Heuristic Report - Implementing A Planning Search

Introduction

In this project we were tasked with solving the air cargo system's transportation problems. We used a planning graph and an automatic domain independent heuristic with A* search to solve the problems, then compared and contrasted the performance results against non-heuristic methods. The project was broken down to two parts, the first part consisted of defining problems using PDDL (Planning Domain Definition Language), and the second part included implementing domain-independent heuristics (breadth-first and depth-first search).

Planning Problems

We were given the initial states and goals of each problem.(Note, that all three problems follow the same Air Cargo Action Schema).

Problem 1	Problem 2	Problem 3
Init(At(C1, SFO) ∧ At(C2, JFK) ∧ At(P1, SFO) ∧ At(P2, JFK) ∧ Cargo(C1) ∧ Cargo(C2) ∧ Plane(P1) ∧ Plane(P2) ∧ Airport(JFK) ∧ Airport(SFO)) Goal(At(C1, JFK) ∧ At(C2, SFO))	Init(At(C1, SFO) ∧ At(C2, JFK) ∧ At(C3, ATL) ∧ At(P1, SFO) ∧ At(P2, JFK) ∧ At(P3, ATL) ∧ Cargo(C1) ∧ Cargo(C2) ∧ Cargo(C3) ∧ Plane(P1) ∧ Plane(P2) ∧ Plane(P3) ∧ Airport(JFK) ∧ Airport(SFO) ∧ Airport(ATL)) Goal(At(C1, JFK) ∧ At(C2, SFO) ∧ At(C3, SFO))	Init(At(C1, SFO) ∧ At(C2, JFK) ∧ At(C3, ATL) ∧ At(C4, ORD) ∧ At(P1, SFO) ∧ At(P2, JFK) ∧ Cargo(C1) ∧ Cargo(C2) ∧ Cargo(C3) ∧ Cargo(C4) ∧ Plane(P1) ∧ Plane(P2) ∧ Airport(JFK) ∧ Airport(SFO) ∧ Airport(ATL) ∧ Airport(ORD)) Goal(At(C1, JFK) ∧ At(C3, JFK) ∧ At(C2, SFO) ∧ At(C4, SFO))

We were asked to find provide metrics on the number of node expansions required, goal tests required and optimal solution for each search algorithm.

Part A - Uninformed Search Strategy

Uninformed search strategy is a strategy that has no additional information about its states beyond what is provided in the problem definition. They are used to generate successors by searching the entire search space, all while checking to see if any states are goal states or non goal states.

Problem 1:

Search Strategy	Expansions	Goal Tests	New Nodes	Plan Length	Time Elapsed
Breadth First Search	43	56	180	6	0.05
Breadth First Tree Search	1458	1459	5960	6	1.2
Depth First Graph Search	12	13	48	12	0.01
Depth Limited Search	101	271	414	50	0.11

Uniform Cost Search	55	57	224	6	0.5
Recursive Best First Search	4429	4230	17029	6	3.50
Greedy Best First Search Graph	**7**	**9**	**28**	**6**	**0.01**

As we can see from the table, Greedy Best First search provides an optimal solution with a plan length of 6 (because it has the shortest plan length,expansions, goal tests, and time) and a sample plan as follows:

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

Problem 2:

Search Strategy	Expansions	Goal Tests	New Nodes	Plan Length	Time Elapsed
Breadth First Search	3401	4627	31049	9	19.80
Breadth First Tree Search	-	-	-	-	-
Depth First Graph Search	350	351	3142	346	1.89
Depth Limited Search	-	-	-	-	-
Uniform Cost Search	4761	4763	43206	9	20.46
Recursive Best First Search	-	-	-	-	-
Greedy Best First Search Graph	550	552	4950	9	2.72

As we can see from the table, the Breadth First Tree search, Depth Limited Search, and Recursive Best First Search have no values. This is because they were taking longer than 10 minutes to run. However, it is clear to deduce that the Greedy Best First search provides an optimal solution(because it has the shortest plan length,expansions, goal tests, and time) with a plan length of 9 and a sample plan as follows:

Load(C1, P1, SFO)
Load(C2, P2, JFK)
Load(C3, P3, ATL)
Fly(P1, SFO, JFK)
Fly(P2, JFK, SFO)
Fly(P3, ATL, SFO)
Unload(C3, P3, SFO)
Unload(C2, P2, SFO)
Unload(C1, P1, JFK)

Problem 3:

Search Strategy	Expansions	Goal Tests	New Nodes	Plan Length	Time Elapsed
Breadth First Search	14491	17947	128184	12	151.23
Breadth First Tree Search	-	-	-	-	-

Depth First Graph Search	1948	1949	16253	1878	35.34
Depth Limited Search	-	-	-	-	-
Uniform Cost Search	17783	17785	155920	12	70.25
Recursive Best First Search	-	-	-	-	-
Greedy Best First Search Graph	4031	4033	35794	22	17.30

