Build a Forward Planning Agent Report

This report consists of four main sections. The first three sections discuss the relation between number of expanded nodes, searching time, and plan's length and the number of actions. The last section discusses three proposed questions about the planning.

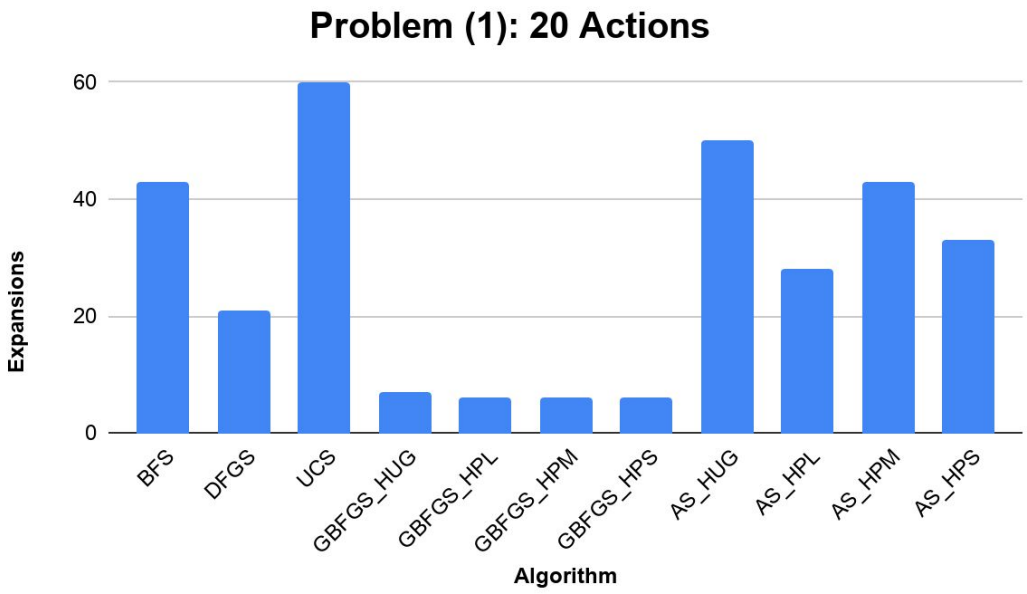
1) Number of nodes expanded against number of actions

