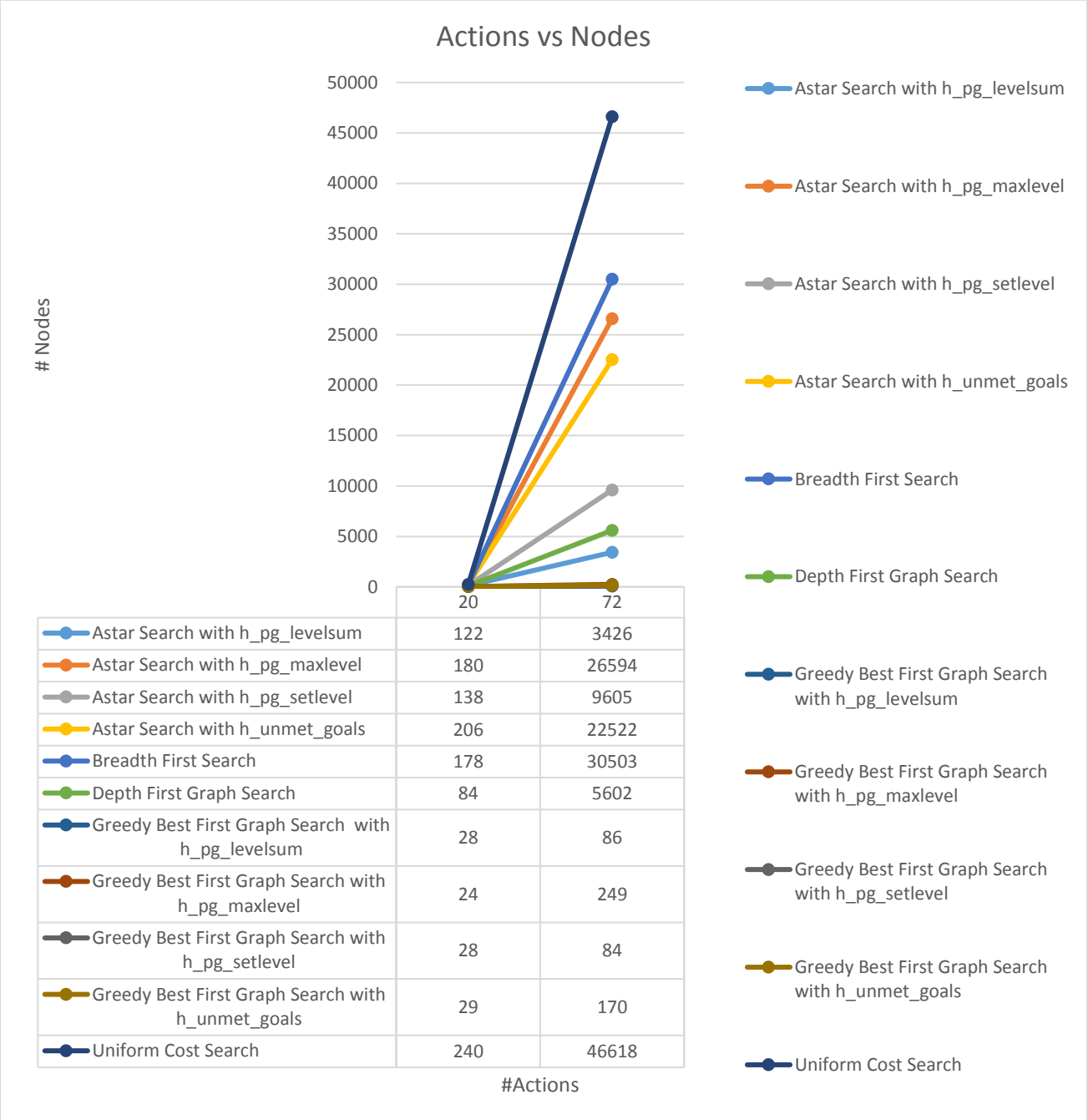
Classical Planning Report

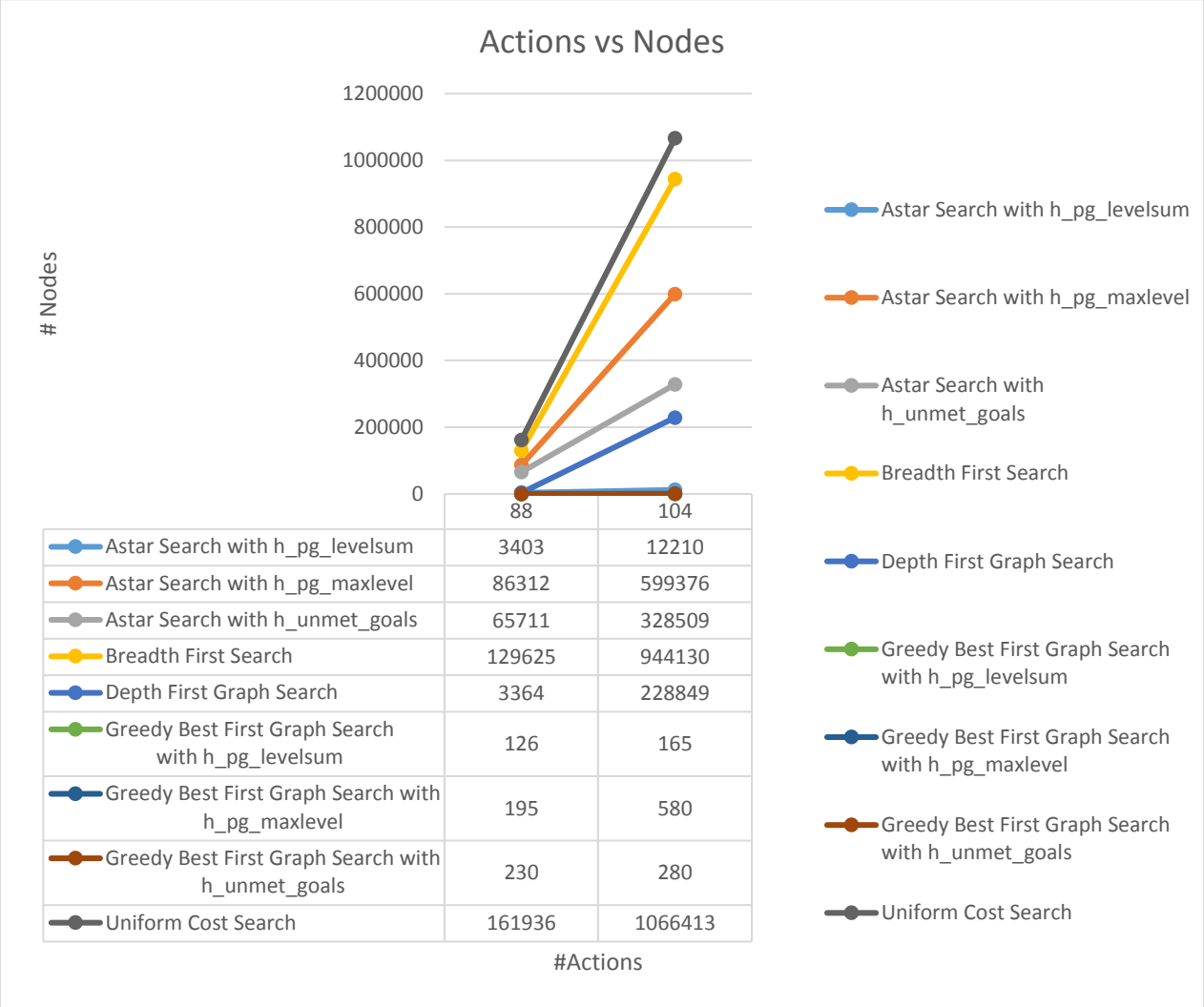
1. Search complexity as a function of domain size, search algorithm, and heuristic.

a. Search & heuristic combinations for air cargo problems 1 and 2:



	20	72
Astar Search with h_pg_levelsum	122	3426
Astar Search with h_pg_maxlevel	180	26594
Astar Search with h_pg_setlevel	138	9605
Astar Search with h_unmet_goals	206	22522
Breadth First Search	178	30503
Depth First Graph Search	84	5602
Greedy Best First Graph Search with h_pg_levelsum	28	86
Greedy Best First Graph Search with h_pg_maxlevel	24	249
Greedy Best First Graph Search with h_pg_setlevel	28	84
Greedy Best First Graph Search with h_unmet_goals	29	170
Uniform Cost Search	240	46618

b. At least one uninformed search, two heuristics with greedy best first search, and two heuristics with A* on air cargo problems 3 and 4:

