April 2020

Udacity
Artificial Intelligence
Nanodegree

Project 2:

Build A Forward-Planning Agent

Akshit Sarin

- **Air Cargo Problem 1:**
  Table showing the all possible combinations of search and heuristics -

	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	28	24
Time Elapsed (sec)	0.006453378	0.003381086	0.00961057	0.001715472	0.00168982	0.136159524
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	**0.00032266**	**0.0016905**	**0.00048052**	**0.000855**	**0.0008445**	**0.0068079**

	GBF SetLevel	A* Unmet	A* LevelSum	A* MaxLevel	A* SetLevel
Actions	20	20	20	20	20
Expansions	6	50	28	43	33
Goal Tests	8	52	30	45	35
New Nodes	28	206	122	180	138
Time Elapsed (sec)	0.511180912	0.009725352	0.456106507	0.468418286	1.203204252
Plan Length	**6**	**6**	**6**	**6**	**6**
Nodes Expanded per Action	**1.4**	**10.3**	**6.1**	**9**	**6.9**
Time Per Action	**0.025559**	**0.000486**	**0.022805**	**0.0234209**	**0.06016**

- **Air Cargo Problem 2:**
  Table showing the all possible combinations of search and heuristics -

	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)	2.062067352	3.191628075	3.405826192	0.019362964	3.953807891	6.18403265
Plan Length	**9**	**619**	**9**	**9**	**9**	**9**
Nodes Expanded per Action	**423.65**	**77.805**	**647.47**	**2.361**	**1.194**	**3.458**
Time Per Action	**0.0286**	**0.0443**	**0.04703**	**0.00026**	**0.05491**	**0.0858**

	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)	12.64426743	2.310200204	105.5288498	501.238922	2846.11284
Plan Length	**9**	**9**	**9**	**9**	**9**
Nodes Expanded per Action	**1.166**	**312.805**	**47.583**	**369.361**	**133.402**
Time Per Action	**0.1756**	**0.3208**	**0.363**	**6.961**	**39.529**

- **Air Cargo Problem 3:**
  Table showing one uninformed search (DFS), two heuristics with greedy best first search (Unmet and LevelSum), and two A* heuristics (Unmet, LevelSum).

	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)	1.1876	0.0365	9.3393	8.55104	203.3263822
Plan Length	**392**	**15**	**14**	**12**	**12**
Nodes Expanded per Action	**38.227**	**2.613**	**1.431**	**746.71**	**34.373**
Time Per Action	**0.0134**	**0.000414**	**0.10612**	**0.09717**	**2.3105**

- **Air Cargo Problem 4:**
  Table showing one uninformed search (BFS), two heuristics with greedy best first search (Unmet and LevelSum), and two A* heuristics (Unmet, LevelSum).

	Breadth First	GBF Unmet	GBF LevelSum	A* Unmet	A* LevelSum
Actions	104	104	104	104	104
Expansions	99736	29	17	34330	1208
Goal Tests	114953	31	19	34332	1210
New Nodes	944130	280	165	328509	15
Time Elapsed (sec)	99.095	0.06128	16.349	56.831	254.63
Plan Length	**14**	**18**	**17**	**14**	**15**
Nodes Expanded per Action	**9078.173**	**2.6923**	**1.5865**	**3158.74**	**0.1442**
Time Per Action	**0.9528**	**0.00058**	**0.1572**	**0.5464**	**2.44**

Findings / Analysis -

We try to find the optimal solution for each of the four problems, by analysing and comparing various methods/algorithms. The problems are arranged in an increasing order of complexity. An optimal solution is based on certain factors, considering and selecting trade-offs between space or time complexities. The same are analyzed below -

For the **first** problem, considering the 'time per action' metric, uninformed DFS performs the best amongst all, whereas considering 'nodes expanded per action' metric, Greedy Best First with 'MaxLevel' heuristic outshines all other methods, with a nodes expanded per action ratio of just 1:2. But considering both the metrics, the most fulfilling method is Greedy Best First along with 'LevelSum' heuristic.

For the **second** problem, considering the 'time per action' metric, Greedybest First performed the best out of all techniques, whereas considering 'nodes expanded per action' metric, Greedy Best First with 'LevelSum' heuristic was the obvious winner, with a ratio of just 1:194. Another interesting thing to note is that DFS performs very poorly, with the 'plan length' showing an increase of 6777.77%, compared to all other plan lengths.

For the **third** problem, Greedy Best First with 'Unmet' heuristic performed outstandingly well, having a super low value of 'time per action', being only 0.000414. It's performance measure for the other metric, 'nodes expanded per action', was also comparable to the most optimal one (Greedy Best First with LevelSum - 1.431 & GBF Unmet - 2.613)

For the **fourth** problem, GBF with 'Unmet' heuristic took the least amount of time (0.06 sec), and performed the best under the criteria 'time per action' - 0.00058. A* with LevelSum performed the best under 'nodes expanded per action' - 0.1442. Uninformed Search (BFS) performed the worst, having the most (and considerably high) number of expansions and new nodes created.

Throughout all problems, Greedy Best First approach, along with 'LevelSum' heuristic has been a constant achiever, giving the optimal results with very low tradeoffs. Also, generally, DFS gives a very large number of plan lengths, as compared to all other techniques.

As the complexities of problems increased, a gradual pattern emerged amongst the different approaches -

- The uninformed search techniques start giving non optimal results, having a very high number of expansions and created nodes, which implies they have high space complexities. Whereas, the Greedy and A* techniques hold good for increasing complexities in terms on space complexities.
- Another pattern that has emerged is that the A* technique with 'SetLevel' heuristic has a very high time complexity, since the time taken to get to the solution increases drastically, with respect to increase in problem's complexities. Such is the case with uninformed searches, where too, the time-complexities are very high.
- Considering the 'Plan lengths' of the solutions, there is a general increase as complexity increases, but are comparable amongst all other techniques except DFS. DFS's plan lengths increase very drastically as the complexity increases.

Questions / Answers -

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?**

   Greedy Best First with 'LevelSum' heuristic is the best choice for such situations, since having a low 'time taken to find solution' and 'time per action' imply that it works very fast, which is ideal for programs that function in real-time. Also, having a low 'nodes expanded per action' implies that it is space efficient.

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)**

   As seen in problem 4, when the domain is large, Greedy Best First with 'Unmet' heuristic, since it takes the least time to arrive at a solution and has the lowest time per action value. It also has a very less 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?**

   According to me, A* algorithm with 'Unmet' heuristics must be used, since it consistently shows a low value of plan length throughout the problems with increasing complexities, and it also has a very low 'time per action' value.