Project 3: Planning

Markus Buchholz

1. Introduction

Following project report covers test results where several search strategies were used to solved Air Cargo transport problems. Three given project problems were defined and finally implemented into Python code. Investigation and results analysis performed during the project allowed choosing best heuristic for solving given problems.
In order to solve problem both uninformed and heuristic strategies were used. Uniformed strategies contradictory to heuristic do not have any additional information about the state, beyond that provided in the problem definition. Heuristic strategies have knowledge about the state and can distinguish the goal state from non-goal.

Due to long run time results for some strategies are missing (for problem 2: breadth first tree search and problem 3 breadth first tree search; depth limited search).

In order to give visual overview over the test results is was necessary to applied logarithmic scale.

2. Optimal plan for given problems

Optimal plans for problem 1, 2 and 3 were figured out applying A* with h1 heuristic:

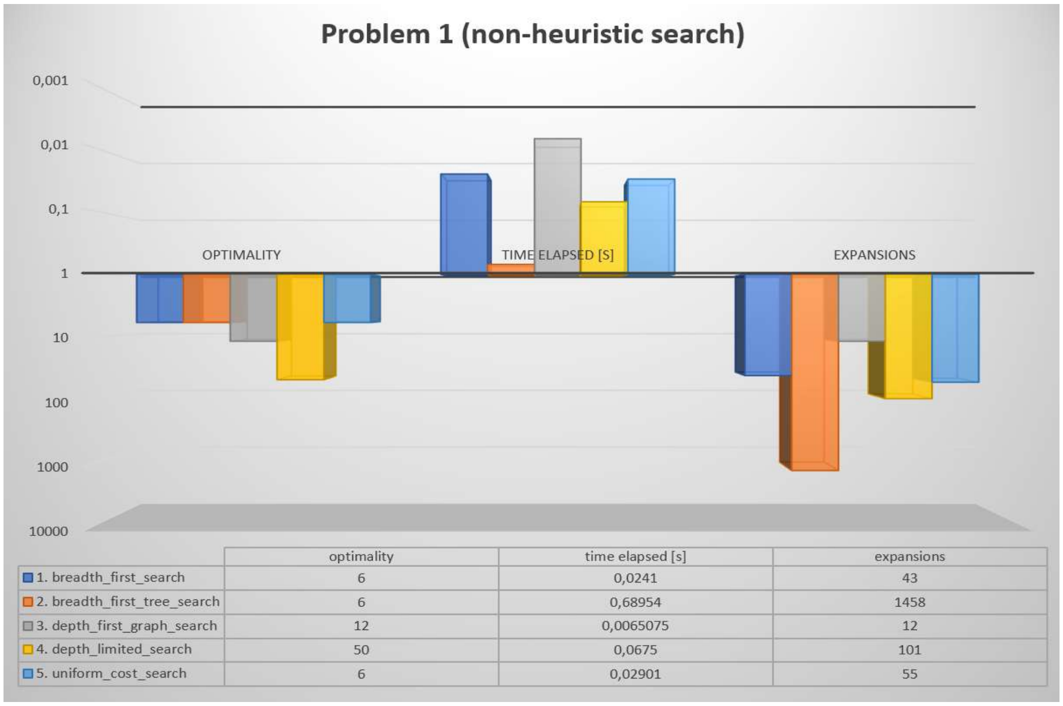
PROBLEM 1
Plan length: 6 Time elapsed in seconds: 0.02817713522693265
Load (C1, P1, SFO)
Load (C2, P2, JFK)
Fly (P1, SFO, JFK)
Fly (P2, JFK, SFO)
Unload (C1, P1, JFK)
Unload (C2, P2, SFO)

PROBLEM 2
Plan length: 9 Time elapsed in seconds: 8.55449256097429
Load (C1, P1, SFO)
Load (C2, P2, JFK)
Load (C3, P3, ATL)
Fly (P1, SFO, JFK)
Fly (P2, JFK, SFO)
Fly (P3, ATL, SFO)
Unload (C3, P3, SFO)
Unload (C2, P2, SFO)
Unload (C1, P1, JFK)