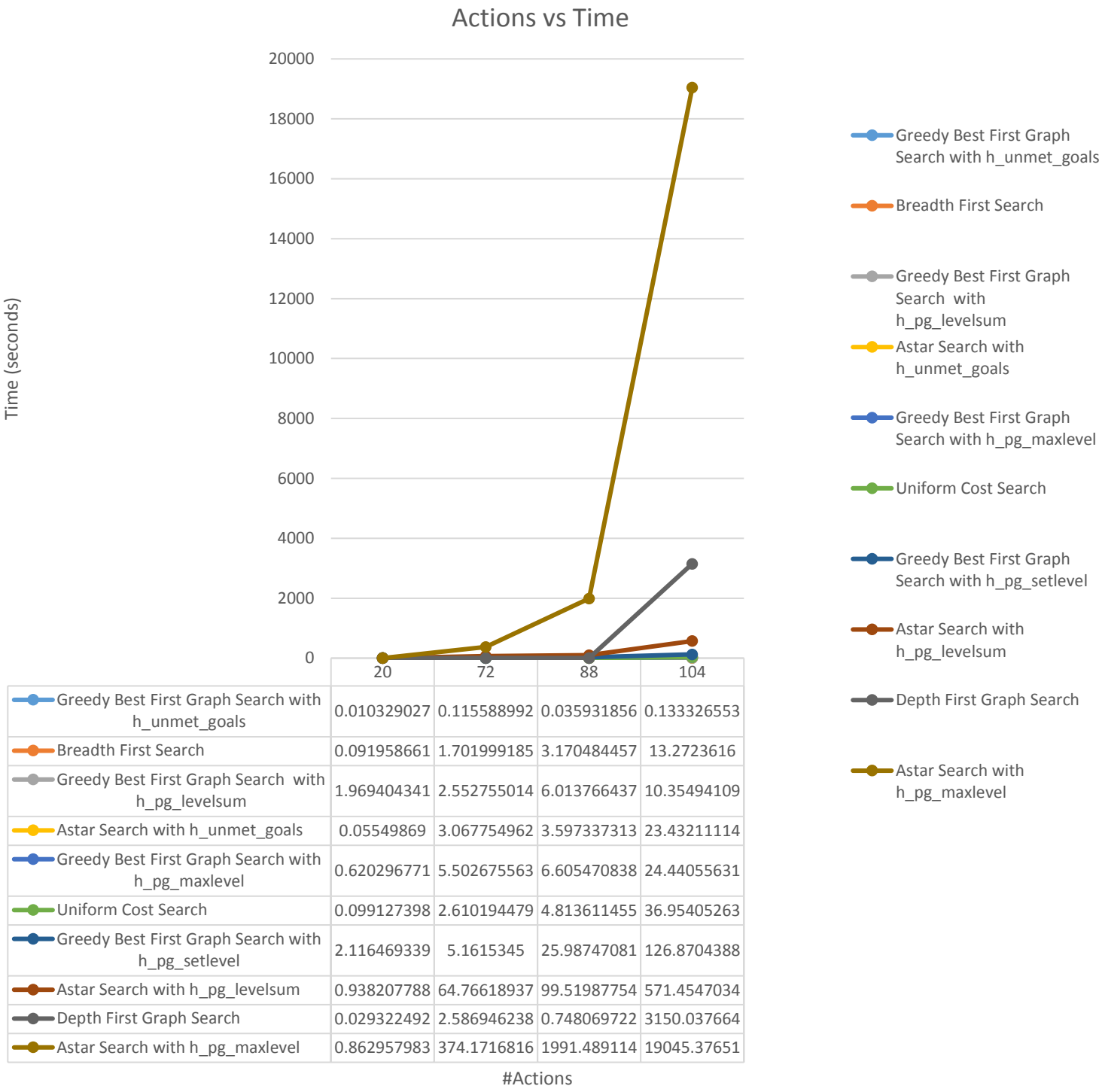


	88	104
Astar Search with h_pg_levelsum	3403	12210
Astar Search with h_pg_maxlevel	86312	599376
Astar Search with h_unmet_goals	65711	328509
Breadth First Search	129625	944130
Depth First Graph Search	3364	228849
Greedy Best First Graph Search with h_pg_levelsum	126	165
Greedy Best First Graph Search with h_pg_maxlevel	195	580
Greedy Best First Graph Search with h_unmet_goals	230	280
Uniform Cost Search	161936	1066413

According to the charts above, we can see that the number of nodes increases as the problem size increases for all algorithms.

