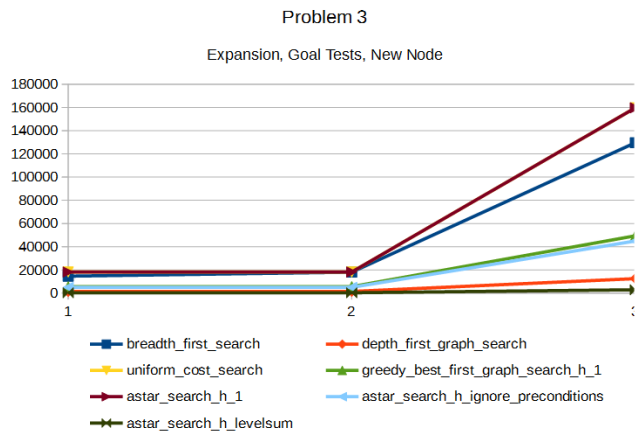
AI – Search project

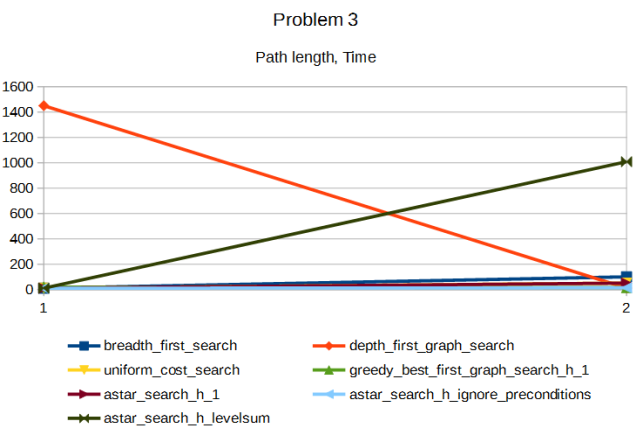
the optimal steps for Problem 1, 2, 3 are as below

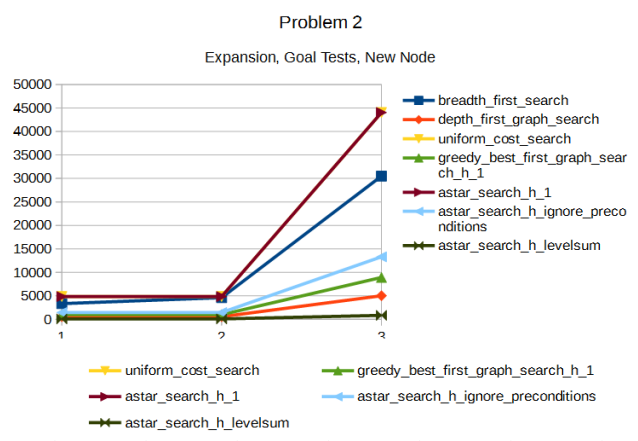
Problem 1, step = 6	Problem 2, step = 9	Problem 3, step = 12
Load(C1, P1, SFO) Fly(P1, SFO, JFK) Unload(C1, P1, JFK) Load(C2, P2, JFK) Fly(P2, JFK, SFO) Unload(C2, P2, SFO)	Load(C1, P1, SFO) Fly(P1, SFO, JFK) Load(C2, P2, JFK) Fly(P2, JFK, SFO) Load(C3, P3, ATL) Fly(P3, ATL, SFO) Unload(C3, P3, SFO) Unload(C2, P2, SFO) Unload(C1, P1, JFK)	Load(C2, P2, JFK) Fly(P2, JFK, ORD) Load(C4, P2, ORD) Fly(P2, ORD, SFO) Load(C1, P1, SFO) Fly(P1, SFO, ATL) Load(C3, P1, ATL) Fly(P1, ATL, JFK) Unload(C4, P2, SFO) Unload(C3, P1, JFK) Unload(C2, P2, SFO) Unload(C1, P1, JFK)

