Experiments Report for Search Algorithms (for Air Cargo Problems)
Project: Build a Forward-Planning Agent

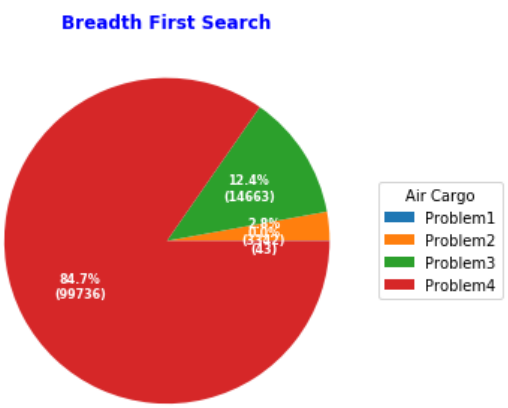
Student Name: Ashish Sarode
Nano-Degree name: Artificial Intelligence Nanodegree.

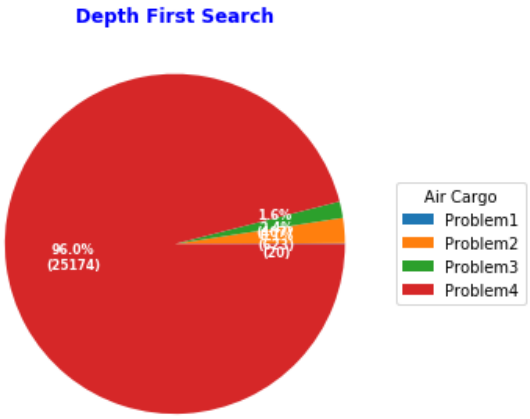
Conventions used in the report:

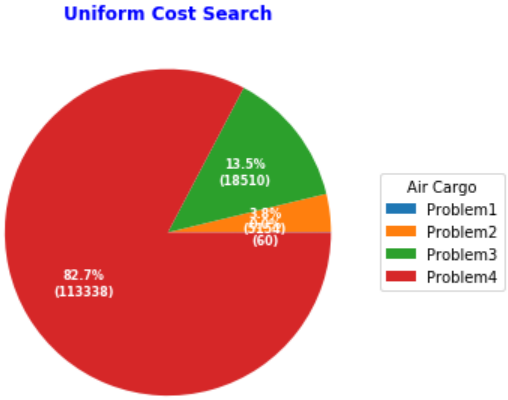
1. In tables,
   i. Green background indicates best performers
   ii. Red background indicates worse performers
   iii. Yellow background indicates, performance which is little worse.
2. Often in tables and charts, short form are used -
   i. BFS – Breadth first search.
   ii. DFGS – Depth first graph search.
   iii. UCS – Uniform cost search.
   iv. GBFGS_unmet – Greedy best first graph search with unmet goals heuristic.
   v. GBFGS_levelsum - Greedy best first graph search with levelsum heuristic.
   vi. GBFGS_maxlevel - Greedy best first graph search with maxlevel heuristic.
   vii. GBFGS_setlevel - Greedy best first graph search with setlevel heuristic.
   viii. AStar_unmet – A* search with unmet goals heuristic.
   ix. AStar_levelsum – A* search with levelsum heuristic.
3. Charts sometimes do not specifically mention graph version of search algorithm

Analyze the search complexity as function of domain size, search algorithm and heuristic.