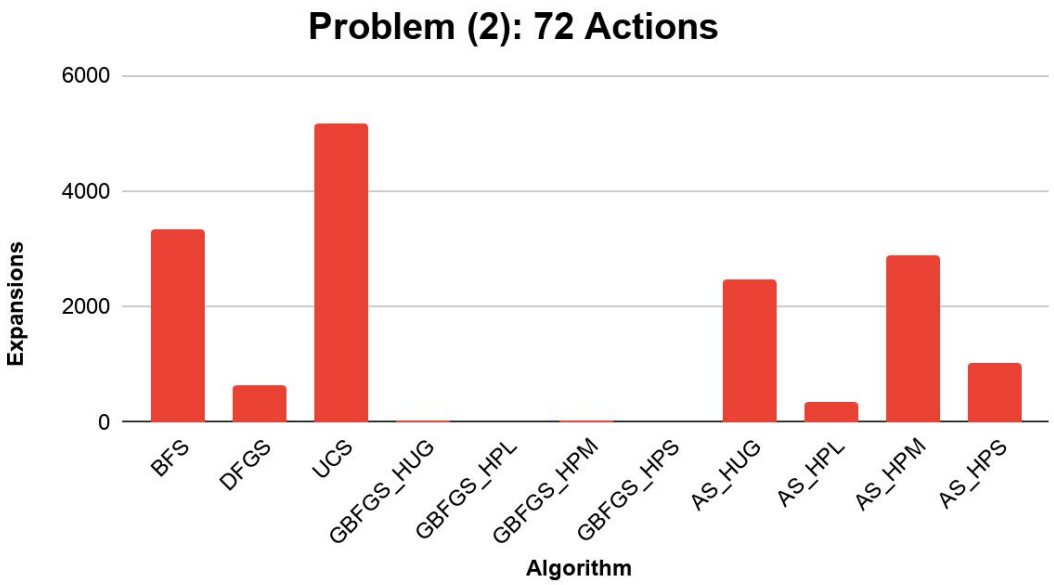
   a) Tables

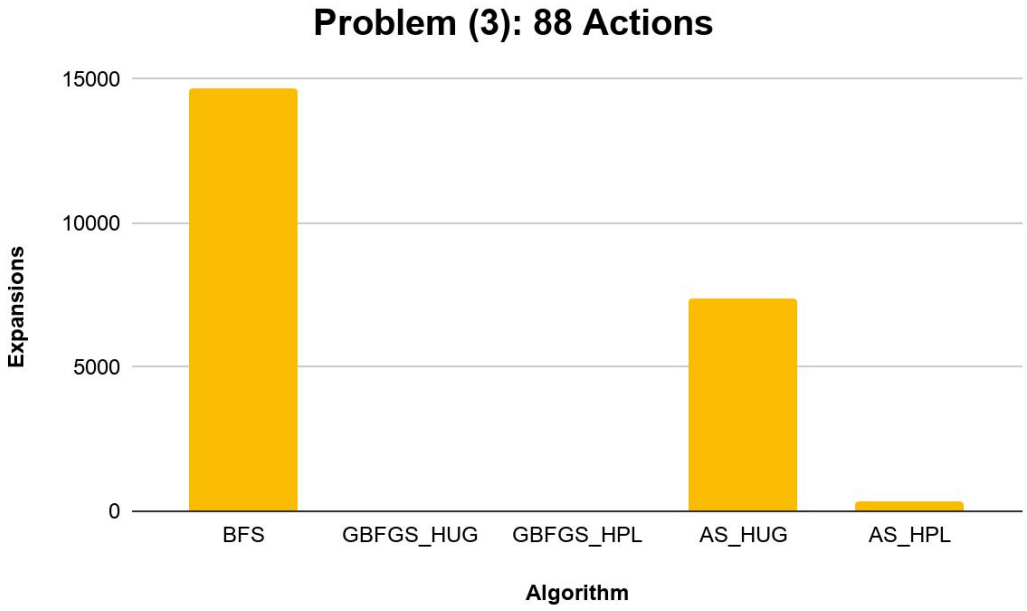
Algorithm	Problem (1)		Problem (2)	
	Expansions	Actions	Expansions	Actions
BFS	43	20	3343	72
DFGS	21	20	624	72
UCS	60	20	5154	72
GBFGS_HUG	7	20	17	72
GBFGS_HPL	6	20	9	72
GBFGS_HPM	6	20	27	72
GBFGS_HPS	6	20	9	72
AS_HUG	50	20	2467	72
AS_HPL	28	20	357	72
AS_HPM	43	20	2887	72
AS_HPS	33	20	1037	72

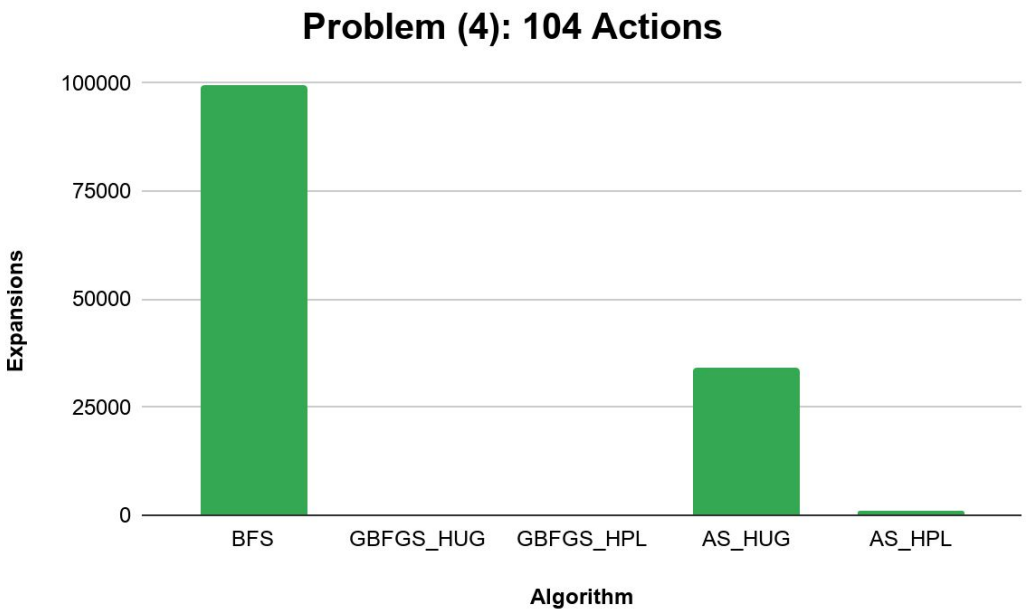
Algorithm	Problem (3)		Problem (4)	
	Expansions	Actions	Expansions	Actions
BFS	14663	88	99736	104
GBFGS_HUG	25	88	29	104
GBFGS_HPL	14	88	17	104
AS_HUG	7388	88	34330	104
AS_HPL	369	88	1208	104

