Universität Zürich[UZH]

Introduction to Artificial Intelligence
Exercise Sheet 3

Laurin van den Bergh, 16-744-401
Yufeng Xiao, 19-763-663
Nora Beringer, 19-734-227

Universität Zürich
Institut für Informatik

Due: March 16, 2022

Exercise 3.1

a) The wolf and goat can never be alone on a river bank, as well as the cabbage and goat can never be alone on a river bank.
Heuristic: number of items on the wrong river bank, except the farmer.
Initial state: $\langle L, L, L, L\rangle$
Goal state: $\langle R, R, R, R\rangle$
The goal state is reached in at least three steps, as wolf, goat and cabbage need to be placed separately on the goal river bank. We ignored states which will lead to a sink state. The result is informative in the way that we know that reaching the goal state can be achieved in at least three steps.
Our heuristic counts the number of items on the left river bank.
b) It is admissible as it doesn't overestimate the cost of reaching the goal state as the lowest possible cost from the initial state is at least three and the highest heuristic value we can achieve is 3. To reach a goal state each item has to be moved to the right river bank. Therefore our heuristic provides a lower bound on the number of necessary moves.
Side note: If the heuristic is admissible, then it is safe and goal-aware.
c) It is consistent as the drop in the heuristic value is at maximum 1, $h(s) - h(s') \leq 1$ for any two states s, s', because only one of the items can be transported from the left to the right shore at a time. The action cost is constant 1, $cost(a) = 1$ for every action a. Therefore $h(s) \leq h(s')$ and $h(s) \leq h(s') + cost(a)$ for all transitions $s \xrightarrow{a} s'$.

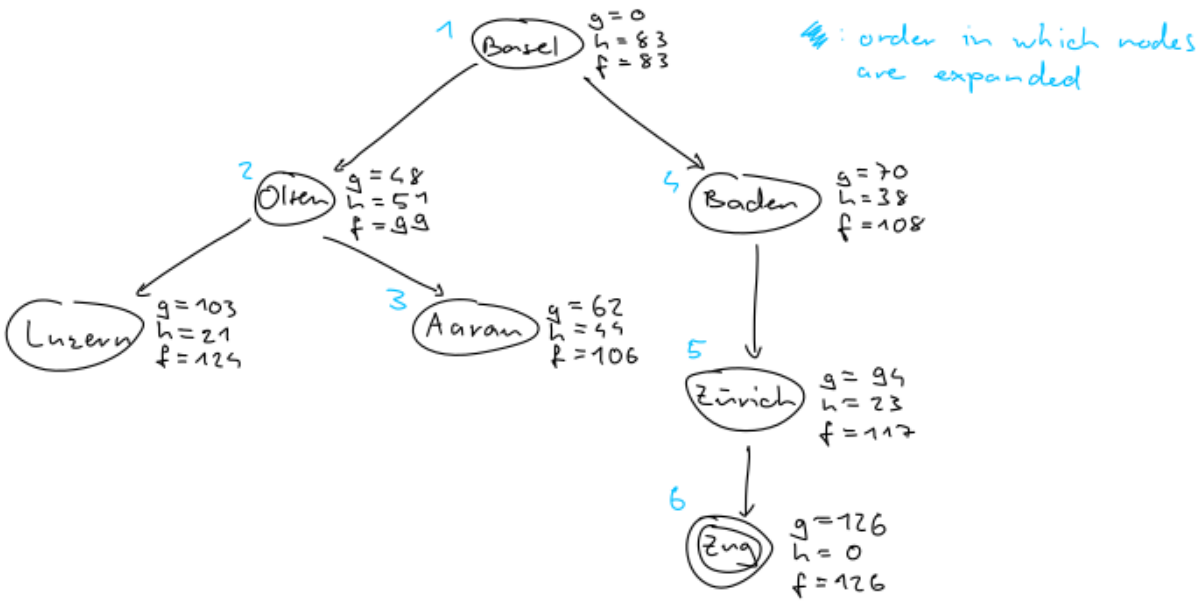
Exercise 3.2

a) and b) see Figure 1.

