Heuristic Analysis

In this report I analysed three uninformed search methods, namely breadth-first search, depth-first search and uniform cost search, as well as A* using the ignore preconditions and level-sum heuristics. Please refer to Tables 1-3 for results.

As expected, BFS, UCS and A* all find the solution with the optimal path length, namely:

- Problem 1 – six actions
- Problem 2 – nine actions
- Problem 3 – twelve actions

DFS is significantly faster than the other search methods, however it returns the first solution found. This leads to significant deviations from the optimal solution (i.e. for problem 2, DFS returned a path length of 619 compared to the optimal solution of length 6). This is similar to the "Four Corners" problem in the PacMan lab, where DFS found a solution but it was not the optimal one. The reason why DFS does not always find the optimal solution is because it searches the tree depth first and halts as soon as it finds the first viable solution[1].

BFS and UCS require similar time and number of expansions before reaching the solution. This is because the tree is not weighted, and therefore UCS essentially explores the tree in the same way BFS does.

When compared to level-sum, the ignore-preconditions heuristic requires a larger number of expanded nodes (i.e. more memory), while in turn finding a solution in significantly less time. For problem 3, A* with ignore-preconditions finds a solution in 17.98 seconds, compared to 301.2 seconds when using level-sum (see *Table 3*). In other words, A* with the level-sum heuristic uses less memory but requires a longer execution time when compared to the ignore preconditions heuristic. This heuristic is can be inadmissible, but it works well for problems that are largely decomposable[2]. The additional time constraint is because level-sum needs to create a PlanningGraph whereas ignore-preconditions does not.

Search Type	Expansions	Goal Tests	New Nodes	Execution time (s)	Path length
BFS	43	56	180	0.036	6
DFS	21	22	84	0.018	20
UCS	55	57	224	0.043	6
A* Ignore Preconditions	41	43	170	0.038	6
A* level-sum	11	13	50	0.694	6

[1] Artificial Intelligence: A Modern Approach (2010, 3rd Ed.), by S. Russell & P. Norvig, p.87

[2] ibidem, p.402

Table 1. Results for Problem 1

Search Type	Expansions	Goal Tests	New Nodes	Execution time (s)	Path length
BFS	3,343	4,609	30,509	15.75	9
DFS	624	625	5,602	4.208	619
UCS	4,780	4,782	43,381	14.09	9
A* Ignore Preconditions	1,450	1,452	13,303	4.494	9
A* level-sum	86	88	841	63.00	9

Table 2. Results for Problem 2

Search Type	Expansions	Goal Tests	New Nodes	Execution time (s)	Path length
BFS	14,663	18,098	129,631	118.1	12
DFS	408	409	3,364	2.159	392
UCS	17,882	17,884	156,769	59.75	12
A* Ignore Preconditions	5,034	5,036	44,886	17.98	12
A* level-sum	314	316	2,894	301.2	12

Table 3. Results for Problem 3

Optimal sequence for Problem 1 (not unique):
1. Load(C1, P1, SFO)
2. Load(C2, P2, JFK)
3. Fly(P2, JFK, SFO)
4. Unload(C2, P2, SFO)
5. Fly(P1, SFO, JFK)
6. Unload(C1, P1, JFK)

Optimal sequence for Problem 2 (not unique):
1. Load(C1, P1, SFO)
2. Fly(P1, SFO, JFK)
3. Unload(C1, P1, JFK)
4. Load(C2, P2, JFK)
5. Fly(P2, JFK, SFO)
6. Unload(C2, P2, SFO)
7. Load(C3, P3, ATL)
8. Fly(P3, ATL, SFO)
9. Unload(C3, P3, SFO)

Optimal sequence for Problem 3 (not unique):
1. Load(C1, P1, SFO)
2. Fly(P1, SFO, ATL)
3. Load(C3, P1, ATL)
4. Fly(P1, ATL, JFK)
5. Unload(C1, P1, JFK)
6. Load(C2, P2, JFK)
7. Fly(P2, JFK, ORD)
8. Load(C4, P2, ORD)
9. Fly(P2, ORD, SFO)
10. Unload(C2, P2, SFO)

11. Unload(C3, P1, JFK)
12. Unload(C4, P2, SFO)

Bibliography

1. Artificial Intelligence: A Modern Approach (2010, 3rd Ed.), by S. Russell & P. Norvig