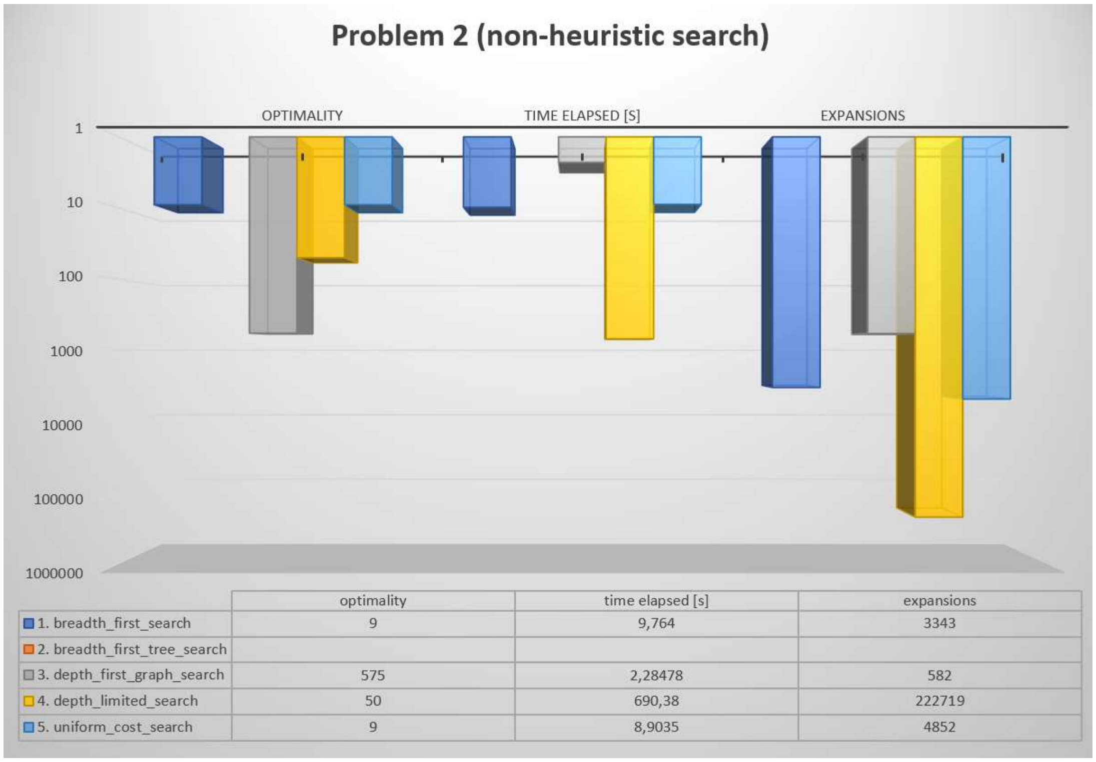
PROBLEM 3
Plan length: 12 ime elapsed in seconds: 41.66123828318897
Load (C1, P1, SFO)
Load (C2, P2, JFK)
Fly (P1, SFO, ATL)
Load (C3, P1, ATL)
Fly (P2, JFK, ORD)
Load (C4, P2, ORD)
Fly (P2, ORD, SFO)
Fly (P1, ATL, JFK)
Unload (C4, P2, SFO)
Unload (C3, P1, JFK)
Unload (C2, P2, SFO)
Unload (C1, P1, JFK)

3. Compare and contrast non-heuristic search

Below figures depict the comparison of result metrics (optimality, time elapsed, number of node expansions) for applied non-heuristic search strategies. As it was mentioned in introduction to following project, in problem 2 and problem 3 some result are missing (due to long run time). The breath first tree search strategy is most time consuming. For solving problem 1 used approximately 100 times longer time then the quickest strategy: depth first graph search. This strategy solves all three given problem much fasted then other. The other strategy which takes second place in "speed ranking" is uniform cost search. Beside the solving time the optimality metrics should be also considered in order to evaluate given search strategies. This time the fasted strategy is poor. The length to goal (cost) is much higher than the other. For moderate problem complexity (problem 2 and 3) it costs approximately 60 times higher than second speed ranked strategy: uniform cost search. Similar optimality (to uniform cost search) can be measured for breath first search but the computation time is 2 time higher. Other strategies: breath first tree search and deep limited search seems to be not proper strategies for solving given problems (takes too long time then the other).

