This report assists in understanding the progression search in term of speed, optimality, and complexity of as the size of a problem increases.

Search Algorithms can be selected as follows:

- Uninformed:
  - UCS: uniform cost search
  - BFS: breadth first search
  - DFGS: depth first graph search

: Excellent

- With Heuristics:
  - GBFGS: greedy best first graph search with:
    - h_unmet_goals
    - h_pg_levelsum
    - h_pg_maxlevel
    - h_pg_setlevel
  - A*: aster search with:
    - h_unmet_goals
    - h_pg_levelsum
    - h_pg_maxlevel
    - h_pg_setlevel

Air Cargo Problems are as follows:

Problem	# Airplanes	# Cargo Items	# Airports
1	2	2	2
2	3	3	3
3	2	4	4
4	2	5	4

First, Uninformed algorithms:

Problem	Heuristic	Actions	Expansions	Goal Tests	New Nodes	Plan Length	Seconds
1	breadth_first_search	20	43	56	178	6	0,02
1	depth_first_graph_search	20	21	22	84	20	0,00
1	uniform_cost_search	20	60	62	240	6	0,01
2	breadth_first_search	72	3343	4609	30503	9	1,68
2	depth_first_graph_search	72	624	625	5602	619	2,34
2	uniform_cost_search	72	5154	5156	46618	9	2,65
3	breadth_first_search	88	14663	18098	129625	12	8,44
3	uniform_cost_search	88	18510	18512	161936	12	13,31
4	breadth_first_search	104	99736	114953	944130	14	82,12
4	uniform_cost_search	104	113339	113341	1066413	14	92,26

: Awesome

Comments:

- Problem 1:
  - optimal plan length => BFS & UCS (plan length= 6)
  - fastest => DFGS (regardless of a higher Plan Length, the lowest number of node expansions).
  - DFGS => shows no optimal value nor guaranteed solution, thus, it should be excluded from Problems 3 and 4

: This is the statement which got me here. Well done.

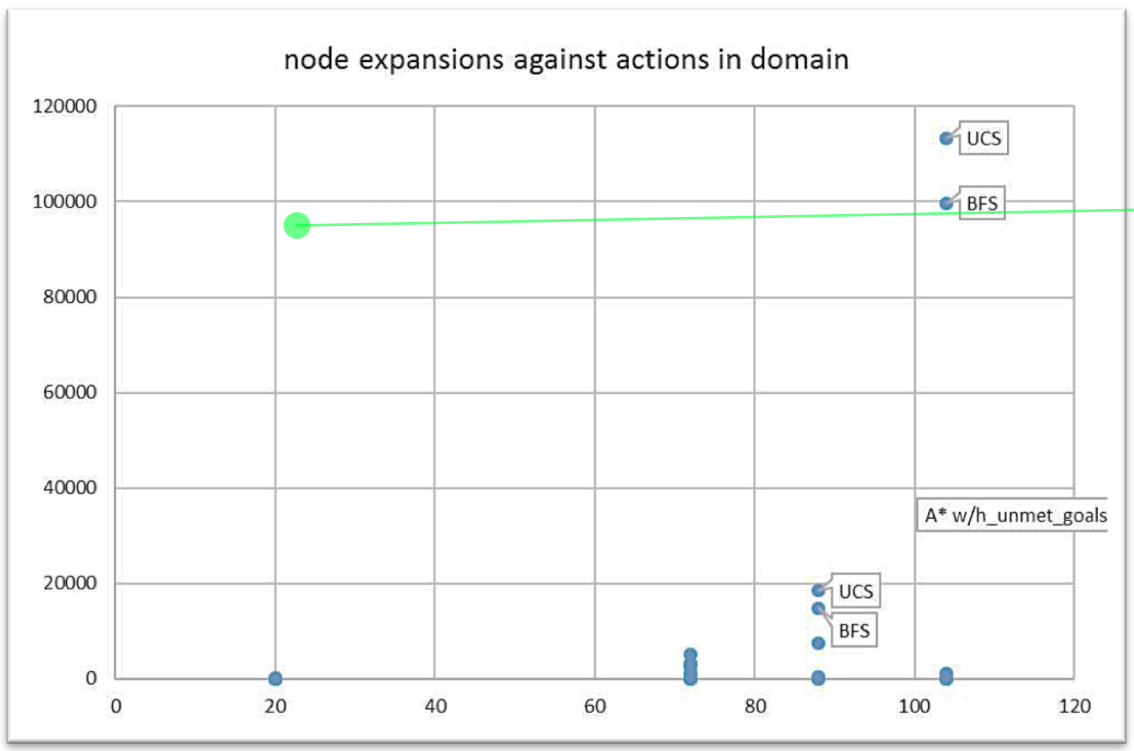
- Problem 2, 3 and 4:
    - optimal plan length => BFS and UCS.
    - fastest => BFS (time may escalate quickly with significant depth)