b) Charts




Problem (2): 72 Actions
6000
4000
2000
0
Expansions
BFS
DFGS
UCS
GBFGS_HUG
GBFGS_HPL
GBFGS_HPM
GBFGS_HPS
AS_HUG
AS_HPL
AS_HPM
AS_HPS
Algorithm


Problem (3): 88 Actions
15000
10000
5000
0
Expansions
BFS
GBFGS_HUG
GBFGS_HPL
AS_HUG
AS_HPL
Algorithm


Problem (4): 104 Actions
100000
75000
50000
25000
0
Expansions
BFS
GBFGS_HUG
GBFGS_HPL
AS_HUG
AS_HPL
Algorithm

c) Analysis

- The expanded nodes increase by increasing the number of actions.
- This increase is significant in the case of using uninformed search or A* search algorithms.
- The increase is minor in the case of using greedy best first graph search algorithms.
- Greedy best first graph search produces the lowest number of expanded nodes.

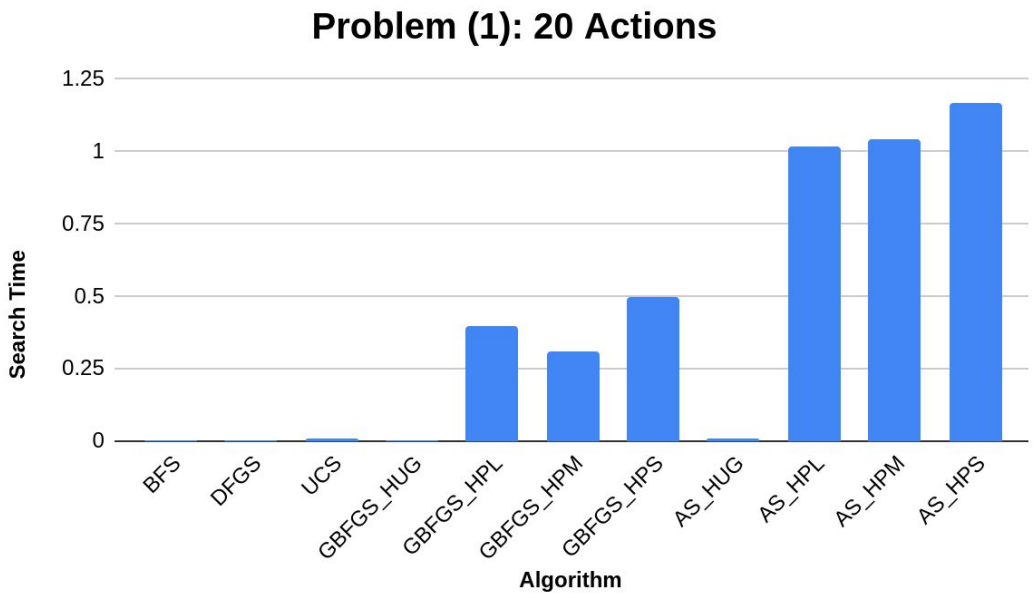
2) Search Time against number of actions

