AI planning:
Experiment Results

Octavio Navarro-Hinojosa

1 Uninformed planning searches

The results for running the problems *air_cargo_1*, *air_cargo_2*, and *air_cargo_3*, with different search algorithms can be seen in Tables 1, 2, and 3 respectively. It is worth mentioning that the search algorithms breadth first tree search, and depth limited search took more than 10 minutes to find a plan for problems 2 and 3, and thus the corresponding metrics were not included in the results.

Table 1: Results for the *air_cargo_1* problem.

Algorithm	Expansions	Goal Tests	Time elapsed	Plan length	Optimality
breadth_first_search	43	56	0.054	6	Yes
breadth_first_tree_search	1458	1459	1.363	6	Yes
depth_first_graph_search	12	13	0.011	12	No
depth_limited_search	101	271	0.129	50	No
uniform_cost_search	55	57	0.051	6	Yes

Table 2: Results for the *air_cargo_2* problem.

Algorithm	Expansions	Goal Tests	Time elapsed	Plan length	Optimality
breadth_first_search	3343	4609	18.537	9	Yes
depth_first_graph_search	582	583	3.945	575	No
uniform_cost_search	55	57	16.341	9	Yes

Table 3: Results for the *air_cargo_3* problem.

Algorithm	Expansions	Goal Tests	Time elapsed	Plan length	Optimality
breadth_first_search	14663	18098	133.743	12	Yes
depth_first_graph_search	627	628	4.175	596	No
uniform_cost_search	18235	18237	72.508	12	Yes

2 Planning searches with heuristics

The results for running the problems *air_cargo_1*, *air_cargo_2*, and *air_cargo_3*, with A* search and with different heuristics can be seen in Tables 4, 5, and 6 respectively.

Table 4: Results for the *air_cargo_1* problem with heuristics.

Heuristic search	Expansions	Goal Tests	Time elapsed	Plan length	Optimality
A* with h1	55	57	0.051	6	Yes
A* with ignore preconditions	41	43	0.059	6	Yes
A* with pg levelsum	11	13	1.242	6	Yes

Table 5: Results for the *air_cargo_2* problem with heuristics.

Heuristic search	Expansions	Goal Tests	Time elapsed	Plan length	Optimality
A* with h1	4852	4854	16.877	9	Yes
A* with ignore preconditions	1450	1452	5.982	9	Yes
A* with pg levelsum	86	88	98.381	9	Yes

Table 6: Results for the *air_cargo_3* problem with heuristics.

Heuristic search	Expansions	Goal Tests	Time elapsed	Plan length	Optimality
A* with h1	18235	18237	73.743	12	Yes
A* with ignore preconditions	5040	5042	23.807	12	Yes
A* with pg levelsum	318	320	460.447	12	Yes

3 Analysis

3.1 Optimal plans

For the *air_cargo_1* problem, the optimal plan produced by the search algorithms is the following:

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

Even though each algorithm produced the same optimal plan, the steps to accomplish them were not necessarily reported in the same order; this happened for each of the problems tested. For example, the A* search with the ignore preconditions heuristic produced the following plan for the first problem:

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

The optimal plan for the *air_cargo_2* problem is the following:

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

Finally, the optimal plan for the *air_cargo_3* problem is the following:

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

3.2 Non-heuristic search performance

For the following analysis, only the breadth first search, depth first graph search, and uniform search algorithms are considered. The other algorithms are excluded because their execution time for problems 2 and 3 were more than 10 minutes, and the results from problem 1 can not be compared to them.

For the uninformed planning searches, the search algorithm used can drastically affect the performance of the plan generation. The algorithms that produced optimal plans were the breadth first search and uniform cost search, while the depth first algorithm failed to produce one.

For the depth first algorithm, the number of expansions is relatively low when compared to the other algorithms' expansions (at least for problems 1 and 3). Since the algorithm explores as far as possible along each branch before backtracking, it can be seen that even though a plan was produced, it may not have been the optimal one. This is also reflected in the fact that depth first searches were performed in a fraction of the time of the execution time of the other algorithms: a solution was found quickly because it was found first, not necessarily because it was an optimal one.

The breadth first algorithm produced an optimal plan, but the algorithm was not necessarily the best performing one. Even though the number of expansions is not as large as with depth first, the execution time increased more sharply as the problem became more complex. For problem 1, breadth first was only 1 millisecond slower than uniform cost. However, it was 2 seconds slower in problem 2, and 61.235 seconds slower in problem 3. For simple problems, breadth first search may be a better option because of the lower number of expansions, but for larger, more complex problems, uniform cost search is the better performing option.

4 Heuristic search performance

Even though all the heuristics found optimal plans for each problem, the performance of the search is heavily impacted by the heuristic used. The h1 heuristic was the one that needed the more expansions to produce an optimal plan, followed by the ignore preconditions heuristic, and finally the pg levelsum heuristic needed the least expansions to create an optimal plan. However, the execution time for each heuristic is the metric that differentiates each one.

The pg levelsum heuristic was the slower one for each of the problems; as the problem became more complex, it took exponentially longer to complete. The h1 heuristic took relatively the same amount of time to the ignore preconditions heuristic for the first problem. For the second and third problems, there is a clear difference in execution time. For the second problem, the ignore preconditions heuristic took 2.821 times less time to produce a plan than the h1 heuristic. For problem 3, the ignore preconditions heuristic took 3.097 times less time to execute.

The level sum heuristic was clearly the worse performing one both in terms of expansions and execution time, while the ignore preconditions heuristic was the better performing one.

5 Conclusions

The best approach to generate plans for the given problems was the A* search with the ignore preconditions heuristic. It performed better than both the non-heuristic search method, and the other heuristic search methods, in terms of both number expansions and execution time.

The ignore preconditions heuristic performed best because it estimates the minimum number of actions that must be carried out from the current state in order to satisfy all of the goal conditions by ignoring the preconditions required for an action to be executed; essentially relaxing the problem (by removing preconditions) and creating an admissible heuristic for the original problem [1].

By using a consistent heuristic, the A* search algorithm is complete and optimal, and together are able to produce an optimal plan in the least amount of time.

References

[1] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach.* Prentice Hall, Third Edition, 2010.