Planning Search Heuristic Analysis

---SAI PAVAN NUNNA

For this project, we implemented a planning search agent to solve deterministic logistics planning problems for an Air Cargo transport system. We use a **planning graph** and **automatic domain-independent heuristics with A* search** and compare their results/performance against several **uninformed non-heuristic search methods** (breadth-first, depth-first, etc.).

Problem 1 Results:

```
python run_search.py -p 1 -s 1 2 3 4 5 6 7
```

Search Strategy	**Optimal**	**Path Length**	**Execution Time (s)**	**Node Expansions**
Breadth First Search	**Yes**	6	0.045	43
Breadth First Tree Search	**Yes**	6	1.569	1458
Depth First Graph Search	No	12	0.009	12
Depth Limited Search	No	50	0.121	101
Uniform Cost Search	**Yes**	6	0.054	55
Recursive Best First Search	**Yes**	6	3.95	4229
Greedy Best First Graph Search	**Yes**	6	0.007	7

Problem 2 Results:

```
python run_search.py -p 2 -s 1 3 5 7
```

Search Strategy	**Optimal**	**Path Length**	**Execution Time (s)**	**Node Expansions**
Breadth First Search	**Yes**	9	21.22	3401
Breadth First Tree Search	**-**	-	-	-
Depth First Graph Search	No	346	2.478	350
Depth Limited Search	-	-	-	-
Uniform Cost Search	**Yes**	9	17.8	4761
Recursive Best First Search	**-**	-	-	-
Greedy Best First Graph Search	**Yes**	9	1.9	550

Problem 3 Results:

```
python run_search.py -p 3 -s 1 3 5 7
```

Search Strategy	**Optimal**	**Path Length**	**Execution Time (s)**	**Node Expansions**
Breadth First Search	**Yes**	12	167.811	14491
Breadth First Tree Search	**-**	-	-	-
Depth First Graph Search	No	1878	33.0	1948
Depth Limited Search	-	-	-	-
Uniform Cost Search	**Yes**	12	79.911	17783
Recursive Best First Search	**-**	-	-	-
Greedy Best First Graph Search	**Yes**	22	18.29	4031

Analysis:

With this 3-problem set, **Breadth First Search** and **Uniform Cost Search** are the only two uninformed search strategies that **yield an optimal action plan**. When it comes to execution speed and memory usage, **Depth First Graph Search** is the **fastest and uses the least memory**. However, it does not generate an optimal action plan (problem 1: plan length of 12 instead of 6, problem 2: plan length of 346 instead of 9, problem 3: plan length of 1878 instead of 12).

For problems 2 and 3, the Depth First Graph Search plan lengths are so much longer than the optimal path length that it wouldn't make sense to use this search strategy. **Greedy Best First Graph Search is the best alternative**.

In problems 1 and 2, it manages to find the optimal path. In problem 3, it does not find the optimal path but the path length it generates is 22 instead of 10, which is much better than Depth First Graph Search (1878 path length!). Moreover, it still provides execution time savings and uses less memory than the best search strategy for an optimal solution (**Breadth First Search).**

Informed (Heuristic) Search Strategies Analysis:

Informed search strategy — one that uses problem-specific knowledge beyond the definition of the problem itself — can find solutions more efficiently than can an uninformed strategy. In this section, we compare the performance of **A* Search using three different heuristics**. Here again, we evaluate these strategies in terms of **speed**, **memory usage** and **optimality**.

