PROJECT SPECIFICATION

Build a Forward Planning Agent

Planning Graph Implementation

<table>
<tr><th>CRITERIA</th><th>MEETS SPECIFICATIONS</th><th>STUDENT COMMENTS</th></tr>
<tr>
<td>Mutexes pass all test cases</td>
<td>(AUTOGRADED) Student code passes all Project Assistant test cases for:
ActionLayer mutual exclusion rules:
_inconsistent_effects()
_interference()
_competing_needs()
LiteralLayer mutual exclusion rules:
_inconsistent_support()
_negation()</td>
<td>ActionLayer mutual exclusion rules implemented in my_planning_graph.py:
_inconsistent_effects() → lines 13-29
_interference()→ lines 33-49
_competing_needs() → 53-69
LiteralLayer mutual exclusion rules implemented in my_planning_graph.py :
_inconsistent_support() → lines 75-91
_negation() → lines 93-97</td>
</tr>
</table>

Heuristic Implementation

CRITERIA	MEETS SPECIFICATIONS	STUDENT COMMENTS

Planning graph heuristics pass all test cases	(AUTOGRADED) Student code passes all Project Assistant test cases for: Correctly implemented
	
	• `PlanningGraph` class heuristics:
	  • `h_levelsum()`
	  • `h_maxlevel()`
	  • `h_setlevel()`
		• `PlanningGraph` class heuristics implemented in my_planning_graph.py:
		  • `h_levelsum() → lines 144-180`
		  • `h_maxlevel() → lines 183-221`
		  • `h_setlevel() → lines 225-277`

Experimental Results

	Actions				Expansions				Goal Tests				New Nodes				Time (seconds)				Plan Length			
Methods	P.1 A	P.2 A	P.3 A	P.4 A	P.1 E	P.2 E	P.3 E	P.4 E	P.1 G	P.2 G	P.3 G	P.4 G	P.1 NN	P.2 NN	P.3 NN	P.4 NN	P.1 T	P.2 T	P.3 T	P.4 T	P.1 PL	P.2 PL	P.3 PL	P.4 PL
breadth_first_search	20	72	88	104	43	3343	14663	99736	56	4609	18098	114953	178	30503	129625	944130	0.0275	0.3396	0.7922	5.6901	6	9	12	14
depth_first_graph_search	20	72	88	104	21	624	408	25174	22	625	409	25175	84	5602	3364	228849	0.0062	0.5251	0.1822	712.3256	20	619	392	24132
uniform_cost_search	20	72	88	104	60	5154	18510	113339	62	5156	18512	113341	240	46618	161936	1066413	0.0166	0.6573	1.1968	8.7772	6	9	12	14
greedy_best_first_graph_search With h_unmet_goals	20	72	88	104	7	17	25	29	9	19	27	31	29	170	230	280	0.0018	0.0243	0.0265	0.02798	6	9	15	18
greedy_best_first_graph_search With h_pg_level_sum	20	72	88	104	6	9	14	17	8	11	16	19	28	86	126	165	0.6646	2.1538	4.6336	8.1692	6	9	14	17
greedy_best_first_graph_search With h_pg_max_level	20	72	88	104	6	27	21	56	8	29	23	58	24	249	195	580	0.1978	4.2927	5.4246	19.4873	6	9	13	17
greedy_best_first_graph_search With h_pg_set_level	20	72	88	104	45	1417	11102	56600	47	1419	11104	56602	190	13755	101690	564988	1.6688	509.0421	4271.132	44628.0437	8	16	18	22
astar_search with h_unmet_goals	20	72	88	104	50	2467	7388	34330	52	2469	7390	34332	206	22522	65711	328509	0.0162	0.5646	0.8209	5.2661	6	9	12	14
astar_search with h_pg_level_sum	20	72	88	104	28	357	369	1208	30	359	371	1210	122	3426	3403	12210	0.3307	50.1059	80.9358	528.5884	6	9	12	15
astar_search with h_pg_max_level	20	72	88	104	43	2887	9580	62077	45	2889	9582	62079	180	26594	86312	599376	0.3329	287.6382	1470.1103	16245.3985	6	9	12	14
astar_search with h_pg_set_level	20	72	88	104	55	4367	17367	105284	57	4369	17369	105286	226	39865	153169	998285	1.1568	1444.5514	7171.3226	84988.384	6	9	12	14