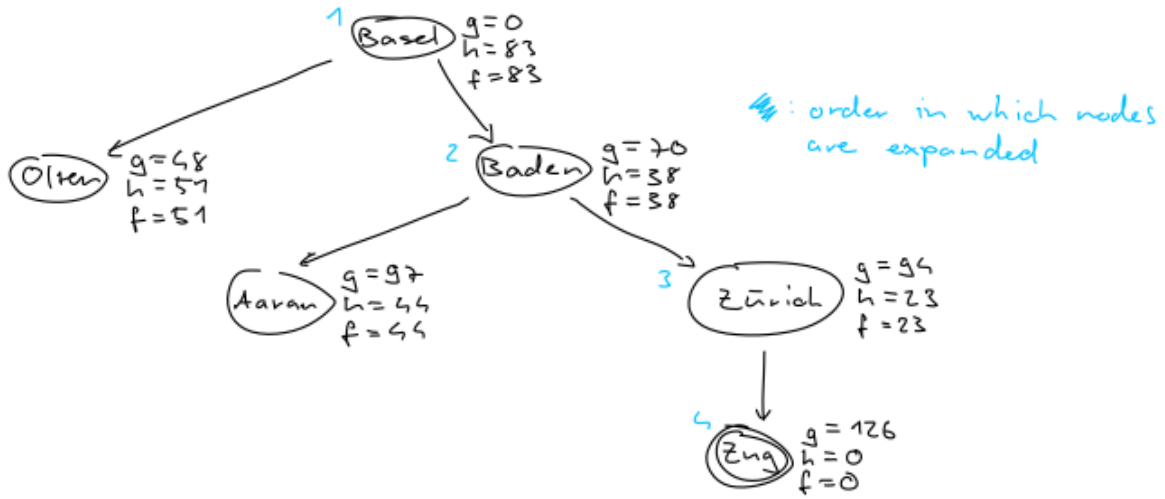
c) The heuristic is admissible, and thus also safe and goal-aware, and consistent, therefore we know that A* and greedy best first search will find the optimal solution even if no reopening is used. It is also to be expected, that uses fewer steps than A* (4 vs. 6) as it is known to be very fast.

Exercise 3.3

b) We can clearly see (Figure 3) that the plan costs increase in the problem size for all algorithms, which is to be expected. Also we see that for small problems, the differences between the algorithms is almost 0. Note that for the plots in Figure 2 the y axes are logarithmic, to reduce the effect of the A* algorithm, and depth-first search for the cost, which generally has lot higher values. Further, we see that the costs for weighted A* increase as the weight increases. This makes sense as A* guarantees the optimal cost and the further we increase the weight, the guaranteed upper bound on the cost increases,



(a)



(b)

Figure 1

however speed decreases dramatically for these algorithms. Also we can se that breath-first search and A* expand a lot more states than the other algorithms, this of course correlates with the runtime of the algorithms. We get a similar picture for the generated states, with the difference that, of course, all have a larger scale and breadth-first search comparatively generates less states than the other algorithms.

Note that we edited the files "search.py" and "experiment.py" alongside "weighted_astar_search.py" in order to allow us to make some plots. Therefore we also added the file "plots.py".

