Forward Planning: Air Cargo Analysis

Adam DeJans

Analysis of Search Complexity

Algorithm	Problem	Actions	Expansions	Goal Tests	New Nodes
BFS	1	20	43	56	178
DFS	1	20	21	22	84
Uniform Cost	1	20	60	62	240
Greedy - unmet goals	1	20	7	9	29
Greedy - level sum	1	20	6	8	28
Greedy - max level	1	20	6	8	24
Greedy - set level	1	20	6	8	28
A* - unmet goals	1	20	50	52	206
A* - level sum	1	20	28	30	122
A* - max level	1	20	43	45	180
A* - set level	1	20	33	35	138
BFS	2	72	3343	4609	30503
DFS	2	72	624	625	5602
Uniform Cost	2	72	5154	5156	46618
Greedy - unmet goals	2	72	17	19	170
Greedy - level sum	2	72	9	11	86
Greedy - max level	2	72	27	29	249
Greedy - set level	2	72	9	11	84
A* - unmet goals	2	72	2467	2469	22522
A* - level sum	2	72	357	359	3426
A* - max level	2	72	2887	2889	26594
A* - set level	2	72	1037	1039	9605
BFS	3	88	14663	18098	129625
Greedy - unmet goals	3	88	25	27	230
Greedy - level sum	3	88	14	16	126
A* - unmet goals	3	88	7388	7390	65711
A* - level sum	3	88	369	371	3403
BFS	4	104	99736	114953	944130
Greedy - unmet goals	4	104	29	31	280
Greedy - level sum	4	104	17	19	165
A* - unmet goals	4	104	34330	34332	328509
A* - level sum	4	104	1208	1210	12210

Table 1. Forward planning algorithm comparisons on air cargo problems.

Greedy algorithms have the least search complexity by far at all problem difficulty levels. On simple problems the best uninformed search (BFS) is relatively the same as A*. As the the problem size increases we see that A* ends up being increasingly better than the uninformed search.

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? Since the problem is assumed to be in a restricted domain, we can use our data from problem set 1 (and 2) to make a decision. Since the algorithm needs to operate in real-time, we need the algorithm to be fast. Thus, we can choose the fastest algorithm in problem instance 1 and 2 and extrapolate the answer to this question to be: Greedy - unmet goals.

Analysis of Search Time

Algorithm	Problem	Actions	Expansions	Goal Tests	New Nodes	Time Elapsed
BFS	1	20	43	56	178	0.0065
DFS	1	20	21	22	84	0.0035
Uniform Cost	1	20	60	62	240	0.0094
Greedy - unmet goals	1	20	7	9	29	0.0017
Greedy - level sum	1	20	6	8	28	0.1117
Greedy - max level	1	20	6	8	24	0.0753
Greedy - set level	1	20	6	8	28	0.3362
A* - unmet goals	1	20	50	52	206	0.0091
A* - level sum	1	20	28	30	122	0.2547
A* - max level	1	20	43	45	180	0.2868
A* - set level	1	20	33	35	138	1.5622
BFS	2	72	3343	4609	30503	1.9875
DFS	2	72	624	625	5602	2.9462
Uniform Cost	2	72	5154	5156	46618	3.2727
Greedy - unmet goals	2	72	17	19	170	0.0189
Greedy - level sum	2	72	9	11	86	2.0140
Greedy - max level	2	72	27	29	249	2.9780
Greedy - set level	2	72	9	11	84	7.7496
A* - unmet goals	2	72	2467	2469	22522	2.0811
A* - level sum	2	72	357	359	3426	51.4442
A* - max level	2	72	2887	2889	26594	556.0864
A* - set level	2	72	1037	1039	9605	938.0391
BFS	3	88	14663	18098	129625	10.5229
Greedy - unmet goals	3	88	25	27	230	0.0791
Greedy - level sum	3	88	14	16	126	4.7372
A* - unmet goals	3	88	7388	7390	65711	8.0412
A* - level sum	3	88	369	371	3403	156.3867
BFS	4	104	99736	114953	944130	94.0430
Greedy - unmet goals	4	104	29	31	280	0.1089
Greedy - level sum	4	104	17	19	165	8.2514
A* - unmet goals	4	104	34330	34332	328509	112.3878
A* - level sum	4	104	1208	1210	12210	551.5090

Table 2. Forward planning algorithm comparisons on air cargo problems.

On the smallest problem instances we don't see much of a difference in the time it takes per solve. However, as the problem grows, we see that the greedy approach is by far the fastest. We also see that although the search complexity is reduced in the informed searches, we pay with the trade-off of higher search times. This seems counter-intuitive.

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) Since the problem is assumed to have a very large domain, we can use our data from problem set 3 (and 4) to make a decision. If we are concerned about time we can use an A* algorithm (unmet goals); however, if the business problem requires us to have an optimal solution we may consider using BFS instead.

Analysis of Optimality

Algorithm	Problem	Expansions	Goal Tests	New Nodes	Time Elapsed	Plan Length
BFS	1	43	56	178	0.0065	6
DFS	1	21	22	84	0.0035	20
Uniform Cost	1	60	62	240	0.0094	6
Greedy - unmet goals	1	7	9	29	0.0017	6
Greedy - level sum	1	6	8	28	0.1117	6
Greedy - max level	1	6	8	24	0.0753	6
Greedy - set level	1	6	8	28	0.3362	6
A* - unmet goals	1	50	52	206	0.0091	6
A* - level sum	1	28	30	122	0.2547	6
A* - max level	1	43	45	180	0.2868	6
A* - set level	1	33	35	138	1.5622	9
BFS	2	3343	4609	30503	1.9875	9
DFS	2	624	625	5602	2.9462	619
Uniform Cost	2	5154	5156	46618	3.2727	9
Greedy - unmet goals	2	17	19	170	0.0189	9
Greedy - level sum	2	9	11	86	2.0140	9
Greedy - max level	2	27	29	249	2.9780	9
Greedy - set level	2	9	11	84	7.7496	9
A* - unmet goals	2	2467	2469	22522	2.0811	9
A* - level sum	2	357	359	3426	51.4442	9
A* - max level	2	2887	2889	26594	556.0864	9
A* - set level	2	1037	1039	9605	938.0391	9
BFS	3	14663	18098	129625	10.5229	12
Greedy - unmet goals	3	25	27	230	0.0791	15
Greedy - level sum	3	14	16	126	4.7372	14
A* - unmet goals	3	7388	7390	65711	8.0412	12
A* - level sum	3	369	371	3403	156.3867	12
BFS	4	99736	114953	944130	94.0430	14
Greedy - unmet goals	4	29	31	280	0.1089	18
Greedy - level sum	4	17	19	165	8.2514	17
A* - unmet goals	4	34330	34332	328509	112.3878	14
A* - level sum	4	1208	1210	12210	551.5090	15

Table 3. Forward planning algorithm comparisons on air cargo problems.

The first observation we see is that BFS always gives an optimal solution. Secondly we see that DFS rarely gives an optimal solution - in fact for these problem instances the plan length was much, much larger than the other algorithms. In the largest instance we see that using A* with unmet goals is the only optimal informed algorithm that was chosen. Lastly we notice that A* is always closer to optimal than greedy - in particular when the problem instance grows in size.

Which algorithm or algorithms would be most appropriate for planning problems where it is important to find only optimal plans? In our case BFS was the only high level algorithm to give optimal solution in all instances. A* with unmet goals heuristic also gives an optimal solution so we could consider this as well.

https://www.adamdejans.com Detroit, MI adam.dejans@gmail.com