Measurements with scaling expansions and Search Time 100% normalization

	Expansions				Time (seconds)				Plan Length			
Methods	P.1 E	P.2 E	P.3 E	P.4 E	P.1 T	P.2 T	P.3 T	P.4 T	P.1 PL	P.2 PL	P.3 PL	P.4 PL
breadth_first_search	0.04311382	3.351848881	14.70181279	100	0.483295548	5.968260663	13.92242667	100	6	9	12	14
depth_first_graph_search	0.083419401	2.478747915	1.62071979	100	0.000870388	0.073716289	0.025578191	100	20	619	392	24132
uniform_cost_search	0.05293853	4.547419688	16.33153636	100	0.189126373	7.488720777	13.6353279	100	6	9	12	14
greedy_best_first_graph_search With h_unmet_goals	24.13793103	58.62068966	86.20689655	100	6.433166548	86.84774839	94.71050751	100	6	9	15	18
greedy_best_first_graph_search With h_pg_level_sum	35.29411765	52.94117647	82.35294118	100	8.135435538	26.36488273	56.7203643	100	6	9	14	17
greedy_best_first_graph_search With h_pg_max_level	10.71428571	48.21428571	37.5	100	1.015020039	22.02819272	27.83659101	100	6	9	13	17
greedy_best_first_graph_search With h_pg_set_level	0.0795053	2.503533569	19.61484099	100	0.003739353	1.14063279	9.570511378	100	8	16	18	22
astar_search with h_unmet_goals	0.145645208	7.186134576	21.52053597	100	0.307628036	10.72140673	15.58838609	100	6	9	12	14
astar_search with h_pg_level_sum	2.317880795	29.55298013	30.54635762	100	0.062562856	9.479190236	15.31168675	100	6	9	12	15
astar_search with h_pg_max_level	0.069268811	4.650675774	15.4324468	100	0.002049196	1.770582605	9.049395126	100	6	9	12	14
astar_search with h_pg_set_level	0.052239657	4.14782873	16.49538391	100	0.001361127	1.699704515	8.438003245	100	6	9	12	14

CRITERIA	MEETS SPECIFICATIONS	

Analyze the search complexity as a function of domain size, search algorithm, and heuristic.

Report includes a table or chart to analyze the number of nodes expanded against number of actions in the domain.

