

1. The heuristic used was only seeing the elements that are not at their goal state.

2.

Case	UCS	A* consistent	A* not consistent
3 (A, B); (C, D, E); () (E); (C, B, A); (D)	413	187	790
3 (A); (B); (C); () (); (A); (B); (C)	77	42	57
4 (A, B); (C, D); (E, F) (B, A); (C, D); (F, E)	4405	2369	4258

Table 1. Comparison of how many nodes every algorithm visited

3. We can see that UCS visited the most nodes in almost every case, with A* no consistent is kind of tricky, the heuristic is given randomly so there will be cases where A* no consistent would be more optimal than A* consistent or worse than UCS.
4. A* consistent is optimal because of the way the heuristic was implemented, with this algorithm, it'll try always to reach the optimal path visiting the less nodes.
5. Complex algorithms are always better if we implemented in the right way (and the problem deserves its use) they always look the optimal solution, some cons are that they're harder to implement than simple ones, and also they require most of the time more computational power.