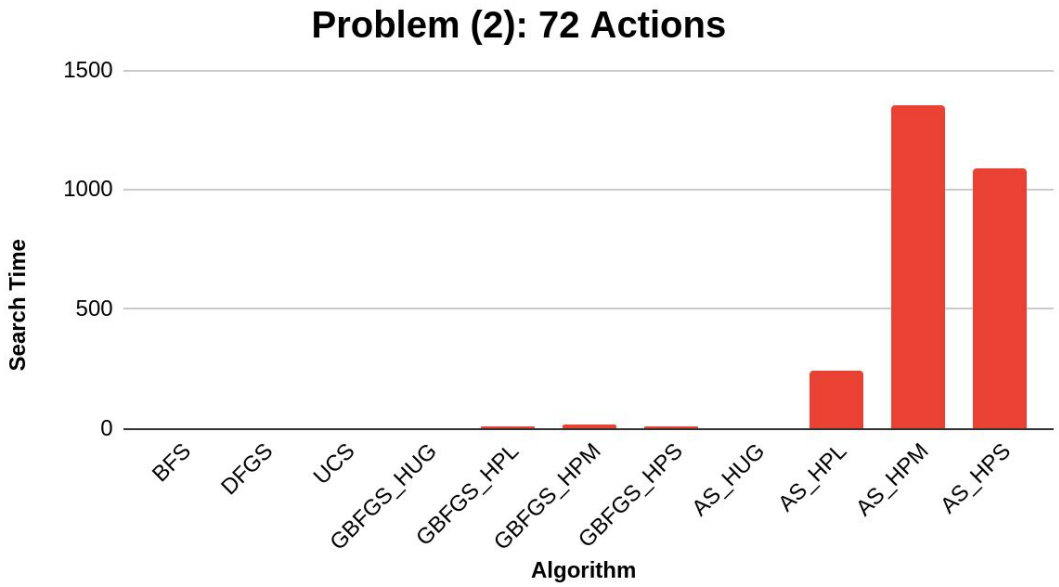
a) Tables

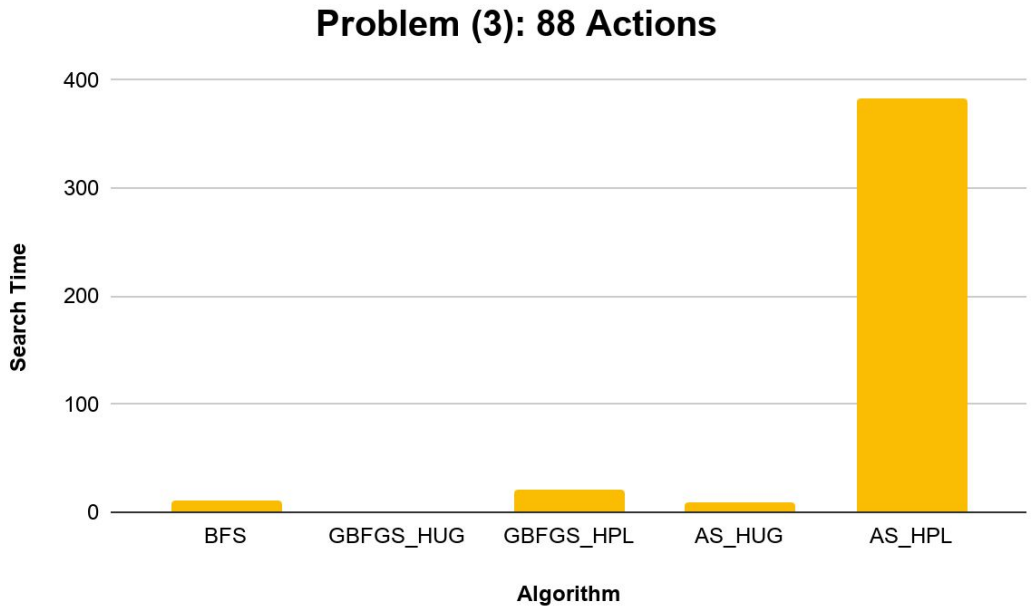
Algorithm	Problem (1)		Problem (2)	
	Search Time (s)	Actions	Search Time (s)	Actions
BFS	0.0063	20	2.09	72
DFGS	0.0034	20	3.13	72
UCS	0.01	20	3.42	72
GBFGS_HUG	0.002	20	0.019	72
GBFGS_HPL	0.4	20	9.41	72
GBFGS_HPM	0.31	20	19.01	72
GBFGS_HPS	0.5	20	12.33	72
AS_HUG	0.01	20	2.37	72
AS_HPL	1.02	20	239.81	72
AS_HPM	1.04	20	1356.75	72
AS_HPS	1.17	20	1091.75	72

