Informatics 2D Coursework 2 Report

1. (5 marks) Task 2.1 Design

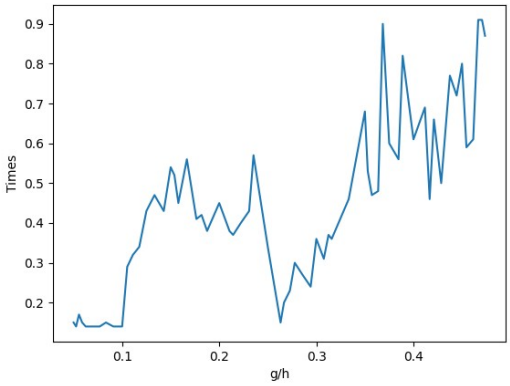
My harder problem included a second instance of mineBot. This increases the number of branches in the search tree by introducing another value to consider for each action involving the mineBot, thereby increasing run time and making the problem "harder".

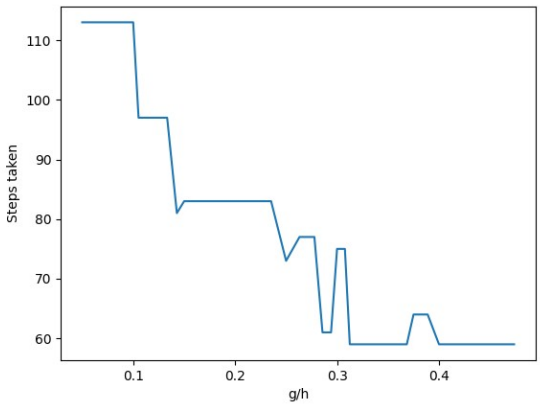
2. (10 marks) Task 2.2 Evaluation

In this experiment, the weight values of g and h in a best-first search algorithm where varied to measure the effect on the search performance.

In total 90 tests were run for different combinations of the weights of g and h. The tests were conducted such that g<(h/2) is always true. This was due to time and computation restraints as when g is close to h, the run time of the algorithm increases. A special test was run with equal weights using the 'problem-1.pddl' file as this is slightly less computationally expensive. The purpose of this test was to confirm that equal weights does in fact still allow the algorithm to function. The results of this test were a run time of 1112 seconds, and a path length of 62 steps.

The quantitative results of the tests are displayed below in graphs depicting the ratio g/h against the total number of steps in the completed plan for this ratio, and the total time (seconds) taken to find the plan.





It was observed, as mentioned above, that as the ratio of g to h increases, the run time of the algorithm would increase. It was also shown that the total number of path steps would decrease as the ratio increases.

When considering the function of g and h in the search heuristic, these results are to be expected. A greater weight on g would force the algorithm to prioritise proximity to the start node, this means more nodes are expanded during the search thereby increasing run time.

The expansion of more nodes is also directly correlated to the decrease in the number of steps in the path. More expanded nodes means more parts of the search tree are explored so the algorithm comes closer to finding the optimal path.

3. (25 marks) Task 3.4 Your Extension

Another realistic factor of the mine world would be the durability of the hammer. A hammer cannot be used an infinite amount of times without breaking, and in an industrial mine would likely have to be repaired on a regular basis.

This would increase the complexity of the plan as a working hammer is necessary to the break ores which are vital to the bot achieving its goal.

This factor was implemented using a `durability` function which exists for each hammer and decreases by 1 upon each instance of `BREAK`. An action `REPAIR` was implemented, it takes the tile, energy-station, bot and hammer as parameters. If the bot is holding the hammer and on the estation tile, the hammer's durability is increased to 2.