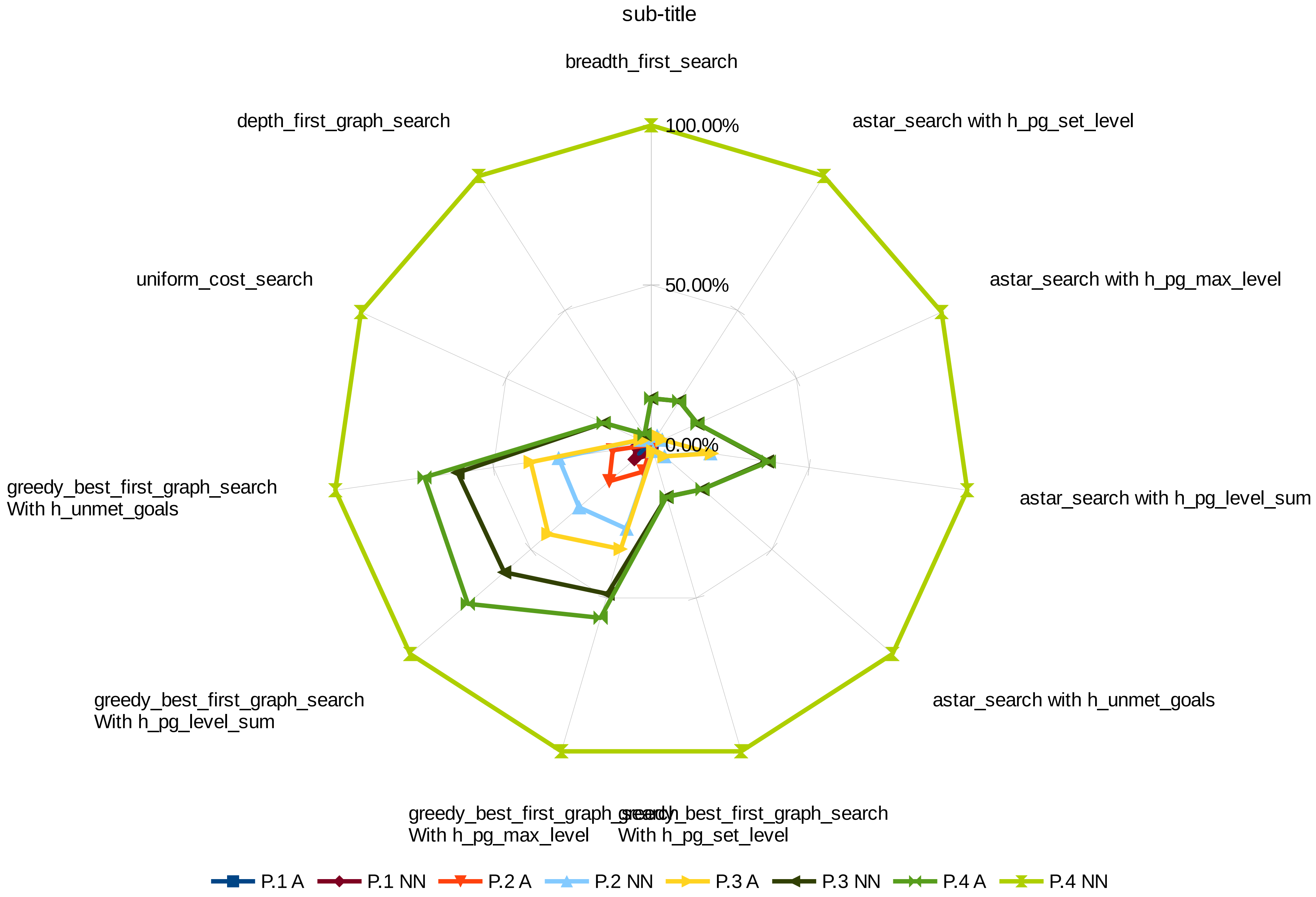
- The chart or table includes data for all search & heuristic combinations for air cargo problems 1 and 2
- The chart or table includes data **at least** one uninformed search, two heuristics with greedy best first search, and two heuristics with A* on air cargo problems 3 and 4
- Report includes at least a one paragraph discussion of these results that analyzes the growth trends as the problem size increases

The table in the previous section involves results for all problems and all algorithms in the table below and the graph. Generally we see a positive correlation between the number of nodes and the number of actions except in the case of depth_first_search and greedy_best_first_search with h_pg_max_level. The growth of the new nodes is also dependent on the search method along with the number of actions.

The graph in the next page shows the correlation better.

	Actions				New Nodes			
Methods	P.1 A	P.2 A	P.3 A	P.4 A	P.1 NN	P.2 NN	P.3 NN	P.4 NN
breadth_first_search	20	72	88	104	178	30503	129625	944130
depth_first_graph_search	20	72	88	104	84	5602	3364	228849
uniform_cost_search	20	72	88	104	240	46618	161936	1066413
greedy_best_first_graph_search With h_unmet_goals	20	72	88	104	29	170	230	280
greedy_best_first_graph_search With h_pg_level_sum	20	72	88	104	28	86	126	165
greedy_best_first_graph_search With h_pg_max_level	20	72	88	104	24	249	195	580
greedy_best_first_graph_search With h_pg_set_level	20	72	88	104	190	13755	101690	564988
astar_search with h_unmet_goals	20	72	88	104	206	22522	65711	328509
astar_search with h_pg_level_sum	20	72	88	104	122	3426	3403	12210
astar_search with h_pg_max_level	20	72	88	104	180	26594	86312	599376
astar_search with h_pg_set_level	20	72	88	104	226	39865	153169	998285

Actions Vs New Nodes:



CRITERIA	MEETS SPECIFICATIONS	

Analyze search time as a function of domain size, search algorithm, and heuristic.

