AIND-Planning report

Optimal plans

For all three air-cargo problems, an optimal solution could be found by both uninformed and informed searches. By definition the uninformed breadth-first-search always find an optimal solution. For the informed case – the A* search with both heurisics also find an optimal solution. There might be some differences in the order of the steps taken, but the solution has the same length, so I'm showing just one of the possible plans.

Problem 1 optimal plan - length 6:

```
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
Unload(C2, P2, SFO)
Fly(P1, SFO, JFK)
Unload(C1, P1, JFK)
```

Problem 2 optimal plan – length 9:

```
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Load(C3, P3, ATL)
Fly(P2, JFK, SFO)
Unload(C2, P2, SFO)
Fly(P1, SFO, JFK)
Unload(C1, P1, JFK)
Fly(P3, ATL, SFO)
Unload(C3, P3, SFO)
```

Problem 3 optimal plan – length 12:

```
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P2, JFK, ORD)
Load(C4, P2, ORD)
Fly(P1, SFO, ATL)
Load(C3, P1, ATL)
Fly(P1, ATL, JFK)
Unload(C1, P1, JFK)
Unload(C3, P1, JFK)
Fly(P2, ORD, SFO)
Unload(C2, P2, SFO)
Unload(C4, P2, SFO)
```

Non-heuristic search result metrics

Problem 1	Optimality	Time elapsed, s	Nodes	Plan length
Breadth-first search	yes	0,0391	43	6
Depth-first search	no	0,0183	21	20
Depth-limited search	no	0,119	101	50
Uniform-cost search	yes	0,0466	55	6

Problem 2	Optimality	Time elapsed, s	Nodes	Plan length
Breadth-first search	yes	16,653	3343	9
Depth-first search	no	3,927	624	619
Depth-limited search	no	N/A	N/A	N/A
Uniform-cost search	yes	15,090	4853	9

Problem 3	Optimality	Time elapsed, s	Nodes	Plan length
Breadth-first search	yes	120,208	14663	12
Depth-first search	no	2,116	408	392
Depth-limited search	no	N/A	N/A	N/A
Uniform-cost search	yes	65,380	18235	12

The result tables for the three problems show values that are expected for the used search algorithms and the metrics for each one of them is according its definition. Each table corresponds entirely to the comparison table of uninformed search strategies[1]. The optimal solution for the three air-cargo problems is found only be the BFS and UCS algorithms. However looking at the time elapsed and the nodes expansion clearly shows the tradeoffs of using this searches. Compared to the depth-first search results, they are 2x slower and 2x more nodes are expanded just for the simple first problem and getting to 30-60x slower time for the hardest problem 3 with 10-15x more nodes expanded. The optimality comes at a pretty high price for such a simple problem with this small number of fluents. On the other side the

[1] Stuart Russel and Peter Norvig, AI: A Modern Approach, 3rd edition, chapter 3, figure 3.21

comparison of the plan lengths between BFS/UCS and DFS is somewhat acceptable for the first problem, but for the 2nd and 3rd is just a huge difference that would be just unusable for solving a real-world problem even though it is faster. I've also added the depth-limited search in order to show its limitation for this kind of problems. Based on the DFS algorithm, but with only 50 as maximum depth, it is clear that for problem 2 and 3, it cannot find a solution in a reasonable time. As seen by the DFS plan lengths for those problems – 619 and 392, it won't be possible for the depth-limited search to find them in such a short time. Given the fact that the optimal solutions are for 9 and 12 moves, the maximum depth of 50 is not limiting this algorithm of finding a solution, but the search time is just too big.

Heuristic search result metrics

Problem 1	**Optimality**	**Time elapsed, s**	**Nodes**	**Plan length**
A* ignore preconditions	yes	0,048	41	6
A* level sum	yes	0,493	11	6

Problem 2	**Optimality**	**Time elapsed, s**	**Nodes**	**Plan length**
A* ignore preconditions	yes	5,454	1450	9
A* level sum	yes	41,13	86	9

Problem 3	Optimality	Time elapsed, s	Nodes	Plan length
A* ignore preconditions	yes	20,142	4951	12
A* level sum	yes	201,745	312	12

By the definition[2] of an Informed search, which A* is a type of, it can find solution more efficiently and in most cases optimal. In the case of this three air-cargo problems, the A* search algorithm with both heuristic functions produce optimal plan. However this may not always be the case. For A* to produce optimal solution, its heuristic must be admissible[3], but in this case both heuristic functions are not guaranteed to be such[4]. The tradeoff between the two functions is time versus expanded nodes. The faster solution is produced by the **ignore-preconditions heuristic** with around 10x better time. However this comes at the price of expanding 10x more nodes. Ignore-preconditions function gives a decent estimation of the distance to the goal state, but I think that the **level sum heurisic** is the better function in this case, as it gives a lot better prediction for the price of reaching our goals, which is clearly shown by the much less number of node expansions.

The result of the A* search with level sum heuristic are a lot better than the non-heuristic searches – Breadth first search and Uniform cost search, and is concurrent to the Depth first search results. Even the much slower execution time of A* can be justified by the low number of expanded nodes and the optimal result. So I think that using the A* search is much more reasonable and reliable – each time the search can be tracked and explained by following the logic of the heuristic function that it is used and the score that it produces for a given state. On the other side a depth first search is just too inaccurate for searching a good plan compared to finding an optimal plan with A* and also DFS is dependent on the order of the nodes in the search tree. Just a simple shuffle of the goals for any of the tree problems gives different results in plan length and node expansion.

[2] Stuart Russel and Peter Norvig, AI: A Modern Approach, 3rd edition, Chapter 3, Informed Search Strategies – 3.5

[3] Stuart Russel and Peter Norvig, AI: A Modern Approach, 3rd edition, Chapter 3, Optimality of A* - 3.5.2

[4] Stuart Russel and Peter Norvig, AI: A Modern Approach, 3rd edition, Chapter 10 – Heuristic for planning 10.2.3 and Planning graphs for heuristic estimation 10.3.1