Heuristic Analysis

In this analysis the results of running different search stratigies on the 3 cargo problems are presented. The search is performed without and without heuristics to compare the performance of uninformed and informed search algorithms. The metrics prvided are number of node expansions, number of goal test, time elapsed, and optimality of solution.

Problem 1

Algorithm	Nodes	Goal Tests	Time	Optimality
BFS graph	43	56	0.03	Yes
BFS tree	1458	1459	0.9	Yes
DFS graph	21	22	0.01	No
Depth limited	101	271	0.08	No
Uniform Cost	55	57	0.04	Yes
Greedy BFS	7	9	0.005	Yes
A* h_1	55	57	0.04	Yes
A* Ignore Pre	41	43	0.035	Yes
A* Level Sum	11	13	0.6	Yes

DFS is the fastest to find a solution, but the plan is long and not optimal. The fastest to find an optimal solution for this simple problem was greedy BFS. Yet we cannot generalize the results of this problem as it has a small search space.

Optimal Plan

Length: 6 Load(C2, P2, JFK) Load(C1, P1, SFO) Fly(P2, JFK, SFO) Unload(C2, P2, SFO) Fly(P1, SFO, JFK) Unload(C1, P1, JFK)

Problem 2

Algorithm	Nodes	Goal Tests	Time	Optimality
BFS graph	3343	4609	8.5	Yes
BFS tree	N/A	N/A	N/A	N/A
DFS graph	101	271	3.3	No
Depth limited	N/A	N/A	N/A	N/A
Uniform Cost	4852	4854	12	Yes
Greedy BFS	990	992	2.42	No
A* h_1	4852	4854	12	Yes
A* Ignore Pre	1450	1452	4.3	Yes
A* Level Sum	86	88	57.5	Yes

As the search space grows, BFS tree and dpeth limited failed to generate a plan in reasonable time. Again Greedy BFS was the fastest, but failed to generate an optimal plan. With a larger search space we can see the value of good heurisitcis. A* with level sum expanded the least number of nodes, and A* with ignore preconditions was the fastest to find an optimal plan.

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

Algorithm	Nodes	Goal Tests	Time	Optimality
BFS graph	14663	18098	42.2	Yes
BFS tree	N/A	N/A	N/A	N/A
DFS graph	408	409	1.7	No
Depth limited	N/A	N/A	N/A	N/A
Uniform Cost	18235	18237	52	Yes
Greedy BFS	5614	5616	19.16	No
A* h_1	18235	18237	62.5	Yes
A* Ignore Pre	5040	5042	19.65	Yes
A* Level Sum	318	320	339.9	Yes

Problem 3 is more complex with a larger search space, but still much simpler than real world prblems. We can observe that DFS is still the fastest to find a solution, but far from optimal. BFS graph is still able to find an optimal solution in reasonable time, but the fastest to find an optimal solution was A* with ignore precondition, and the algorithm with least node expansions was A* with level sum heuristic.

Optimal Plan

Length: 12Load(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)

Analysis

- As the search space grows slightly BFS tree search, and depth limited failed to find a solution in reasonable time.
- For the three problems presented, DFS was the fastest to prove an existance of a solution, but generated a considerably costly plan.
- Of the uinformed search algorithms, BFS graph was the fastest to find an optimal solution for the three problems. Greedy BFS found a good plan in better time, but not the optimal plan.
- Informed search, with a constant heuristic function, performed like uninformed uniform cost search.
- Informed search with good heuristic performed better than uninformed as the search space grows. Level sum did a good job at guiding the algorithm towards an optimal solution with the least node expansion. Yet the high cost of the level sum calculation made it considerably slower than ignore perconditions.

Conclusion

As stated in the book, relaxing the problem is a good way to come up with a heuristic that does not overestimate the problem. Considering only the positive interactions proved to be a good strategy to find a heuristic for the problems reviewed in this report.