Report includes a table or chart to analyze the search time against the number of actions in the domain.

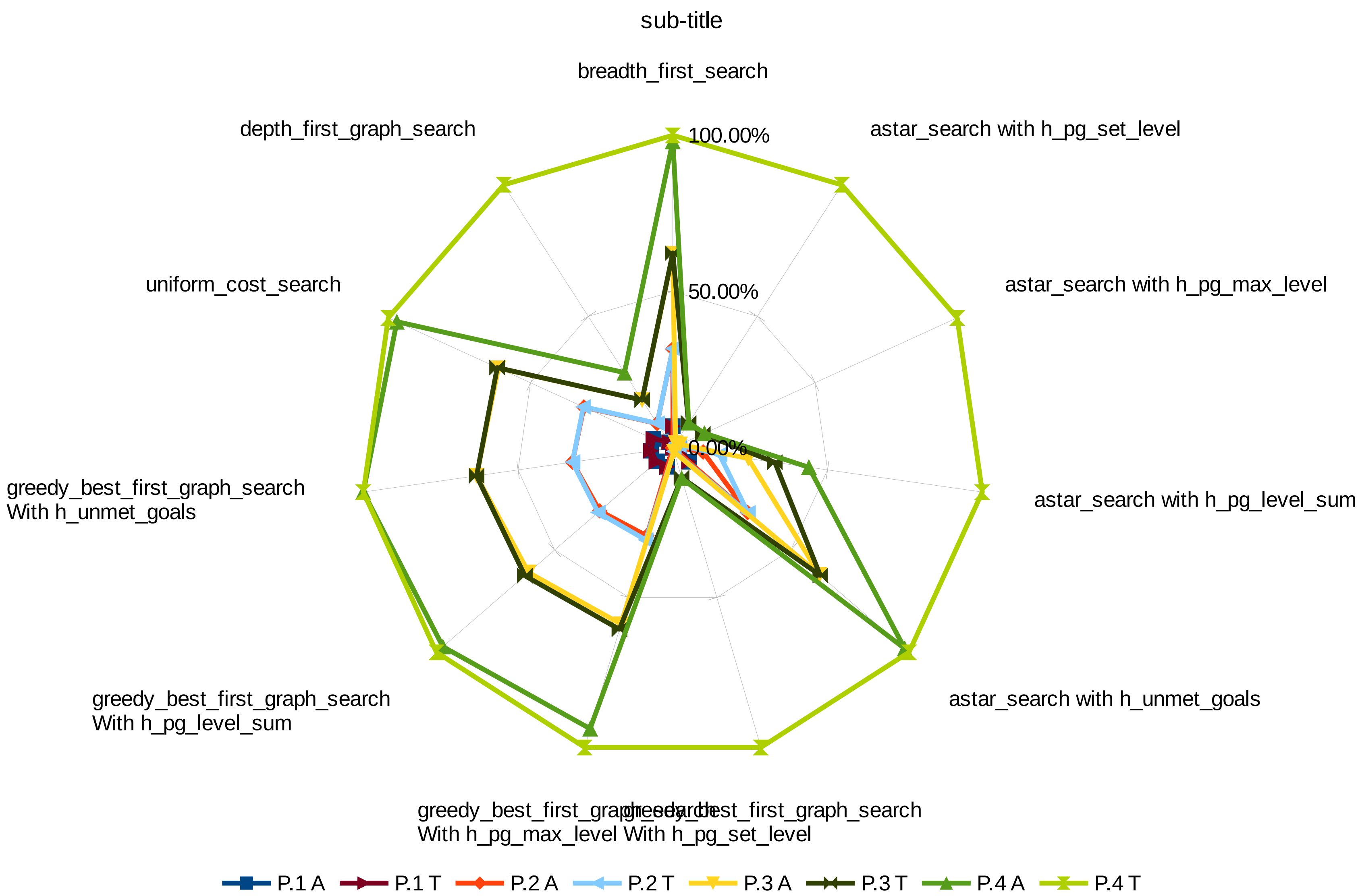
- The chart or table includes data for all search & heuristic combinations for air cargo problems 1 and 2
- The chart or table includes data **at least** one uninformed search, two heuristics with greedy best first search, and two heuristics with A* on air cargo problems 3 and 4
- Report includes at least a one paragraph discussion of these results that analyzes the growth trends as the problem size increases

The table in the previous section involves results for all problems and all algorithms in the table below. The number of actions and the search time for all search algorithms and all problems is reported below.

