Udacity AI Nanodegree Project 2 Report

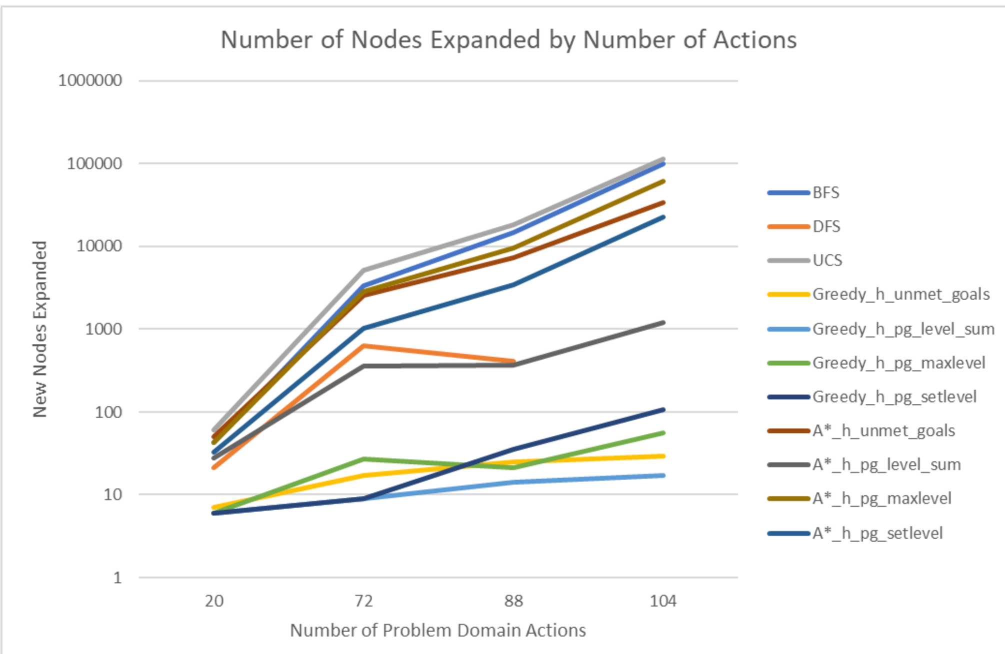
Kenny Carlsen

3/3/21

General Note: A plot of relevant data is provided in each section. Additional source data/plots are provided at the end of the report.

Note on algorithms executed: All algorithms were evaluated for all problems, however, due to an error in extracting the data for DFS on problem 4 the results for that particular algorithm/problem combination are not included here.

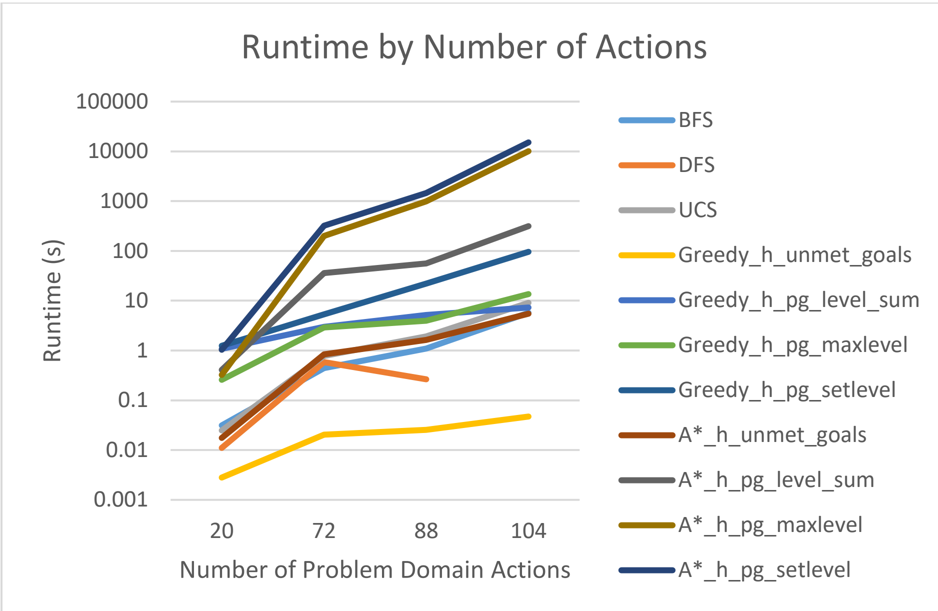
Section 1: Search Complexity as a Function of Domain Size