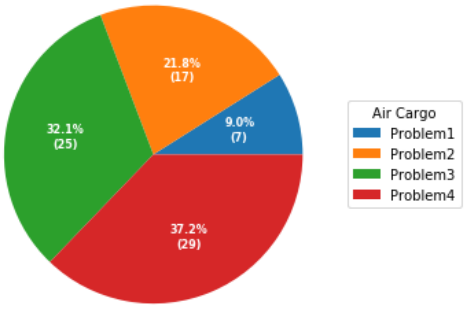
Search Algorithm	Heuristic	Air Cargo Problem Domain Sizes			
		Domain Size	Domain Size	Domain Size	Domain Size
		20	72	88	104
		Number of Nodes Expanded			
Breadth First	-	43	3343	14663	99736
Depth First Graph	-	21	624	408	25174
Uniform Cost	-	60	5154	18510	113339
Greedy Best First Graph	h_unmet_goals	7	17	25	29
Greedy Best First Graph	h_pg_levelsum	6	9	14	17
Greedy Best First Graph	h_pg_maxlevel	6	27	21	56
Greedy Best First Graph	h_pg_setlevel	6	9	35	107
A*	h_unmet_goals	50	2467	7388	34330
A*	h_pg_levelsum	28	357	369	1208
A*	h_pg_maxlevel	43	2887	-	-
A*	h_pg_setlevel	33	1037	-	-



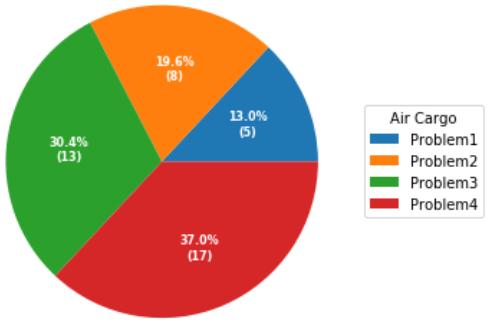




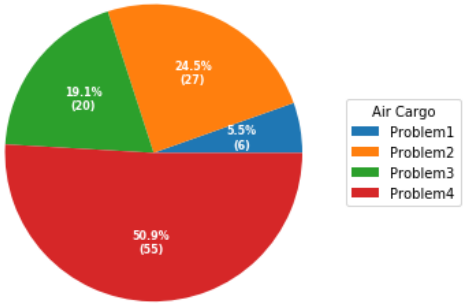
Greedy Best First Search | h_umet_goals



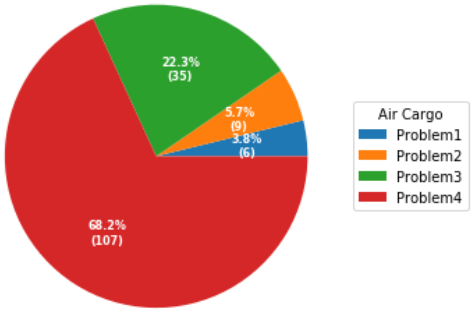
Greedy Best First Search | h_pg_levelsum



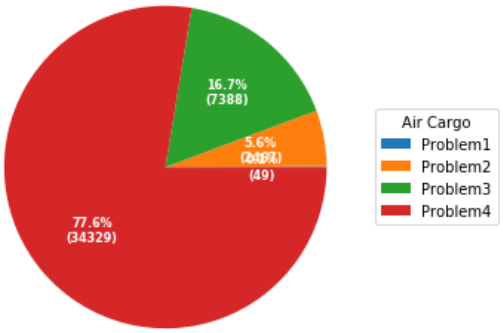
Greedy Best First Search | h_pg_maxlevel



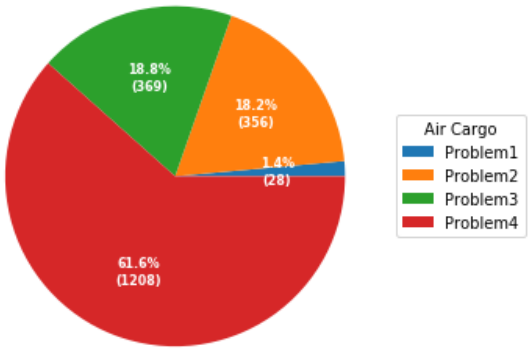
Greedy Best First Search | h_pg_setlevel