The results indicate that there is a positive correlation between the problem size, number of actions and the search method used. More than the number of actions, the search time is sensitive to the search algorithm and the type of problem used. This can be observed in search algorithms greedy_best_first_graph_search with h_pg_set_level, astar_search with h_pg_level_sum, astar_search with h_pg_max_level, astar_search with h_pg_set_level

	Actions				Time (seconds)			
Methods	P.1 A	P.2 A	P.3 A	P.4 A	P.1 T	P.2 T	P.3 T	P.4 T
breadth_first_search	20	72	88	104	0.0275	0.3396	0.7922	5.6901
depth_first_graph_search	20	72	88	104	0.0062	0.5251	0.1822	712.3256
uniform_cost_search	20	72	88	104	0.0166	0.6573	1.1968	8.7772
greedy_best_first_graph_search With h_unmet_goals	20	72	88	104	0.0018	0.0243	0.0265	0.02798
greedy_best_first_graph_search With h_pg_level_sum	20	72	88	104	0.6646	2.1538	4.6336	8.1692
greedy_best_first_graph_search With h_pg_max_level	20	72	88	104	0.1978	4.2927	5.4246	19.4873
greedy_best_first_graph_search With h_pg_set_level	20	72	88	104	1.6688	509.0421	4271.132	44628.0437
astar_search with h_unmet_goals	20	72	88	104	0.0162	0.5646	0.8209	5.2661
astar_search with h_pg_level_sum	20	72	88	104	0.3307	50.1059	80.9358	528.5884
astar_search with h_pg_max_level	20	72	88	104	0.3329	287.6382	1470.1103	16245.3985
astar_search with h_pg_set_level	20	72	88	104	1.1568	1444.5514	7171.3226	84988.384