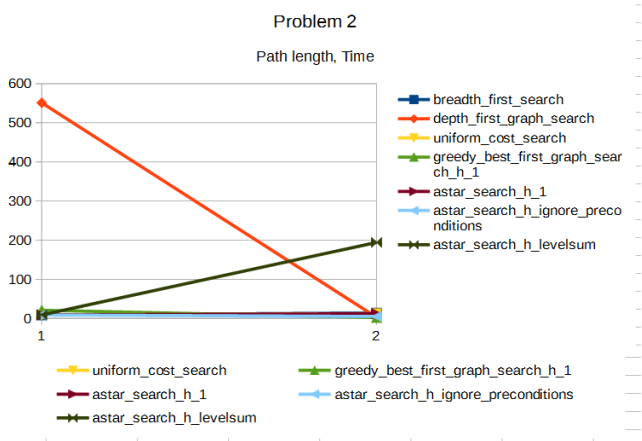
Detail running results

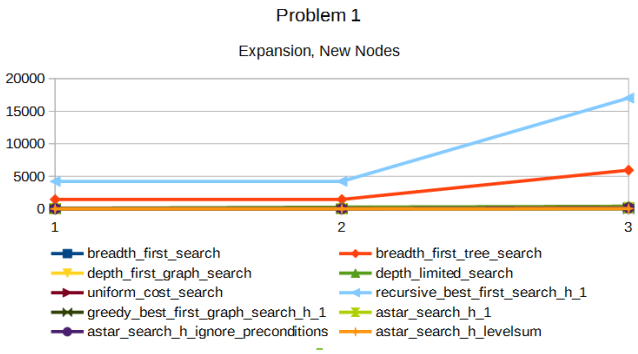
Problem	Algorithm	Expansion	Goal tests	New Node	path length	Time
3	breadth_first_search	14663	18098	129631	12	101.6
3	depth_first_graph_search	1501	1502	12519	1451	12.97
3	uniform_cost_search	18235	18237	159716	12	52.72
3	greedy_best_first_graph_search_h_1	5613	5615	49420	21	16.49
3	astar_search_h_1	18235	18237	159716	12	52.15
3	astar_search_h_ignore_preconditions	5040	5042	44944	12	14.89
3	astar_search_h_levelsum	316	318	2914	12	1009.35
2	breadth_first_search	3343	4609	30509	9	13.81
2	depth_first_graph_search	559	560	5008	551	2.92
2	uniform_cost_search	4852	4854	44030	9	12.04
2	greedy_best_first_graph_search_h_1	990	992	8910	21	2.41
2	astar_search_h_1	4852	4854	44030	9	12.11
2	astar_search_h_ignore_preconditions	1450	1452	13303	9	4.99
2	astar_search_h_levelsum	86	88	841	9	194.27
1	breadth_first_search	43	56	180	6	0.03
1	breadth_first_tree_search	1458	1459	5960	6	0.94
1	depth_first_graph_search	12	13	48	12	0.008
1	depth_limited_search	101	271	414	50	0.09
1	uniform_cost_search	55	57	224	6	0.037
1	recursive_best_first_search_h_1	4229	4230	17029	6	2.7
1	greedy_best_first_graph_search_h_1	7	9	28	6	0.005
1	astar_search_h_1	55	57	224	6	0.037
1	astar_search_h_ignore_preconditions	41	43	170	6	0.03
1	astar_search_h_levelsum	11	13	50	6	2.06

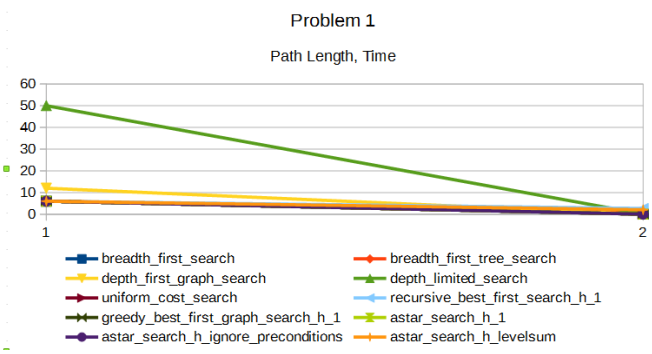












we can see A*-search with heuristics of level sum and A*-search_ignore_preconditions outperform other peers given that optimal strategy scheduled. level sum have the smallest expansion, new node and Goal tests, but its time cost is much higher than A*-search_ignore_preconditions due to the higher overhead of calculating the heuristic function, O(n^2) v.s. O(n). so after balancing all trade-off, A*-search_ignore_preconditions is the best.