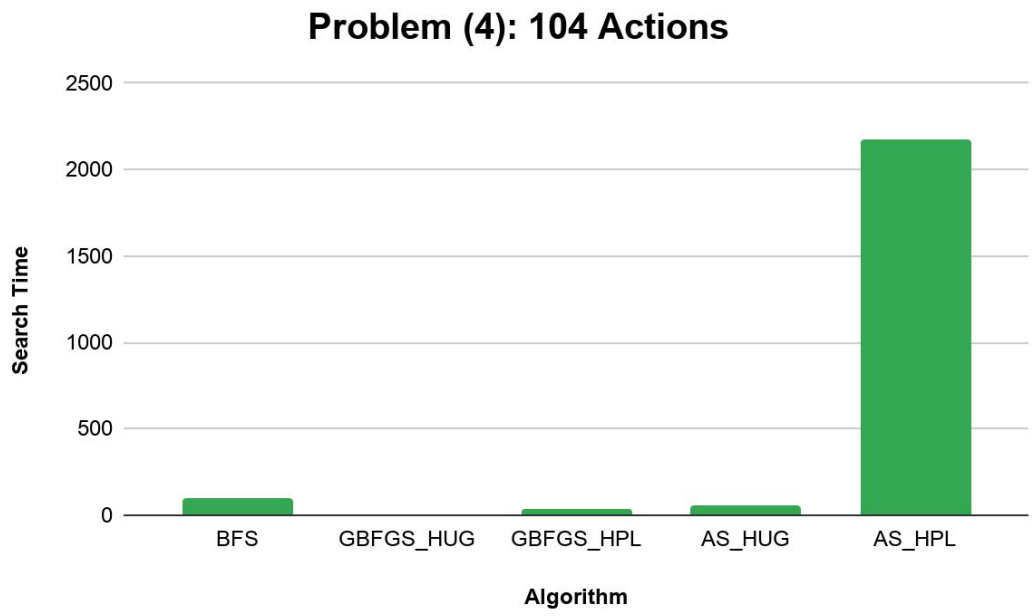
Algorithm	Problem (3)		Problem (4)	
	Search Time (s)	Actions	Search Time (s)	Actions
BFS	10.84	88	97.37	104
GBFGS_HUG	0.04	88	0.06	104
GBFGS_HPL	21.25	88	37.09	104
AS_HUG	8.64	88	56.6	104
AS_HPL	383.67	88	2170.49	104

b) Charts




Problem (2): 72 Actions
1500
1000
500
0
Search Time
BFS
DFGS
UCS
GBFGS_HUG
GBFGS_HPL
GBFGS_HPM
GBFGS_HPS
AS_HUG
AS_HPL
AS_HPM
AS_HPS
Algorithm


Problem (3): 88 Actions
400
300
200
100
0
Search Time
BFS
GBFGS_HUG
GBFGS_HPL
AS_HUG
AS_HPL
Algorithm



c) Analysis

- The search time increases by increasing the number of actions.
- All A* search algorithms, *except A* search with h_unmet_goals*, have more significant search time compared with other algorithms.
- A* search with h_unmet_goals has less search time compared with other A* search algorithms.

3) Length of plans against number of actions

a) Tables

Algorithm	Problem (1)		Problem (2)	
	Length	Actions	Length	Actions
BFS	6	20	9	72
DFGS	20	20	619	72
UCS	6	20	9	72
GBFGS_HUG	6	20	9	72
GBFGS_HPL	6	20	9	72
GBFGS_HPM	6	20	9	72

GBFGS_HPS	6	20	9	72
AS_HUG	6	20	9	72
AS_HPL	6	20	9	72
AS_HPM	6	20	9	72
AS_HPS	6	20	9	72

Algorithm	Problem (3)		Problem (4)	
	Length	Actions	Length	Actions
BFS	12	88	14	104
GBFGS_HUG	15	88	18	104
GBFGS_HPL	14	88	17	104
AS_HUG	12	88	14	104
AS_HPL	12	88	15	104

4) Questions

a) 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?

The most efficient algorithm for planning a very restricted domain is ***greedy best first graph search with h_unmet_goals*** as this algorithm has the lowest plan length with lowest search time.

b) 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)

The answer to this question depends on two main factors (1) How fast we need to generate the plan (2) How important we need to find the optimal plan length.

- ***Greedy best first graph search with h_unmet_goals*** is the fastest one, but it does not produce the optimal plan length.
- ***Astar search with h_unmet_goals*** this algorithm is the third quickest but produces the optimal plan length.

c) Which algorithm or algorithms would be most appropriate for planning problems where it is important to find only optimal plans?

Breadth first search *and* ***Astar search with h_unmet_goals algorithm*** *were able to find the optimal plans in all four cases.*