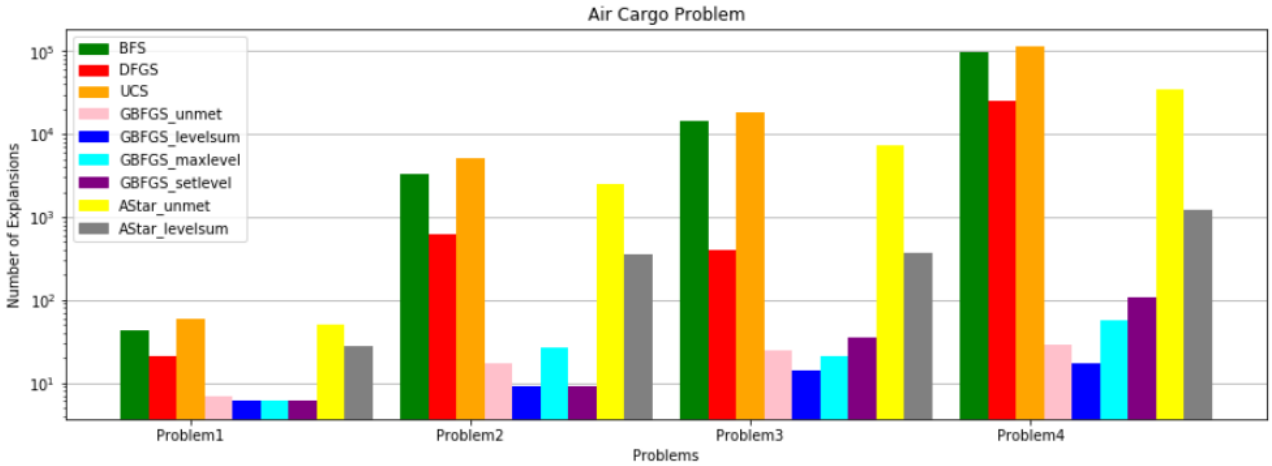


A* Search | h_umet_goals



A* Search | h_pg_levelsum

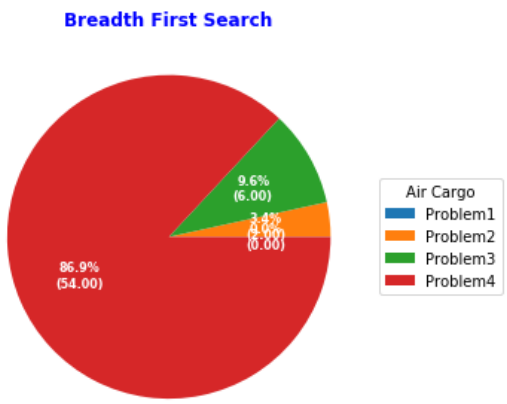


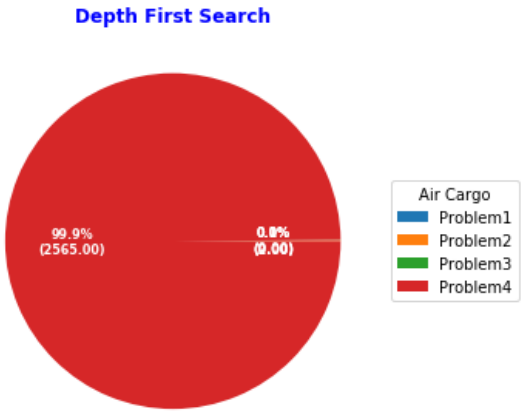


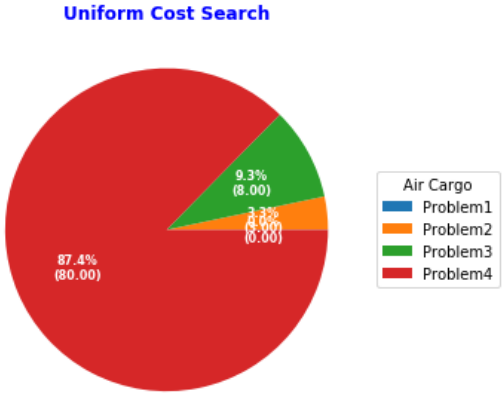
Breadth first search and Uniform cost search seems to be expanding maximum number of nodes, which is expected. Depth first search and A* with unmet goals heuristic seems to be little better than BFS and UCS. Greedy best first graph search algorithm's variants with different heuristic seems to be expanding minimum nodes.
Greedy best first search also seems to be good option, when number of domain size is higher, compared to other algorithms greedy best first search variants have less growth rate in expanded nodes.
From bar chart, it can be easily seen that Greedy best first graph search with level sum heuristic is consistently having least node expansions in all 4 domain sizes.