Problem 1 Results:

Search Strategy	**Optimal**	**Path Length**	**Execution Time (s)**	**Node Expansions**
A* Search with h1 heuristic	**Yes**	6	0.052	55
A* Search with Ignore Preconditions	**Yes**	6	0.070	41
A* Search with Level Sum heuristic	**Yes**	6	1.96	11

Problem 2 Results:

Search Strategy	**Optimal**	**Path Length**	**Execution Time (s)**	**Node Expansions**
A* Search with h1 heuristic	**Yes**	9	18.18	4761
A* Search with Ignore Preconditions	**Yes**	9	8.6	1450
A* Search with Level Sum heuristic	**Yes**	9	460.59	86

Problem 3 Results:

Search Strategy	**Optimal**	**Path Length**	**Execution Time (s)**	**Node Expansions**
A* Search with h1 heuristic	**Yes**	12	82.01	17783
A* Search with Ignore Preconditions	**Yes**	12	32.9	5003
A* Search with Level Sum heuristic	**-**	-	-	-

Analysis:

All heuristics yield an optimal action plan. Of the two strategies mentioned above, **A* Search with Ignore Preconditions heuristic is the fastest**. If we let search run to completion on our machine, **A* Search with Level Sum heuristic uses the least memory**, but its execution time is much slower.

Informed vs Uninformed Search Strategies:

The search strategies that generate optimal plans are Breadth First Search, Uniform Cost Search, and A* Search with all three heuristics.

As we saw earlier, when it comes to execution speed and memory usage of uninformed search strategies, **Breadth First Graph Search** is better and uses less memory than Uniform Cost Search. As for informed search strategies, **A* Search with Ignore Preconditions heuristic** is the fastest and uses the least memory. So, really, the choice is between Breadth First Graph Search and A* Search with Ignore Preconditions heuristic. Here we compare their results against our 3-problem set.

Problem 1 Results:

Search Strategy	**Optimal**	**Path Length**	**Execution Time (s)**	**Node Expansions**
Breadth First Search	**Yes**	6	0.045	43
A* Search with Ignore Preconditions	**Yes**	6	0.070	41

Problem 2 Results:

Search Strategy	**Optimal**	**Path Length**	**Execution Time (s)**	**Node Expansions**
Breadth First Search	**Yes**	9	21.22	3401
A* Search with Ignore Preconditions	**Yes**	9	8.6	1450

Problem 3 Results:

Search Strategy	**Optimal**	**Path Length**	**Execution Time (s)**	**Node Expansions**
Breadth First Search	**Yes**	12	167.811	14491
A* Search with Ignore Preconditions	**Yes**	12	32.9	5003

From the results above, because it is faster and uses less memory, **A* Search with Ignore Preconditions heuristic** would be the best choice overall for our Air Cargo problem

Discussion on Observed Results:

All three non-heuristic search strategies, that is; breadth first search, uniform cost search, and depth first graph search, find a solution to all air cargo problems. Breadth first search always considers the shortest path first [1] and a result of it it finds a solution to the problem in a reasonable amount of time and in an optimal way. Depth first graph search does find a quick solution and requires a small amount of memory, but it lacks optimality. It is not optimal because it does not consider if a node is better than another, it simply explores the nodes that take it as deep as possible in the graph even if the goal is to its right [1]. Non-heuristic based search did perform better in problem 1 and 2, which suggest that when working with simple problems using a more elaborated approach, such a A* search with heuristics, is not worth the increase in the solution complexity. Heuristic based search did perform better as the problem complexity increased. This is more evident in the air cargo problem 3, where the "A* Search with 'h_ignore_preconditions'" performance was optimal and the fastest amongst those that were optimal. It's also worth noting that the 'h_pg_levelsum' heuristic did in overall perform poorly, most likely due to the heuristic being too complex. According to the results obtained in this analysis, the breadth first search strategy can solve planning problems both fast and optimality, which makes it a good candidate to start off an analysis when dealing with search planning problems. As the complexity of the problems increase, it might be worth to consider if a heuristic based approach such as **"A* Search with 'h_ignore_preconditions'**" can outperform **breadth first search and thus be used instead.**

Conclusion:

The results above clearly illustrate the benefits of using informed search strategies with custom heuristics over uninformed search techniques when searching for an optimal plan. The benefits are significant both in terms of speed and memory usage. Another, subtler, benefit is that one can customize the trade-off between speed and memory usage by using different heuristics, which is simply not possible with uninformed search strategies.

References:

1. Stuart J. Russell, Peter Norvig (2010), Artificial Intelligence: A Modern Approach (3rd Edition).