Greedy Best First Graph Search with levelsum heuristic has the minimum number of new nodes. It performs better than all other algorithms in terms of number of new nodes.

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

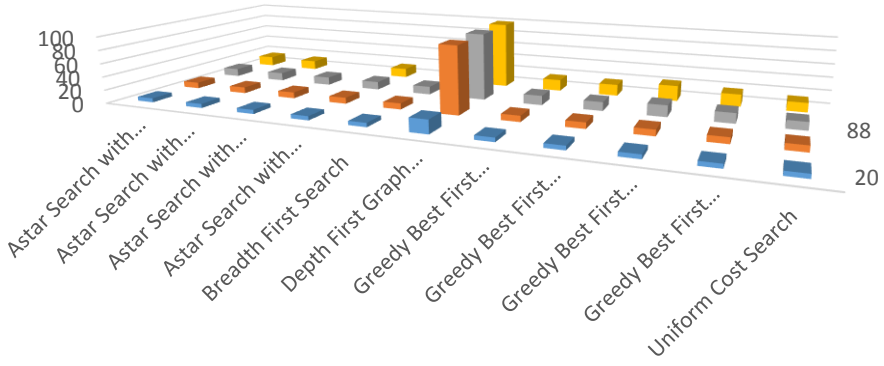


	20	72	88	104
Greedy Best First Graph Search with h_unmet_goals	0.010329027	0.115588992	0.035931856	0.133326553
Breadth First Search	0.091958661	1.701999185	3.170484457	13.2723616
Greedy Best First Graph Search with h_pg_levelsum	1.969404341	2.552755014	6.013766437	10.35494109
Astar Search with h_unmet_goals	0.05549869	3.067754962	3.597337313	23.43211114
Greedy Best First Graph Search with h_pg_maxlevel	0.620296771	5.502675563	6.605470838	24.44055631
Uniform Cost Search	0.099127398	2.610194479	4.813611455	36.95405263
Greedy Best First Graph Search with h_pg_setlevel	2.116469339	5.1615345	25.98747081	126.8704388
Astar Search with h_pg_levelsum	0.938207788	64.76618937	99.51987754	571.4547034
Depth First Graph Search	0.029322492	2.586946238	0.748069722	3150.037664
Astar Search with h_pg_maxlevel	0.862957983	374.1716816	1991.489114	19045.37651

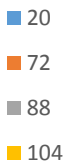
According to the chart above, we can see that the time taken increases as the number of actions increases for all algorithms.

Greedy Best First Graph Search with unmet goals heuristic takes least time. It performs better than all other algorithms in terms of time taken.

3. Optimality of solution as a function of domain size, search algorithm, and heuristic.



	Astar Search with h_pg_levelsum	Astar Search with h_pg_maxlevel	Astar Search with h_pg_setlevel	Astar Search with h_unmet_goals	Breadth First Search	Depth First Graph Search	Greedy Best First Graph Search with h_pg_levelsum	Greedy Best First Graph Search with h_pg_maxlevel	Greedy Best First Graph Search with h_pg_setlevel	Greedy Best First Graph Search with h_unmet_goals	Uniform Cost Search
20	6	6	6	6	6	20	6	6	6	6	6
72	9	9	9	9	9	619	9	9	9	9	9
88	12	12	12	12	12	392	14	13	17	15	12
104	15	14		14	14	24132	17	17	23	18	14



- In a restricted domain, where the number of actions is less, any algorithm is appropriate since all of them takes less time for planning. To optimize the number of the new nodes that are created, the appropriate algorithm will be greedy best first graph search (any heuristic is fine).
- In very large domains, where the number of actions are very high, the appropriate algorithm will be greedy best first graph search (preferable heuristics – levelsum or unmet goals)
- According to point 3 above (chart), Depth first search algorithm is the least optimal solution. Apart from Depth first search, all other algorithms are optimal.