Next,* with Heuristics*:

Problem	Heuristic	Actions	Expansions	Goal Tests	New Nodes	Plan Length	Seconds
1	greedy_best_first_graph_search with h_unmet_goals	20	7	9	29	6	0,00
1	greedy_best_first_graph_search with h_pg_levelsum	20	6	8	28	6	0,29
1	greedy_best_first_graph_search with h_pg_maxlevel	20	6	8	24	6	0,22
1	greedy_best_first_graph_search with h_pg_setlevel	20	6	8	28	6	1,12
1	astar_search with h_unmet_goals	20	50	52	206	6	0,01
1	astar_search with h_pg_levelsum	20	28	30	122	6	0,73
1	astar_search with h_pg_maxlevel	20	43	45	180	6	0,77
1	astar_search with h_pg_setlevel	20	33	35	138	6	2,86
2	greedy_best_first_graph_search with h_unmet_goals	72	17	19	170	9	0,02
2	greedy_best_first_graph_search with h_pg_levelsum	72	9	11	86	9	6,40
2	greedy_best_first_graph_search with h_pg_maxlevel	72	27	29	249	9	13,09
2	greedy_best_first_graph_search with h_pg_setlevel	72	9	11	84	9	31,14
2	astar_search with h_unmet_goals	72	2467	2469	22522	9	1,82
2	astar_search with h_pg_levelsum	72	357	359	3426	9	201,11
2	astar_search with h_pg_maxlevel	72	2887	2889	26594	9	957,38
2	astar_search with h_pg_setlevel	72	1037	1039	9605	9	2.392,57
3	greedy_best_first_graph_search with h_unmet_goals	88	25	27	230	15	0,03
3	greedy_best_first_graph_search with h_pg_levelsum	88	14	16	126	14	14,21
3	astar_search with h_unmet_goals	88	7388	7390	65711	12	6,70
3	astar_search with h_pg_levelsum	88	369	371	3403	12	263,73
4	greedy_best_first_graph_search with h_unmet_goals	104	29	31	280	18	0,07
4	greedy_best_first_graph_search with h_pg_levelsum	104	17	19	165	17	25,79
4	astar_search with h_unmet_goals	104	34330	34332	328509	14	45,96
4	astar_search with h_pg_levelsum	104	1208	1210	12210	15	3.128,92