Note: Source data available in table form at the end of this report

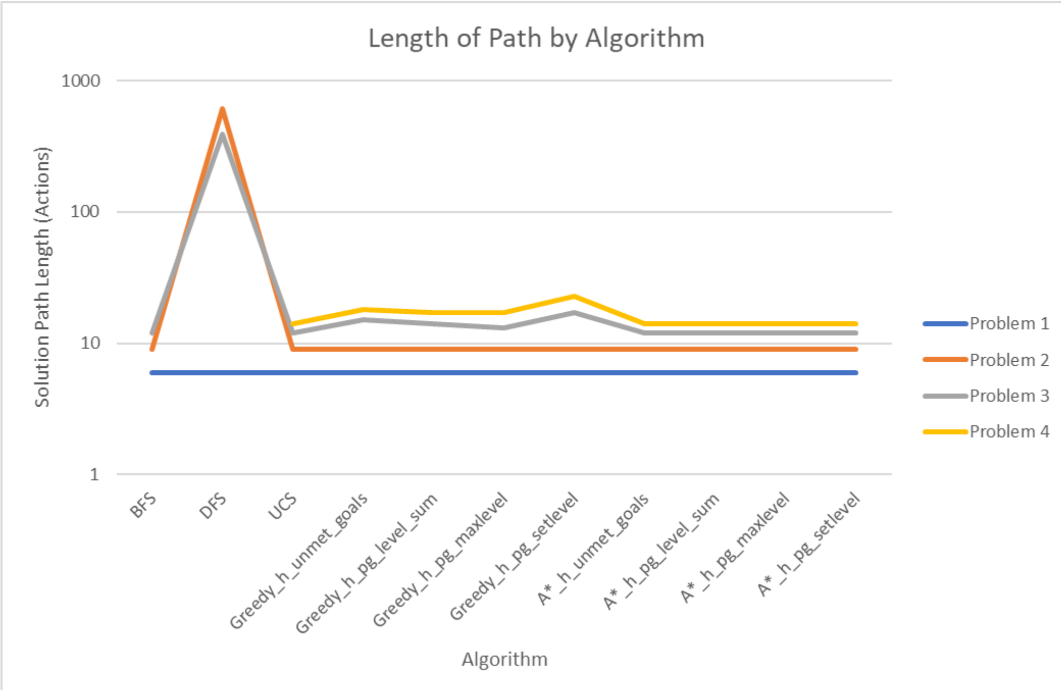
In general across all greedy_level_sum performed the best in terms of number of new nodes expanded as a function of number of problem domain actions, with greedy_setlevel performing comparably for “low” problem domain actions and greedy_maxlevel and greedy_unmet_goals performing comparably for “high” problem domain actions. The results above appear linear on a log-lin plot indicating that these algorithms grow exponentially as a function of problem domain actions, but that the greedy algorithms grow at a slower exponential rate than A* or uniformed counterparts. Interestingly, DFS performs better on the 88 action domain problem than the 72, likely due to the specific set of actions and goals in the domain which produce a favorable set of conditions for DFS performance.

Section 2: Search Time as a Function of Domain Size



As in search complexity, greedy_unmet_goals again performs the best of the algorithms evaluated indicating that the problem space contains few or no 'action traps' which can cause greedy to sub-optimally meet goals to the point of a significant runtime penalty. It's also interesting to note that the uniformed searches generally scaled better than the informed searches in terms of runtime (with the exception of greedy_unmet_goals).

Section 3: Optimality of Solution by Algorithm



Most algorithms (excluding DFS due to its nature of running down tree branches searching for solutions before evaluating 'simpler', less deep solutions) performed comparably, reaching the same optimal answer in most cases. Notable exceptions to this behavior occur on the more complex problem spaces (problem 3 and problem 4) when using greedy algorithms. This is due to properties of the domain which, when decisions are made with greedy algorithms relative to the current position in a graph space, produce solutions which require more nodes than other solutions present elsewhere on the graph which stem from alternate nodes not currently under evaluation. Also note that the A* algorithms consistently find the optimal solution across all domains at the cost of additional run-time and more node expansions (see questions 1 and 2).