As we can see from the table,Depth Limited Search, and Recursive Best First Search have no values. This is because they were taking longer than 10 minutes to run. However, it is clear to deduce that the Breadth First Search provides an optimal solution; because it has the shortest plan length,expansions, goal tests, and time. Unlike the last two problems, however if optimal length was not a primary criteria, we could choose Greedy Best First Search as it has the lowest execution time. Note, the Breadth First Search has a plan length of 12 and a sample plan as follows:

Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P1, SFO, ATL)
Load(C3, P1, ATL)
Fly(P2, JFK, ORD)
Load(C4, P2, ORD)
Fly(P1, ATL, JFK)
Fly(P2, ORD, SFO)
Unload(C4, P2, SFO)
Unload(C3, P1, JFK)
Unload(C2, P2, SFO)
Unload(C1, P1, JFK)

Part B- Informed Search Strategy

An informed search strategy is a strategy that has knowledge about its states beyond what is provided in the problem definition.

Problem 1:

Search Strategy	Expansions	Goal Tests	New Nodes	Plan Length	Time Elapsed
A* Search H1	55	57	224	6	0.05
A* Search Heuristic Ignore Predictions	41	43	170	6	0.06
A* Search Heuristic Level Sum	11	13	50	6	1.61

As we can see from the table, A* Search Heuristic Level sum provides an optimal solution with a plan length of 6 (because it has the shortest plan length,expansions, goal tests, and time) and a sample plan as follows:

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

Problem 2:

Search Strategy	Expansions	Goal Tests	New Nodes	Plan Length	Time Elapsed
A* Search H1	4761	4763	43206	9	15.37
A* Search Heuristic Ignore Predictions	1450	1452	13303	9	7.14
A* Search Heuristic Level Sum	86	88	841	9	317.34

As we can see from the table, A* Search Heuristic Level sum provides an optimal solution with a plan length of 9 (because it has the shortest plan length,expansions, goal tests, and time) and a sample plan as follows:

Load(C1, P1, SFO)
Fly(P1, SFO, JFK)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
Load(C3, P3, ATL)
Fly(P3, ATL, SFO)
Unload(C3, P3, SFO)
Unload(C2, P2, SFO)
Unload(C1, P1, JFK)

Problem 3:

Search Strategy	Expansions	Goal Tests	New Nodes	Plan Length	Time Elapsed
A* Search H1	17783	17785	155920	12	65.15
A* Search Heuristic Ignore Predictions	5003	5005	44586	12	26.63
A* Search Heuristic Level Sum	-	-	-	-	-

As we can see from the table, A* Search Heuristic Ignore Predictions provides an optimal solution with a plan length of 9 (because it has the shortest plan length,expansions, goal tests, and time) and a sample plan as follows:

Load(C2, P2, JFK)
Fly(P2, JFK, ORD)
Load(C4, P2, ORD)
Fly(P2, ORD, SFO)
Unload(C4, P2, SFO)
Load(C1, P1, SFO)
Fly(P1, SFO, ATL)
Load(C3, P1, ATL)
Fly(P1, ATL, JFK)
Unload(C3, P1, JFK)
Unload(C2, P2, SFO)
Unload(C1, P1, JFK)

However, the A* Search Heuristic Level Sum search approach shows no values because it took longer than 10 mins.

Part C - Comparing Uninformed and Informed Search Strategies

The optimal Solution in both approaches are compared

Problem 1:

Search Strategy	Expansions	Goal Tests	New Nodes	Plan Length	Time Elapsed
Greedy Best First search	**7**	**9**	**28**	**6**	**0.01**
A* Search Heuristic Level Sum	86	88	841	9	317.34

Problem 2:

Search Strategy	Expansions	Goal Tests	New Nodes	Plan Length	Time Elapsed
Greedy Best First Search Graph	550	552	4950	9	**2.72**
A* Search Heuristic Level Sum	**11**	**13**	**50**	**6**	**1.61**

Problem 3:

Search Strategy	Expansions	Goal Tests	New Nodes	Plan Length	Time Elapsed
Breadth First Search	14491	17947	128184	12	151.23
A* Search Heuristic Ignore Predictions	**5003**	**5005**	**44586**	**12**	**26.63**

The results above show the benefits of using informed and uninformed search strategies. For problem one, an uninformed search strategy showed faster results and smaller expansions; proving that it is ideal in terms of speed and memory usage.

For problems two and three, informed search strategies showed better results in memory and time. The reason A* search performed better is because the A* search is a complete, optimal and efficient approach. This is because for every node n and every successor n of n generated by any action a, the estimated cost of reaching the goal from n is no greater than the step cost of getting to n plus the estimated cost of reaching the goal from n (Artificial Intelligence A Modern Approach,3rd edition Norvig and Russell, page 98). This means, the algorithm will be capable of leading the search in a good direction, all while allowing for a faster search performance; hence a more accurate optimal solution.
It is important to note that it is common to ignore predictions in search planning (Artificial Intelligence A Modern Approach,3rd edition Norvig and Russell, page 376), this allows us to create relaxed constraints that can help us solve the problem faster by examining lesser state spaces at the same cost. For these reasons, I believe that the A* search approach is ideal and can work with most heuristics to produce an optimal value.