

1. Optimal plan for each problems

Breadth First Search (BFS) searches the shortest node among the leaf nodes of the tree in order to guarantee the optimal solution in the planning problem. Therefore, the optimal solution for each problem can be found in the plan found in BFS.

Problem 1

Load(C1, P1, SFO)

Load(C2, P2, JFK)

Fly(P2, JFK, SFO)

Unload(C2, P2, SFO)

Fly(P1, SFO, JFK)

Unload(C1, P1, JFK)

Problem 2

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

Load(C1, P1, SFO)

Fly(P1, SFO, ATL)

Load(C3, P1, ATL)

Fly(P1, ATL, JFK)

Load(C2, P1, JFK)

Unload(C1, P1, JFK)

Unload(C3, P1, JFK)

Fly(P1, JFK, ORD)

Load(C4, P1, ORD)

Fly(P1, ORD, SFO)

Unload(C2, P1, SFO)

Unload(C4, P1, SFO)

2. non-heuristic search

A non-heuristic search searches for a path to a goal state without additional information. The table below shows the experimental results for three non-heuristic searches.

		Optimality	Time Elapsed (sec)	number of node expansions	Plan Length	Goal Tests	New Nodes
BFS	P1	yes	0.04	43	6	56	180
	P2	yes	18.01	3343	9	4609	30509
	P3	yes	59.31	8602	12	11196	64308
DFS	P1	no	0.03	21	20	22	84
	P2	no	4.4	624	619	625	5602
	P3	no	4.46	1292	875	1293	5744
Uniform	P1	yes	0.04	55	6	57	224
	P2	yes	15.48	4853	9	4855	44041
	P3	yes	39.13	11482	12	11484	85785

- BFS found the optimal plan for all problems. It is desirable to consider BFS first in the problem of finding the optimal solution by non-heuristic search.
- Depth First Search (DFS) did not find the optimal plan for all problems. DFS, however, has advantages in terms of storage cost because it first expands the deepest node. In addition, it takes much less computation time than BFS. There is no need to find an optimal solution, and in a system resource limited environment, it may be better to use DFS than to use BFS.
- Uniform Cost Search (UFS), like BFS, found the optimal plan for all problems. For problems where path cost is not the same, it would be better to use UFS instead of BFS. In the same problem of path cost, UFS searches in much the same way as BFS. However, since BFS finds the solution and ends the search immediately, BFS is a better solution than UFS in this case.

3. Heuristic search

Heuristic search is a method to efficiently perform search using knowledge of the problem. The table below shows the experimental results for two heuristic searches.

		Optimality	Time Elapsed (sec)	number of node expansions	Plan Length	Goal Tests	New Nodes
precondition	P1	yes	0.05	41	6	43	170
	P2	yes	5.62	1450	9	1450	13303
	P3	yes	16.67	4119	12	4121	31485
level sum	P1	yes	1.27	39	6	41	158
	P2	yes	481.02	1129	9	1131	10232
	P3	yes	1134.46	1958	12	1960	15626

- ignore preconditions The heuristic makes the problem easy by ignoring all preconditions for the action and counts the minimum number of steps to reach the goal state. (AIMA 10.2.3) A*-search using Ignore preconditions heuristics found an optimal solution for all problems. It is the same as always that the optimal solution is guaranteed compared to BFS, but the more complex the problem, the better the performance in time and space complexity. BFS exponentially increases the space and time complexity at the depth of the tree. (AIMA 3.4.1) Therefore, A*-search using Ignore precondition heuristics is a better choice than BFS for large step from initial state to goal state.
- A*-search using level sum heuristic found the optimal solution for all problems. This strategy has shown strength in terms of storage cost. In addition to BFS, the number of expansion nodes was much smaller than that of A*-search using Ignore precondition heuristics. On the other hand, it showed very poor performance in terms of time complexity. In all three cases the search time was longer than BFS. A*-search using level sum heuristics will be a good method to use when you need to find an optimal solution in a space complexity environment.

4. Best Heuristics in the problem

The best heuristic differs depending on which of the time complexity and the space complexity is prioritized, but if two factors are considered together, the ignorant precondition heuristic is better heuristic than the level sum heuristic.

First, in terms of time complexity, ignore preconditions heuristic showed better performance than level sum heuristic, and both time complexity and space complexity showed better performance than non-heuristic search that guarantees optimal solution. Therefore, A*-search with ignore preconditions heuristic can always be said to guarantee better performance than non-heuristic search.

In terms of space complexity, the level sum heuristic showed better performance than the ignore precondition heuristic. However, A*-search using level sum heuristics showed worse performance than non-heuristic search in terms of time complexity.