Analyze search time as function of domain size, search algorithm and heuristic. Analyze the optimality of solution as function of domain size, search algorithm and heuristic.

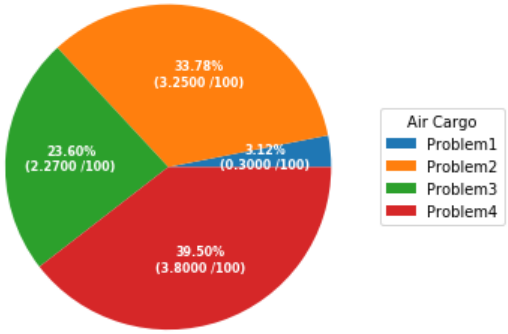
Search Algorithm	Heuristic	Air Cargo Problem Domain Sizes			
		Domain Size	Domain Size	Domain Size	Domain Size
		20	72	88	104
		Time in Seconds			
Breadth First	-	0.0108	2.1716	6.0775	54.9936
Depth First Graph	-	0.0057	2.4775	0.7007	2565.8379
Uniform Cost	-	0.0232	3.0419	8.5944	80.6372
Greedy Best First Graph	h_unmet_goals	0.003	0.0325	0.0227	0.038
Greedy Best First Graph	h_pg_levelsum	0.369	5.3392	8.4452	16.7019
Greedy Best First Graph	h_pg_maxlevel	0.2607	9.8461	9.6022	35.6485
Greedy Best First Graph	h_pg_setlevel	0.4673	9.2087	34.417	194.4106
A*	h_unmet_goals	0.0173	2.0187	5.0175	45.0188
A*	h_pg_levelsum	0.9444	228.4966	177.9669	1108.5266
A*	h_pg_maxlevel	0.7014	1745.0858	-	-
A*	h_pg_setlevel	0.9785	1605.1621	-	-



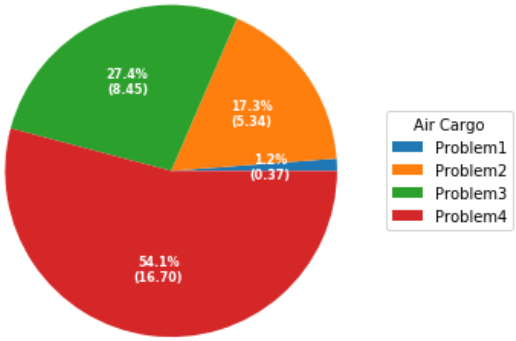




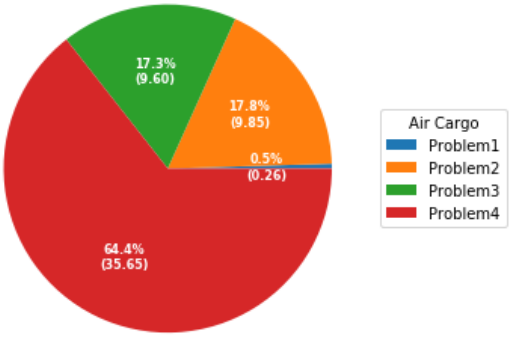
Greedy Best First Search | h_umet_goals



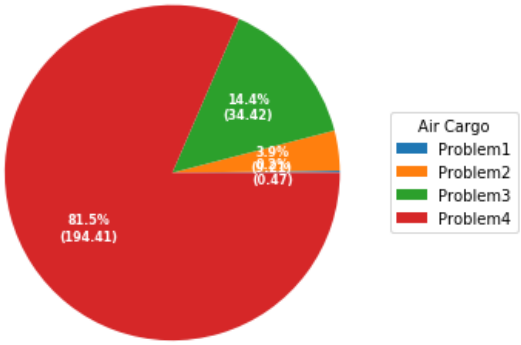
Greedy Best First Search | h_pg_levelsum



