(Un)informed Search Lab

Martín Alejandro Molinero Morales

A01204818

Moisés Montaño Copca

A01271656

- Which heuristics did you use for the A* algorithm?

- To do the UCS we used an heuristic of value 0.

- The consistent heuristic was the number of misplaced containers times 2, because moving each container costs at least two.

- The non-consistent heuristic was the number of misplaced container times ten, because it exceeds the previous rule.

- Test your program with a couple of different problems. Increase the size of the problem to test the limits of your program. Make a table comparing **how many nodes are searched** to find the answer for each problem. For this table, you should compare a number of different problems (at least 3) to avoid a statistical bias. Which of the four algorithms searches the least nodes and which one take the most?

# of stacks	**Max.**	**UCS**	**A* with consistent**	**A* with inconsistent**	**Test file**
4	3	16,834 Nodes. Optimal found.	744 Nodes. Optimal found.	15 Nodes. Optimal not found.	a.in
3	6	9,122 Nodes. Optimal found.	1,458 Nodes. Optimal found.	265 Nodes. Optimal found.	b.in
6	2	26,240 Nodes. Optimal found	518 Nodes. Optimal found.	52 Nodes. Optimal found.	c.in

- Why does this happen?

Because the UCS is sorting the priority queue only by considering the path cost, which will make it search in a lot of nodes if the solution is far from the start. Meanwhile the A* with inconsistent heuristic is looking quicker for furthest distance, however it doesn't always give the optimal solution.

- Which algorithms are optimal? Why?

UCS and A* with a consistent heuristic are optimal, because they don't overestimate the cost, and even when they need to search in more nodes they always give the optimal solution when there is one.

- In your opinion, what are the benefits of simpler algorithms versus more complex ones?

The simpler algorithms always give the optimal solution, nevertheless they may not be the most efficient ones when searching for it.