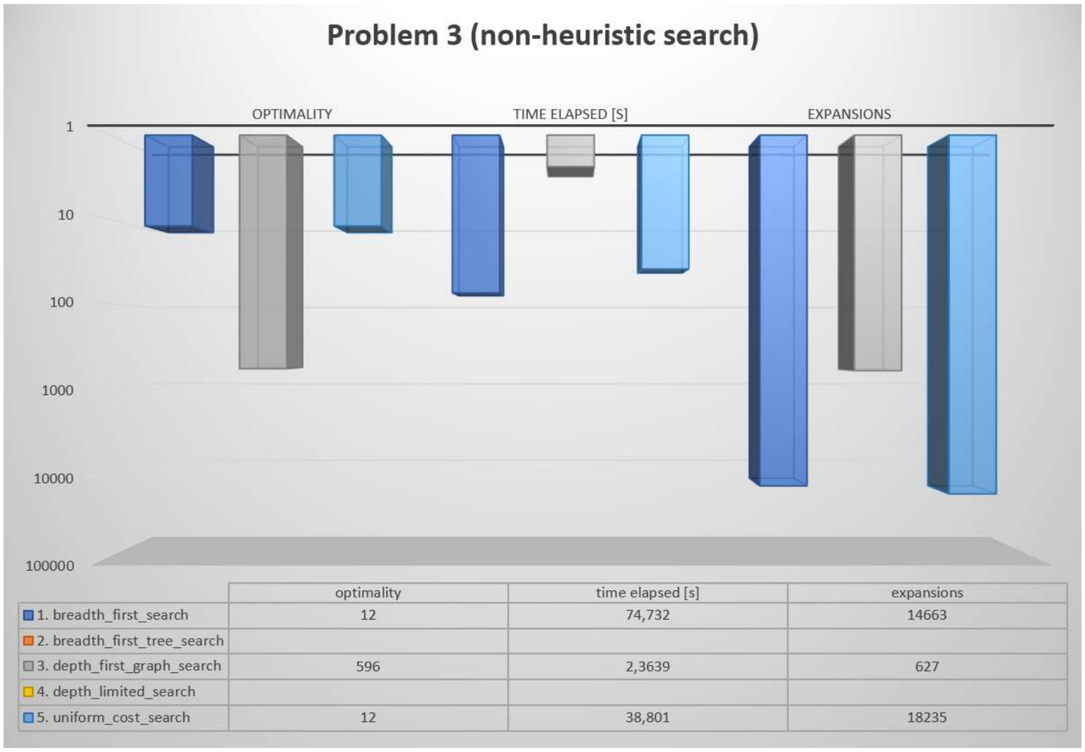


	optimality	time elapsed [s]	expansions
🟦 1. breadth_first_search	6	0,0241	43
🟧 2. breadth_first_tree_search	6	0,68954	1458
⬜ 3. depth_first_graph_search	12	0,0065075	12
🟨 4. depth_limited_search	50	0,0675	101
🟦 5. uniform_cost_search	6	0,02901	55



Problem 2 (non-heuristic search)

	optimality	time elapsed [s]	expansions
🟦 1. breadth_first_search	9	9,764	3343
🟧 2. breadth_first_tree_search			
⬜ 3. depth_first_graph_search	575	2,28478	582
🟨 4. depth_limited_search	50	690,38	222719
🟦 5. uniform_cost_search	9	8,9035	4852



