Implement a planning search project: Heuristic analysis

Jaco Fourie

August 2017

1 Optimal plans

The optimal plan lengths for the planning problems 1,2, and 3 are 6, 9, and 12 respectively. Detailed descriptions of the the 3 optimal plans are found in Table 1.

Table 1: A description of the optimal plans for the 3 planning problems

Cargo problem P1	**Cargo problem P2**	**Cargo problem P3**
Load(C1, P1, SFO)	Load(C3, P3, ATL)	Load(C2, P2, JFK)
Fly(P1, SFO, JFK)	Fly(P3, ATL, SFO)	Fly(P2, JFK, ORD)
Unload(C1, P1, JFK)	Unload(C3, P3, SFO)	Load(C4, P2, ORD)
Load(C2, P2, JFK)	Load(C1, P1, SFO)	Fly(P2, ORD, SFO)
Fly(P2, JFK, SFO)	Fly(P1, SFO, JFK)	Unload(C4, P2, SFO)
Unload(C2, P2, SFO)	Unload(C1, P1, JFK)	Load(C1, P1, SFO)
	Load(C2, P2, JFK)	Fly(P1, SFO, ATL)
	Fly(P2, JFK, SFO)	Load(C3, P1, ATL)
	Unload(C2, P2, SFO)	Fly(P1, ATL, JFK)
		Unload(C3, P1, JFK)
		Unload(C1, P1, JFK)
		Unload(C2, P2, SFO)

2 Metrics for non-heuristic planning

The metrics for 4 different search strategies were measured on all three of the cargo planning problems. The results from these tests can be found in Table 2.

We notice from the results in the table that *bread_first_search*, *uniform_cost_search*, and *astar_search* always succeeded in finding the optimal solution to the planning problem and all returned solutions with equal plan lengths. However, *depth_limited_search* was not able to find a solution in a reasonable amount of time and failed to return plans that were the shortest possible length. In

Problem	Search strategy	Expansions	Goal tests	Time elapsed	Plan length
Air cargo P1	breadth_first_search	43	56	0.023 sec	6
Air cargo P1	depth_limited_search	101	271	0.065 sec	50
Air cargo P1	uniform_cost_search	55	57	0.027 sec	6
Air cargo P1	astar_search (h1)	55	57	0.028 sec	6
Air cargo P2	breadth_first_search	3343	4609	10.22 sec	9
Air cargo P2	depth_limited_search	222719	2053741	665.64 sec	50
Air cargo P2	uniform_cost_search	4852	4854	8.68 sec	9
Air cargo P2	astar_search (h1)	4852	4854	8.67 sec	9
Air cargo P3	breadth_first_search	14663	18098	72.46 sec	12
Air cargo P3	depth_limited_search	??	??	>10 min	??
Air cargo P3	uniform_cost_search	18235	18237	38.52 sec	12
Air cargo P3	astar_search (h1)	18235	18237	38.41 sec	12

Table 2: Non-heuristic test metrics

problem 3 *depth_limited_search* took very long to compute and the search was abandoned after more than 10 minutes of calculation. It is not surprising that *uniform_cost_search*, and *astar_search* did the best here as the heuristic used in *astar_search* is just a constant value meaning it just becomes greedy best first search which makes it identical to the *uniform_cost_search* algorithm. As mentioned in the textbook for this course (Chap.3)[1] A* search is just breadth-first search with an added heuristic cost added to the node step-cost which in this case is just a constant (heuristic h1). Since *uniform_cost_search* is an extension of *breadth-first-search* to make it optimal for any step-cost function [1] we would expect *uniform_cost_search* to be better or at least as good as *breadth-first-search*.

3 Metrics from domain-independent heuristics

As in the previous section, metrics for specific search strategies were measured using all three of the cargo planning problems. In this section we used a planning graph and tested the *ignore_preconditions* and *level_sum* heuristics. The results from our tests are found in Table 3.

Problem	Search strategy	Expansions	Goal tests	Time elapsed	Plan length
Air cargo P1	ignore preconditions	41	43	0.026 sec	6
Air cargo P1	level-sum	11	13	0.423 sec	6
Air cargo P2	ignore preconditions	1450	1452	3.056 sec	9
Air cargo P2	level-sum	86	88	107.11 sec	9
Air cargo P3	ignore preconditions	5040	5042	12.054 sec	12
Air cargo P3	level-sum	318	320	540.57 sec	12

Table 3: Test metrics from domain-independent heuristics

Using the planning graph methods all the search strategies were able to find the optimal solution in less than 10 minutes. The number of expansions and goal tests required to find these solutions were significantly less than those required by the non-heuristic methods. However, due to the increased computational cost of the heuristic calculation these methods were often slower than the non-heuristic methods of Table 2. The *ignore preconditions* found the optimal path much faster than the *level-sum* heuristic due to the simplicity of its calculation even though it required more goal tests and node expansions. This fits with the claim made in the course textbook (chap 10) that while ignoring preconditions may not be a particularly accurate estimate of the path cost it often works well in practice [1].

4 Summary of results

The optimal plan lengths for the planning problems 1,2, and 3 are 6, 9, and 12 respectively. The first problem is the simplest and all the search strategies that were tested was able to find the optimal solution in less than a second. In terms of node expansions and goal tests the *level-sum* heuristic performed the best but was still 10 times slower than all of the non-heuristic methods. For problems this small the extra computational cost involved in calculating complex heuristics is not worth it and simple approaches are best.

The second problem is more complex and we see that now the planning graph methods become viable alternatives. In terms of processing speed the *ignore preconditions* heuristic was the fastest. However, the *level-sum* heuristic was more than 10 times slower than the much simpler *uniform cost search* and *A* (greedy best first) search* strategies. The *level-sum* heuristic found the goal using far fewer node expansions and goal tests but calculating the heuristic was still too expensive to be worth it on this problem.

The third problem is the most complex and also best shows the advantage of using a planning graph and a simple heuristic as apposed to a full exponentially growing graph as used in the non-heuristic search strategies of Table 2. The best performance both in the number of node expansions and actual processing time was achieved by the *ignore preconditions* heuristic and was more than twice as fast as the best results from the non-heuristic strategies. As we saw in the second problem the *level-sum* heuristic found the solution using the least number of node expansions and goal tests but took much longer to calculate in terms of processing time. As stated in the course textbook (Chap 10), the *level-sum* heuristic is a much more accurate estimate of the true cost than the *ignore preconditions* heuristic and works well in problems like ours where the conjunction of goals are largely decomposable [1] which explains the lower number of goal tests and node expansions.

In conclusion, for these relatively simple problems the best overall search strategy is the *ignore preconditions* heuristic that uses a planning graph to approximate the problem and estimates the distance to the goal by ignoring preconditions. This allows one to estimate the best direction to search for the solution in the graph with the minimal computational cost.

References

[1] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach.* Pearson Education, 3 edition, 2010.