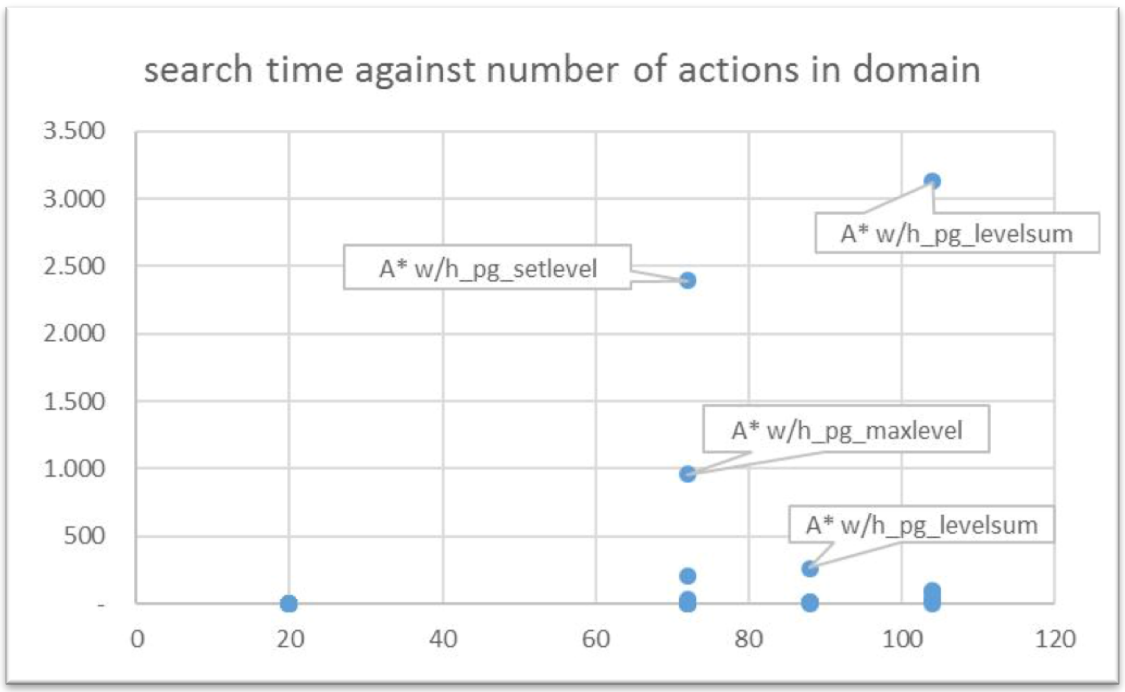
- GBFGS => lowest node expansion.
- A* => optimal in term of lower node expansions and timing.

The number of nodes expanded against actions in domain:

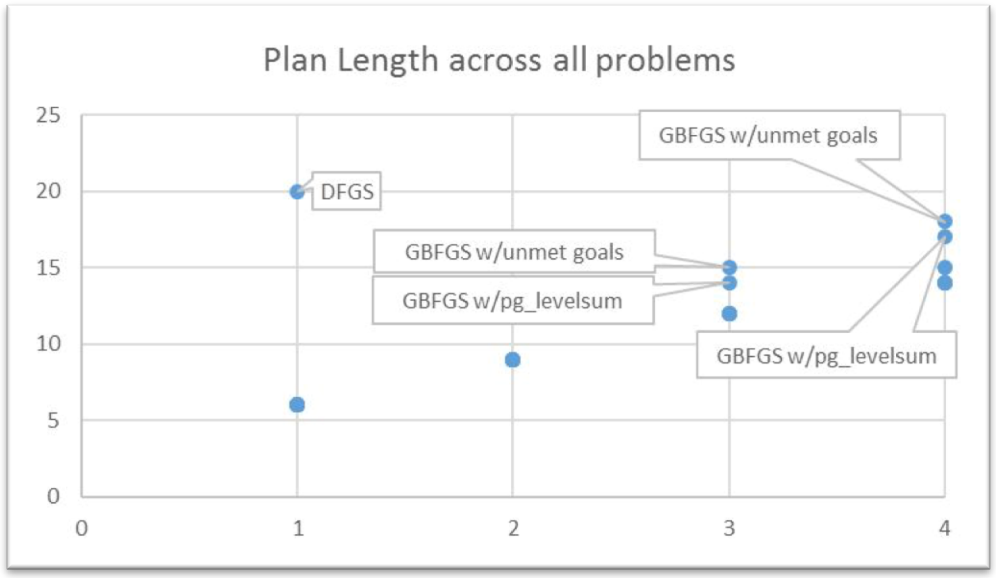


: Nice chart.

Accordingly, uninformed search algorithms are not recommended when the number of actions increases.



Accordingly, selected A* algorithms are impacted greatly in term of time upon the increase of the number of actions.



Accordingly, GBFGS algorithms are impacted greatly in term of Plan Length upon the increase of the complexity of the problem.

- **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?**
  According to the results above, the most appropriate are BFS & UCS.

: Accurate

- **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)**
  According to the results above, the most appropriate are GBFGS with unmet goals.

: Thumbs up

- **Which algorithm or algorithms would be most appropriate for planning problems where it is important to find only optimal plans?**
  According to the results above, the most appropriate are BFS & UCS