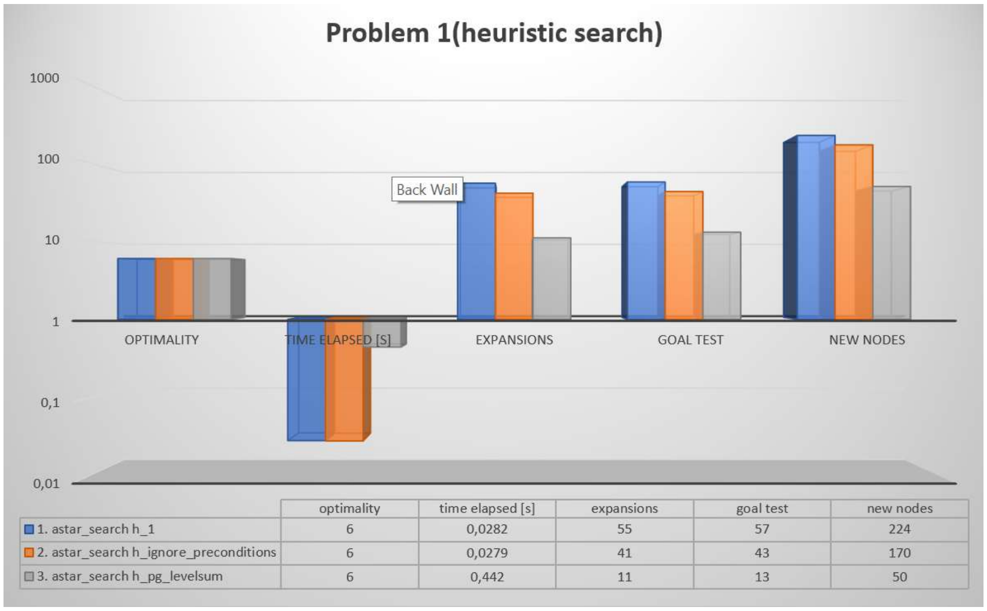
Problem 3 (non-heuristic search)

	optimality	time elapsed [s]	expansions
🟦 1. breadth_first_search	12	74,732	14663
🟧 2. breadth_first_tree_search			
⬜ 3. depth_first_graph_search	596	2,3639	627
🟨 4. depth_limited_search			
🟦 5. uniform_cost_search	12	38,801	18235

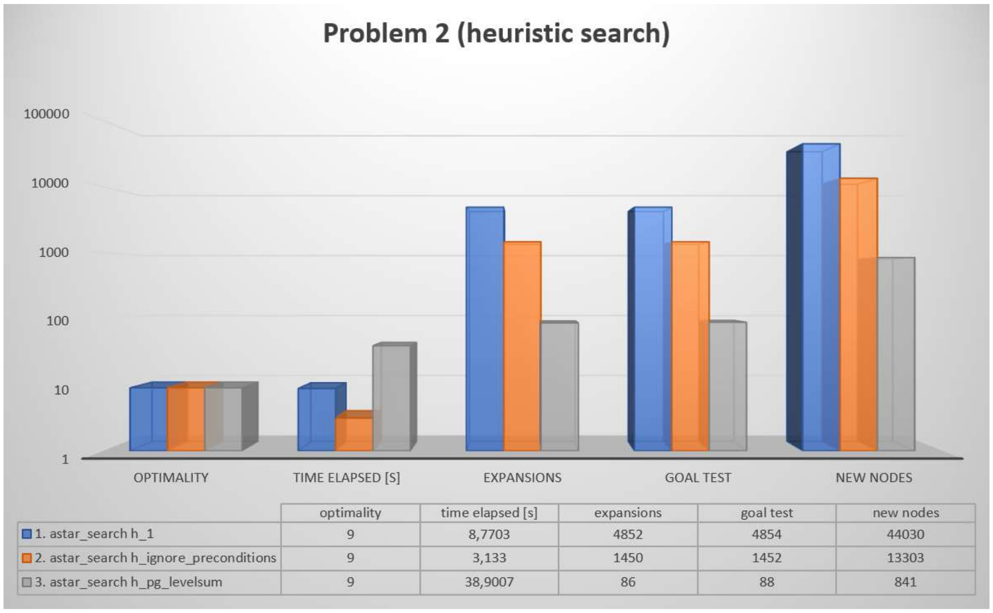
4. Compare and contrast heuristic search

Below figures and test results indicate that the fastest heurist strategy is A* with ignore preconditions which drops all precondition from the actions therefore every action becomes applicable in every state. For the most complex problem (problem 3) regarding strategy is less time consumable (time elapsed is 3 time lower then A* with h1 search and 6 times lower than A* with PG level sum). However, the slowest strategy: A* with PG level sum for all given problems superiors (over other applied in this research) considerable in other metrices. Expansions, goal test and new nodes are lowest over other applied strategies (for problem 3 regarding strategy has approximately 60 times lower metrices then A* with h1).