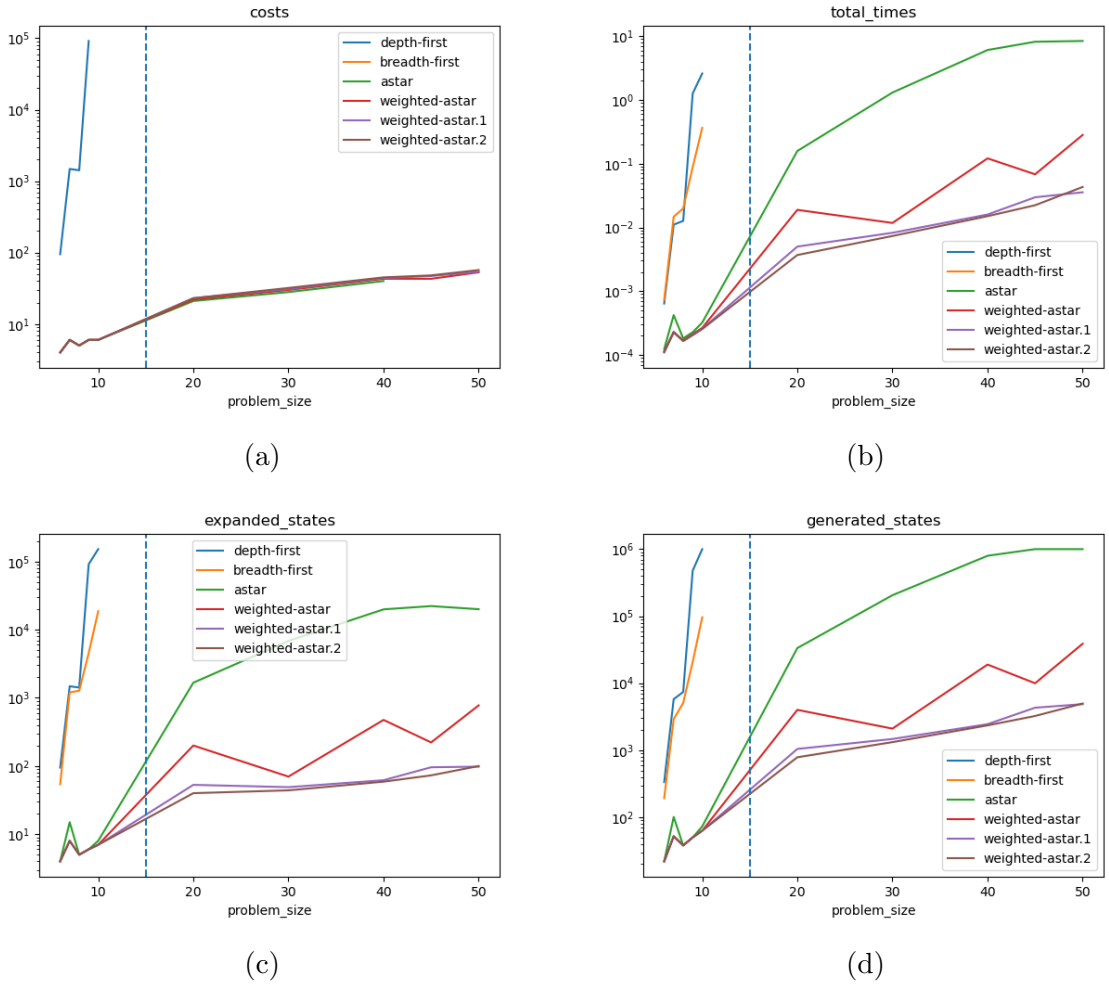


Figure 2: The vertical line separates the simple problems (left) from the hard problems (right).

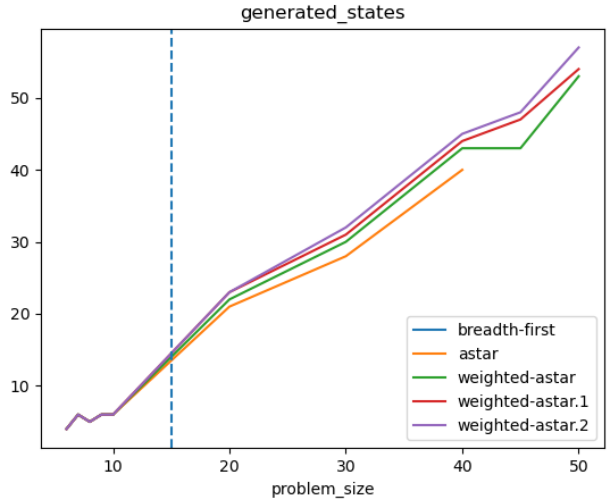


Figure 3