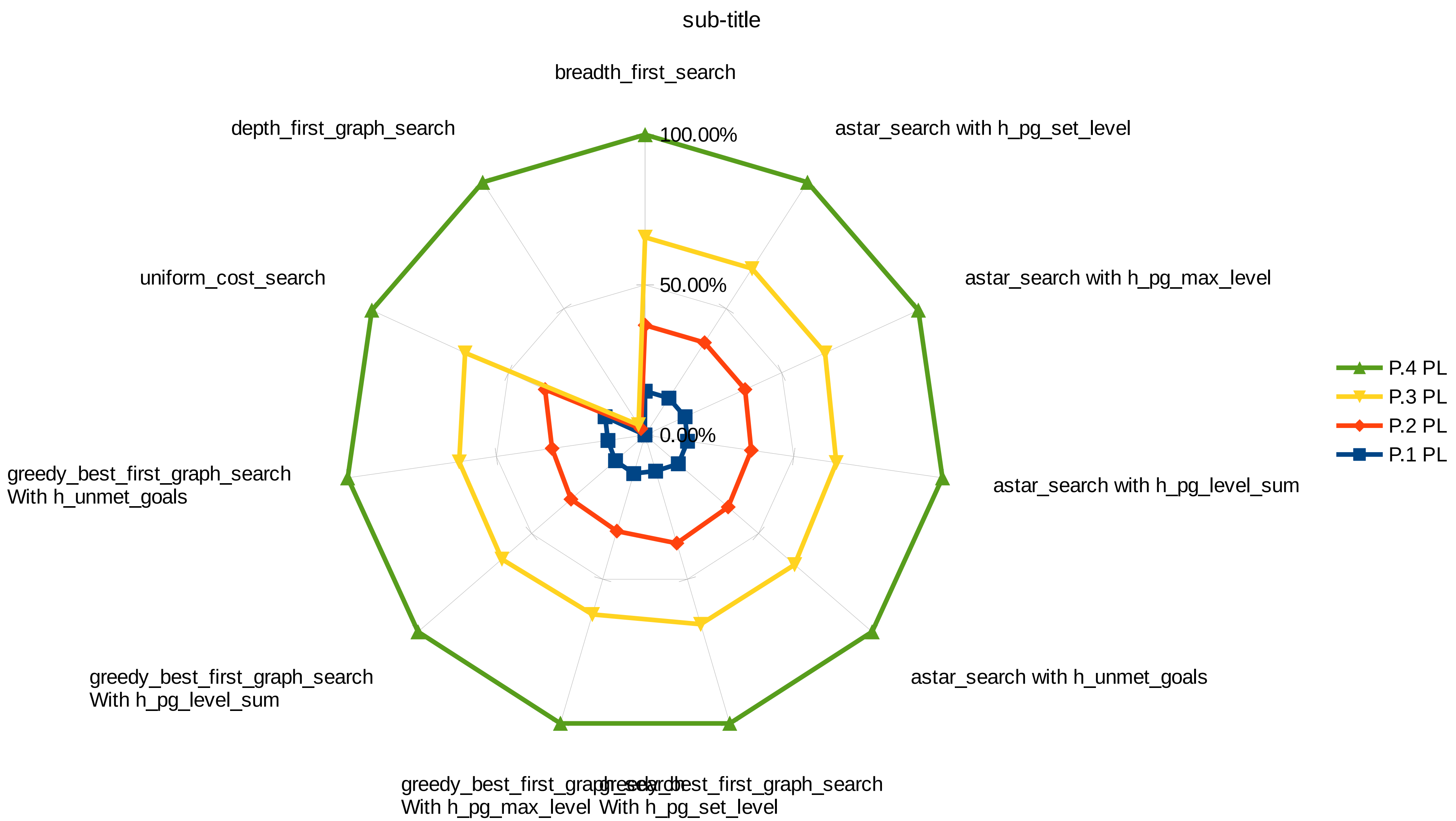
Actions Vs Search Time



CRITERIA	MEETS SPECIFICATIONS	
Analyze the optimality of solution as a function of domain size, search algorithm, and heuristic.	Report includes a table or chart to analyze the length of the plans returned by each algorithm on all search problems. • The chart or table includes data for all search & heuristic combinations for air cargo problems 1 and 2 • The chart or table includes data **at least** one uninformed search, two heuristics with greedy best first search, and two heuristics with A* on air cargo problems 3 and 4	The table in the previous section involves results for all problems and all algorithms in the table below. All the algorithms are generating a proportinal plan lengths except for the depth_first_graph_search. depth_first_graph_search is growing exponentially. The graph in the next page shows depth_first_graph_search as outlier very clearly.

	Plan Length			
Methods	P.1 PL	P.2 PL	P.3 PL	P.4 PL
breadth_first_search	6	9	12	14
depth_first_graph_search	20	619	392	24132
uniform_cost_search	6	9	12	14
greedy_best_first_graph_search With h_unmet_goals	6	9	15	18
greedy_best_first_graph_search With h_pg_level_sum	6	9	14	17
greedy_best_first_graph_search With h_pg_max_level	6	9	13	17
greedy_best_first_graph_search With h_pg_set_level	8	16	18	22
astar_search with h_unmet_goals	6	9	12	14
astar_search with h_pg_level_sum	6	9	12	15
astar_search with h_pg_max_level	6	9	12	14
astar_search with h_pg_set_level	6	9	12	14

Plan Lengths for each algorithm:



CRITERIA	MEETS SPECIFICATIONS	
Report answers all required questions	Submission includes a short answer to each of the following questions. (A short answer should be at least 1-2 sentences at most a small paragraph.) •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? •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_graph_searchwithh_unmet_goals would be the most appropriate for planning in a very restricted domain and also operate in real time mainly due the the minimal search times. • greedy_best_first_graph_searchwithh_unmet_goals is the one that fits better.

CRITERIA	MEETS SPECIFICATIONS	
	•Which algorithm or algorithms would be most appropriate for planning problems where it is important to find only optimal plans?	• astar_searchwithh_unmet_goals, breadth_first_search, uniform_cost_search suit better to do the planning for the optimal as they exhaust all the possibilities