UNIVERSITÉ Concordia UNIVERSITY

DEPARTMENT OF ENGINEERING

Analysis

Presented to

Prof. Leila Kosseim

COMP 472

Prepared by

Annie Tran 40065592

Dann Kaoussarani 40133457

Michael Takenaka 40095917

Fall 2022

1. **The length of the solutions across algorithms and heuristics. When do you have the lowest-cost solution?**

   UCS and A/A* with heuristics 1, 2 and 4.

2. **The admissibility of each heuristic and its influence on the optimality of the solution.**

   Heuristic 1: The number of blocking vehicles.

   This heuristic is admissible because if there is a blocking vehicle, that vehicle has to move. Thus, we do not overestimate the cost of reaching the goal. In terms of optimality, it performed closely to the other heuristics(execution time). The length of the search path is higher than heuristic h3 and h4 when used with the A/A* search algorithm while also finding the shortest solution.

   Heuristic 2: The number of blocked positions.

   This heuristic is admissible because similarly to heuristic 1, if there is a blocking position, that position has to be unblocked. Thus, we do not overestimate the cost of reaching the goal. In terms of optimality, this heuristic performed the best in terms of runtime for most of the puzzles. The length of the search path is the same as h1. We can see that it finds the shortest solution in approximately the same amount of searches as H1. The length of the search path is higher than heuristic h3 and h4 when used with the A/A* search algorithm while also finding the shortest solution.

   Heuristic 3: The value of h1 multiplied by a constant λ of your choice, where $\lambda > 1$.

   This heuristic is not admissible. The act of multiplying the number of blocking cars by a constant larger than 1 makes it so the heuristic will overestimate the actual cost of moving in that position. We can also see that the solution path length in A/A* using heuristic 3 results in search paths longer than the UCS solution for the same initial board. The length of the search path is usually lower than heuristic h1, h2 and h4 when used with the A/A* search algorithm however it does not find the shortest solution.

Heuristic 4: The number of blocked cars+ the minimum amount of cars blocking the previously mentioned blocked car if it's vertical. If the car in front is horizontal return 1 (since it should be easily removed 90% of the time) and return 0 if path is clear

This heuristic is admissible. Due to the structure of the algorithm, the amount of cars blocking the path to the goal + the minimum amount of cars blocking vertical blocked cars, we make it so it is impossible for the algorithm to overestimate the cost to reach the solution. Additionally, by looking at out test (in the A* algorithm) H4 always had the same solution length path as the UCS of the same puzzle. The length of the search path is usually lower than heuristic h1 and h2 when used with the A/A* search algorithm while also finding the shortest solution.

3. **The execution time across algorithms and heuristics. Is an informed search always faster?**

Yes, theoretically, an informed search algorithms should be better (GBFS and A/A*). We can see that in most of our tests, the execution time of the informed algorithms usually finds the solution faster than the UCS algorithm. Additionally, we see that the length of the search path is usually shorter than UCS, because we are first searching paths that have more potential to lead to the solution.

However, in terms of execution times, there are a few instances where UCS is faster than the informed algorithms. For example:

24	UCS	NA	49	13581	57.67000008
24	GBFS	h1	63	9529	62.54849839
24	GBFS	h2	63	9529	53.80150056
24	GBFS	h3	63	9529	58.91300011
24	GBFS	h4	89	9397	65.0565002
24	A/A*	h1	49	12038	52.40549779
24	A/A*	h2	49	12038	51.64299822
24	A/A*	h3	53	11935	76.93350053
24	A/A*	h4	49	11893	60.13850093

We can see here that the UCS has a faster running time than some of the algorithms below. However, if we take a look at the length of the search path, we can see that the informed algorithms always find the solution in smaller amounts of steps. This is due to the extra processing time needed to calculate the value of the heuristic of each board. In short, the execution time might be higher in this case, but the amount of steps to reach there is still shorter.

4. Other interesting facts that you deem worthy of describing.

The fastest algorithm to find a solution (not the shortest one) is the greedy best first search algorithm since it does not care to find the shortest path but just a path and that is reflected in our outputs.

For A/A* the best heuristic from worst to best to find the shortest solution on average is h2, h1, h4.