Udacity Artificial Intelligence Nanodegree
Project 3: Implement a Planning Search
Heuristic Analysis
Mateusz Iwanicki

Optimal solutions to planning problems
Problem 1 can be optimally solved in 6 actions, e.g.
Load(C1, P1, SFO)
Fly(P1, SFO, JFK)
Unload(C1, P1, JFK)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
Unload(C2, P2, SFO)

Problem 2 can be optimally solved in 9 actions, e.g.
Load(C3, P3, ATL)
Fly(P3, ATL, SFO)
Unload(C3, P3, SFO)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
Unload(C2, P2, SFO)
Load(C1, P1, SFO)
Fly(P1, SFO, JFK)
Unload(C1, P1, JFK)

Problem 3 can be optimally solved in 12 actions, e.g.
Load(C2, P2, JFK)
Fly(P2, JFK, ORD)
Load(C4, P2, ORD)
Fly(P2, ORD, SFO)
Unload(C4, P2, SFO)
Load(C1, P1, SFO)
Fly(P1, SFO, ATL)
Load(C3, P1, ATL)
Fly(P1, ATL, JFK)
Unload(C3, P1, JFK)
Unload(C2, P2, SFO)
Unload(C1, P1, JFK)

Uninformed planning searches (results)
All algorithms returned results for problem 1 in a sensible time. Algorithms missing from problem 2 and 3 write-up took too long to run, and had to be aborted before finding the optimal set of actions.

Air Cargo Problem	Search algorithm	Node expansions	Goal tests	Time elapsed (seconds)	Path Length	Optimal solution?
Problem 1	**Breadth First Search**	**43**	**56**	**0.028**	**6**	**YES**
	Breadth First Tree Search	1458	1459	0.783	6	YES
	Depth First Graph Search	**21**	**22**	**0.013**	**20**	**NO**
	Depth Limited Search	101	271	0.078	50	NO
	Uniform Cost Search	**55**	**57**	**0.029**	**6**	**YES**
	Recursive Best First Search	4229	4230	2.214	6	YES

	Greedy Best First Search	**7**	**9**	**0.006**	**6**	**YES**
Problem 2	**Breadth First Search**	**3343**	**4609**	**11.257**	**9**	**YES**
	Depth First Graph Search	**624**	**625**	**2.975**	**619**	**NO**
	Depth Limited Search	222719	2053741	780.864	50	NO
	Uniform Cost Search	**4852**	**4854**	**10.138**	**9**	**YES**
	Greedy Best First Search	990	992	1.922	15	NO
Problem 3	**Breadth First Search**	**14663**	**18098**	**84.600**	**12**	**YES**
	Depth First Graph Search	**408**	**409**	**1.499**	**392**	**NO**
	Uniform Cost Search	**18235**	**18237**	**41.827**	**12**	**YES**
	Greedy Best First Search	5614	5616	13.094	22	NO

Informed (heuristic based) planning searches (results)

Air Cargo Problem	Search algorithm	Node expansions	Goal tests	Time elapsed (seconds)	Path Length	Optimal solution?
Problem 1	A* Search H 1	55	57	0.032	6	YES
	A* Search H Ignore Precond.	**41**	**53**	**0.031**	**6**	**YES**
	A* Search H PG Levelsum	11	13	0.707	6	YES
Problem 2	A* Search H 1	4852	4854	9.313	9	YES
	A* Search H Ignore Precond.	**1450**	**1452**	**3.373**	**9**	**YES**
	A* Search H PG Levelsum	86	88	122.914	9	YES
Problem 3	A* Search H 1	18235	18237	40.848	12	YES
	A* Search H Ignore Precond.	**5040**	**5042**	**13.080**	**12**	**YES**
	A* Search H PG Levelsum	315	317	835.145	12	YES

Analysis of the results

Uninformed search

As the greedy best first search algorithm is not stable, and doesn’t guarantee optimal solution [3] and [1] (section 3.5.1) (it was the best algorithm for problem 1, finding an optimal solution, but failed to repeat it for problem 2 and 3), we focus on the Breadth First Search, Depth First Search and Uniform Cost Search.

Following the explanation in [1] section 3.4, we know that Breadth First Search (section 3.4.1) and Uniform Cost Search (section 3.4.2) are both guaranteed to be complete and optimal, because of how the graph is expanded (by either expanding the shortest paths in terms of number of nodes or length of the path). However, they also take longer to run, and require more memory. In our case, the Uniform Cost Search had much better execution times (especially with complex Problem 3), but expanded marginally more nodes.

Compared to that, Depth First Search (section 3.4.3) performs much faster and uses much less memory. Especially in more complex Problem 2 and Problem 3, the difference in number of nodes expanded was an order of magnitude. However, as the solution is not optimal, in many cases the path lengths were unacceptably long (392 actions for Problem 3, compared to the optimal length of 12).

The Greedy Best First Search algorithm provides a balance between completeness and execution speed. It didn’t manage to find optimal solution in 2 our of 3 problems, but the produced path length of 15 for Problem 2 and 22 for Problem 3 were much shorter than the ones produced by Depth First Search. It also run considerably shorter than Breadth First Search and Uniform Cost Search. (side note: technically, greedy best first search algorithm is not uninformed, as it is using simple heuristic function = distance to the goal [1] section 3.5.1)

Informed (Heuristic based) searches

The base A* Search used the H 1 heuristic, which effectively ignored the heuristic part all together, using the same cost of 1 for all the nodes [h(n)] (as can be deducted from [1] section 3.4.1 with the definition of A* in [2] section C). This is effectively the uniform search algorithm, taking to consideration the length of the path leading to the node (it can be seen from the results, that the number of nodes expanded is the same for those two algorithms)

Comparing the Ignore Preconditions and Level Sum Heuristic, we can see the stark differences. The Ignore Preconditions heuristic is extremely fast, as it is effectively estimation of “unturned” bits in a current state. It requires more node expansion, and goals tests, but is guaranteed to find optimal solution, and it does it fast. The execution time and number of nodes expanded could rival the uninformed Greedy Best First Search. The Level Sum heuristic on the other hand uses much less memory, but requires the longest time to run. Finding a solution to Problem 3 took almost 14 minutes (more than 10 minutes allocated in the task).

For the uninformed search, we propose the Uniform Cost Search as a fast algorithm guaranteed to find optimal solution. For the informed search, the A* search with Ignore Preconditions heuristic is the best choice. Given that air cargo transportation is expensive, we chose to disregard the algorithms yielding the not-optimal solutions.

References

[1] Russell, Stuart J.; Norvig, Peter (2003), Artificial Intelligence: A Modern Approach (2nd ed.), Upper Saddle River, New Jersey: Prentice Hall, ISBN 0-13-790395-2

[2] Hart, P. E.; Nilsson, N. J.; Raphael, B. (1968). "A Formal Basis for the Heuristic Determination of Minimum Cost Paths". IEEE Transactions on Systems Science and Cybernetics SSC4. 4 (2): 100–107. doi:10.1109/TSSC.1968.300136

[3] Black, Paul E. (2 February 2005). "greedy algorithm". Dictionary of Algorithms and Data Structures. U.S. National Institute of Standards and Technology (NIST). https://xlinux.nist.gov/dads//HTML/greedyalgo.html