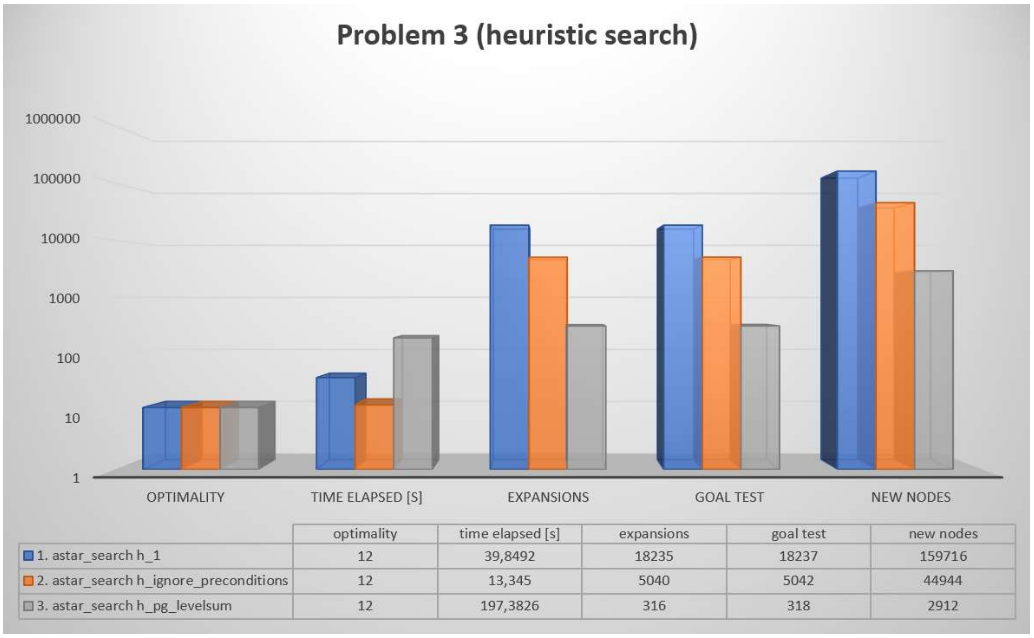
All tested strategies for all given problems achieved the same optimality.



	optimality	time elapsed [s]	expansions	goal test	new nodes
1. astar_search h_1	6	0,0282	55	57	224
2. astar_search h_ignore_preconditions	6	0,0279	41	43	170
3. astar_search h_pg_levelsum	6	0,442	11	13	50



	optimality	time elapsed [s]	expansions	goal test	new nodes
1. astar_search h_1	9	8,7703	4852	4854	44030
2. astar_search h_ignore_preconditions	9	3,133	1450	1452	13303
3. astar_search h_pg_levelsum	9	38,9007	86	88	841



	optimality	time elapsed [s]	expansions	goal test	new nodes
🟦 1. astar_search h_1	12	39,8492	18235	18237	159716
🟧 2. astar_search h_ignore_preconditions	12	13,345	5040	5042	44944
⬜ 3. astar_search h_pg_levelsum	12	197,3826	316	318	2912

5. **What was the best heuristic used in these problems? Was it better than non-heuristic search planning methods for all problems? Why or why not?**

Based on performed tests results, where the solutions for given problems had to be elaborated the it is possible to that for the non-heuristic strategies the **uniform cost search** is optimal strategy to solve problem. Recommended strategy secures the equilibrity between search time and total costs. Other verified strategies perform better in one metrics (speed or optimality) Recommended strategy balances both investigated metrices. Expansion metrics for the recommended strategy increases while the complexity of the problem increases : For problem 1 expansions equals 55 while for the problem 3 expansions 18235 (similar phenomena can be observed also for breadth search first strategy). It is clear justification of theory given in AIMA (paragraph 3.4.2).

For the heuristic search, it seems that **A* search with heuristic ignoring precondition** is the best choice. Strategy secures best results in applied metrices for all given problems. Optimality is the same for all verified strategies.

The best strategy verified in this project is heuristic strategy **A* search with heuristic ignoring precondition**. Beside the optimality is the same for both strategies but the heuristic strategy is faster (problem 2 and 3) than uniform cost search and has lower expansions (problem 1, 2 and 3). In AIMA (paragraph 3.5.2), the author points that the time of A* is a main drawback (problem 3 can be solved faster by depth first graph search). However, algorithm overcomes the space problem without sacrificing optimality or completeness. It was justified in performed research where lower expansions for this recommended strategy were achieved.