identical except that heuristic 6 is an overestimate compared to 5, we see that heuristic 6 performs much more quickly but can also perform sub optimally. This observation can be seen in world 5 where heuristic 5 found the optimal solution with a score of 66 and heuristic 6 got 65, but heuristic 6 was almost 200 times more efficient. Thus, heuristic 5 was more effective than heuristics 1-4 on average but there is a clear trade-off between heuristic 5 and 6 as the world size increases.

## For heuristic #6: how does its solution quality compare with #5.

## Is it performing noticeably worse? How much more efficient is it?

Heuristic 6’s solution quality was good and tied with heuristic 5 for worlds 1-4, but in world 5 it performed sub optimally compared to heuristics 1-5. This means that as the worlds’ get more complex, heuristic 6 will become noticeably more sub optimal, but still beat heuristic 5 or any previous one in efficiency by an outstanding margin as seen in world 5.