UdacityArtificial Intelligence Nanodegree
Cory Robbins
December 2022

Project 2: Build a Forward Planning Agent

Air Cargo Problem 1:
The table below shows all possible combinations of search and heuristic search methods

	Breadth First	Depth First	Uniform Cost	GBF Unmet	GBF LevelSum	GBF MaxLevel
# Actions	20	20	20	20	20	20
Expansions	43	21	60	7	6	6
Goal Tests	56	22	62	9	8	8
New Nodes	178	84	240	29	24	28
Time elapsed (sec.)	.00138	.00075	.00205	.06747	.04765	.15027
Plan length	6	20	6	6	6	6
Nodes expanded per action	8.9	4.2	12	1.45	1.4	1.2
Time per action	.000069	.00003	.00010	.00337	.002382	.0075135

	GBF SetLevel	A* Unmet	A* LevelSum	A* MaxLevel	A* SetLevel
# Actions	20	20	20	20	20
Expansions	6	50	28	43	30
Goal Tests	8	52	30	45	35
New Nodes	28	206	122	180	135
Time elapsed (sec.)	.15027	.00205	.16450	.17127	.03640
Plan length	6	6	6	6	6
Nodes expanded per action	1.4	10.3	6.1	9	6.9
Time per action	.00751	.000102	.008225	.00856	.00182

Air Cargo Problem 2:
The table below shows all possible combinations of search and heuristic search methods

	Breadth First	Depth First	Uniform Cost	GBF Unmet	GBF LevelSum	GBF MaxLevel
# Actions	72	72	72	72	72	72
Expansions	3343	624	5154	17	9	27
Goal Tests	4609	625	5156	19	11	29
New Nodes	30503	5602	46618	170	86	249
Time elapsed (sec.)	.44784	.60971	.75080	.00423	1.5265	3.14345
Plan length	9	619	9	9	9	9
Nodes expanded per action	423.65	77.805	647	2.361	1.194	3.458
Time per action	.00622	.00846	.01042	.000058	.02120	.04365

	GBF SetLevel	A* Unmet	A* LevelSum	A* MaxLevel	A* SetLevel
# Actions	72	72	72	72	72
Expansions	9	2467	357	2887	1037
Goal Tests	11	2469	359	2889	1039
New Nodes	84	22522	3426	26594	9605
Time elapsed (sec.)	3.9176	.49966	40.3917	251.977	322.5062
Plan length	9	9	9	9	9
Nodes expanded per action	1.166	312.805	47.583	369.361	133.402
Time per action	.05441	.00693	.56099	3.4996	4.47925

Air Cargo Problem 3:
The table below shows uniformed search methods Depth First Search and heuristic search methods with Greedy Best First Unmet and LevelSum as well as Astar heuristics Unmet and Levelsum search methods

	Depth First	GBF Unmet	GBF LevelSum	A* Unmet	A* LevelSum
# Actions	88	88	88	88	88
Expansions	408	25	14	7388	369
Goal Tests	409	27	16	7390	371
New Nodes	3364	230	126	65711	3403
Time elapsed (sec.)	.23537	.00752	3.44708	1.78671	63.36962
Plan length	392	15	14	12	12
Nodes expanded per action	1.166	312.805	47.583	369.361	133.402
Time per action	.00267	.000085	.039171	.020303	.072010

Air Cargo Problem 4:
The table below shows uniformed search methods Depth First Search and heuristic search methods with Greedy Best First Unmet and LevelSum as well as Astar heuristics Unmet and Levelsum search methods

	Depth First	GBF Unmet	GBF LevelSum	A* Unmet	A* LevelSum
# Actions	104	104	104	104	104
Expansions	25174	29	17	34330	1208
Goal Tests	25175	31	19	34332	11210
New Nodes	228849	280	165	328509	12210
Time elapsed (sec.)	752.15501	.01261	6.2100	11.8509	344.8501
Plan length	24132	18	17	17	15
Nodes expanded per action	9078.173	2.6923	1.5865	3158.74	.1442
Time per action	7.2322	.0001212	.05971	.11395	3.3158

Finding Results and Analysis:
For our findings we aim to optimize the best search solutions for the four air cargo problems/ By comparing the eleven search methods we arrange them complexity and the various tradeoffs including space and time complexities to analyze the optimality of solution as a function of domain size, search algorithm, and heuristic.

Time per action is the best measurement for complexity in each of the air cargo problems.
For the air cargo problem number one uniformed Depth First search showed the best results in regards to performance against all other methods as the time per action metric was minimized. The Greedt Best First 'MaXlevel' and 'LevelSum' heuristic method is the best in regards to nodes expanded per action metric

For the air cargo problem number two the Greedy Best First time per action ins the best method while nodes expanded per action metric along with the Greedy Best First 'LevelSum heuristic methos is the most effective as DFS performs least effective and increasing the plan length many times than problem one.

For the air cargo problem number three, the Greedy Best First 'Unmet' heuristic method performed the best while time per action was quite low. Nodes expanded per acton is comparable with Greedy Best First 'LevelSum and Greedy Best First 'Unmet'.

For the air cargo problem number four the greedy Best First 'Unmet' heuristic method required the leasy amount of time lapse with .012 seconds and pferomed the best with metrics time per action.0001212 . Nodes expanded per action with a trade off of .1442.

Questions:

1. **Which algorithm or algorithms would be most appropriate for planning in a very restricted domain (i.e., one that has only a few actions) and needs to operate in real time?**

The algorithm which took the lest time lapses was the Greedy Best First 'LevelSum' heuristic for each air cargo problem making it the best method. Furthermore the time per action was the fastest while having low nodes expanded action metric which implies space efficiency.

2. **Which algorithm or algorithms would be most appropriate for planning in very large domains (e.g., planning delivery routes for all UPS drivers in the U.S. on a given day)**

Greedy Best First 'Unmet' heuristic as observed in air cargo problem number four, the method takes the elst amount of time and with the lowest time per action amount while having the elst amount of expansions and new nodes.

3. **Which algorithm or algorithms would be most appropriate for planning problems where it is important to find only optimal plans?**

To me, in my analysis The Greedy Best First 'Unmet' algorithm heuristic is the more appropriate planning problem method as it consistently results in low time and minimized plan length while trade offs in complexity including time per action and time elapsed are also minimized in all problems.