Section 4: Short Answer Questions

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 algorithms (level_sum in particular) would perform most optimally in a very restricted, real-time required domain due to their low node expansion, fast run-time, and their downside (potentially sub-optimal solutions) being mitigated by a restricted search space.

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)

Uninformed searches would perform well in large problem domains (specifically BFS and UCS) due to their ability to reach an optimal or nearly-optimal answer in a relatively short period of time.

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

A* algorithms are the best choice when strict optimality is required as they perform optimally across all use cases evaluated at the expense of a longer run-time.

Additional Source Data:

Problem 1 Results

		# Actions	New Node Expansions	Goal Tests	New Nodes	Runtime	Plan Length
	BFS	20	43	56	178	0.031326	6
	DFGS	20	21	22	84	0.01106	6
	UCS	20	60	62	240	0.024795	6
Greedy BFGS	h_unmet_goals	20	7	9	29	0.002807	6
	h_pg_level_sum	20	6	8	28	1.080178	6
	h_pg_maxlevel	20	6	8	24	0.257626	6
	h_pg_setlevel	20	6	8	28	1.245515	6
A* Search	h_unmet_goals	20	50	52	206	0.017539	6
	h_pg_level_sum	20	28	30	122	0.407583	6
	h_pg_maxlevel	20	43	45	180	0.322959	6
	h_pg_setlevel	20	33	35	138	1.034757	6

Problem 2 Results

		# Actions	New Node Expansions	Goal Tests	New Nodes	Runtime	Plan Length
	BFS	72	3343	4609	30503	0.441276	9
	DFGS	72	624	625	5602	0.584682	619
	UCS	72	5154	5156	46618	0.778457	9
Greedy BFGS	h_unmet_goals	72	17	19	170	0.020367	9
	h_pg_level_sum	72	9	11	86	2.987734	9
	h_pg_maxlevel	72	27	29	249	2.899926	9
	h_pg_setlevel	72	9	11	84	5.32333	9
A* Search	h_unmet_goals	72	2567	2469	22522	0.839894	9
	h_pg_level_sum	72	357	359	3426	35.831352	9
	h_pg_maxlevel	72	2887	2889	26594	200.482075	9
	h_pg_setlevel	72	1037	1039	9605	320.043721	9

Problem 3 Results

		# Actions	New Node Expansions	Goal Tests	New Nodes	Runtime	Plan Length
	BFS	88	14663	18098	129625	1.089955	12
	DFGS	88	408	409	3364	0.26323	392
	UCS	88	18510	18512	161936	1.918656	12
Greedy BFGS	h_unmet_goals	88	25	27	230	0.025341	15
	h_pg_level_sum	88	14	16	126	5.197635	14
	h_pg_maxlevel	88	21	23	195	3.969924	13
	h_pg_setlevel	88	35	37	345	21.997462	17
A* Search	h_unmet_goals	88	7388	7390	65711	1.629617	12
	h_pg_level_sum	88	369	371	3403	56.431944	12
	h_pg_maxlevel	88	9580	9582	86312	994.611309	12
	h_pg_setlevel	88	3423	3425	31596	1447.699186	12

Problem 4 Results

		# Actions	New Node Expansions	Goal Tests	New Nodes	Runtime	Plan Length
	BFS	104	99736	114953	944130	5.727469	14
	DFGS						
	UCS	104	113339	113341	1066413	9.077405	14
Greedy BFGS	h_unmet_goals	104	29	31	280	0.047138	18
	h_pg_level_sum	104	17	19	165	7.305587	17
	h_pg_maxlevel	104	56	58	580	13.589116	17
	h_pg_setlevel	104	107	109	1164	95.986711	23
A* Search	h_unmet_goals	104	34330	34332	328509	5.489794	14
	h_pg_level_sum	104	1208	1210	12210	314.69059	14
	h_pg_maxlevel	104	62077	62079	599376	10084.45204	14
	h_pg_setlevel	104	22606	22608	224229	15154.3854	14