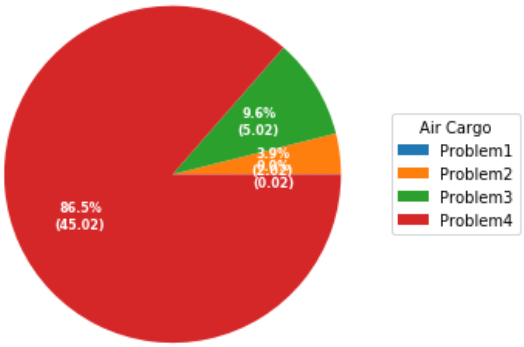
Greedy Best First Search | h_pg_maxlevel



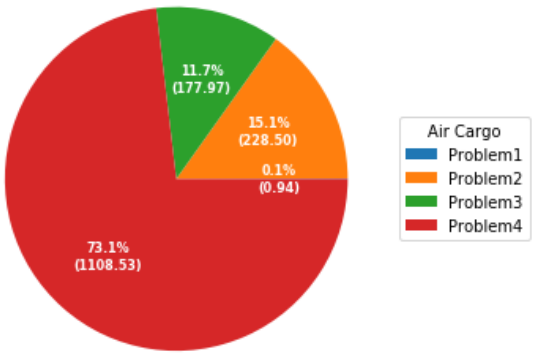
Greedy Best First Search | h_pg_setlevel

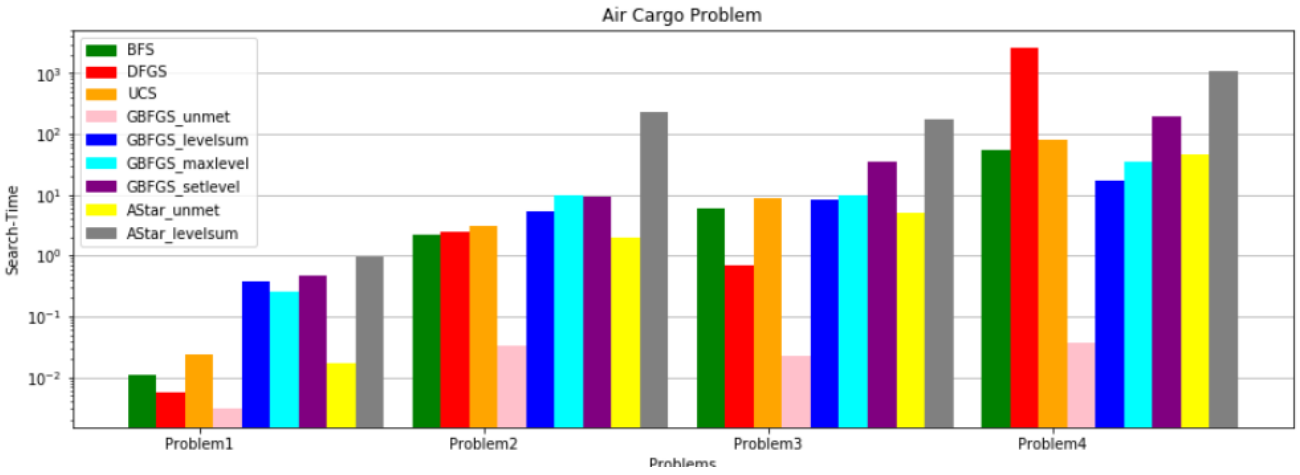


A* Search | h_umet_goals



A* Search | h_pg_levelsum

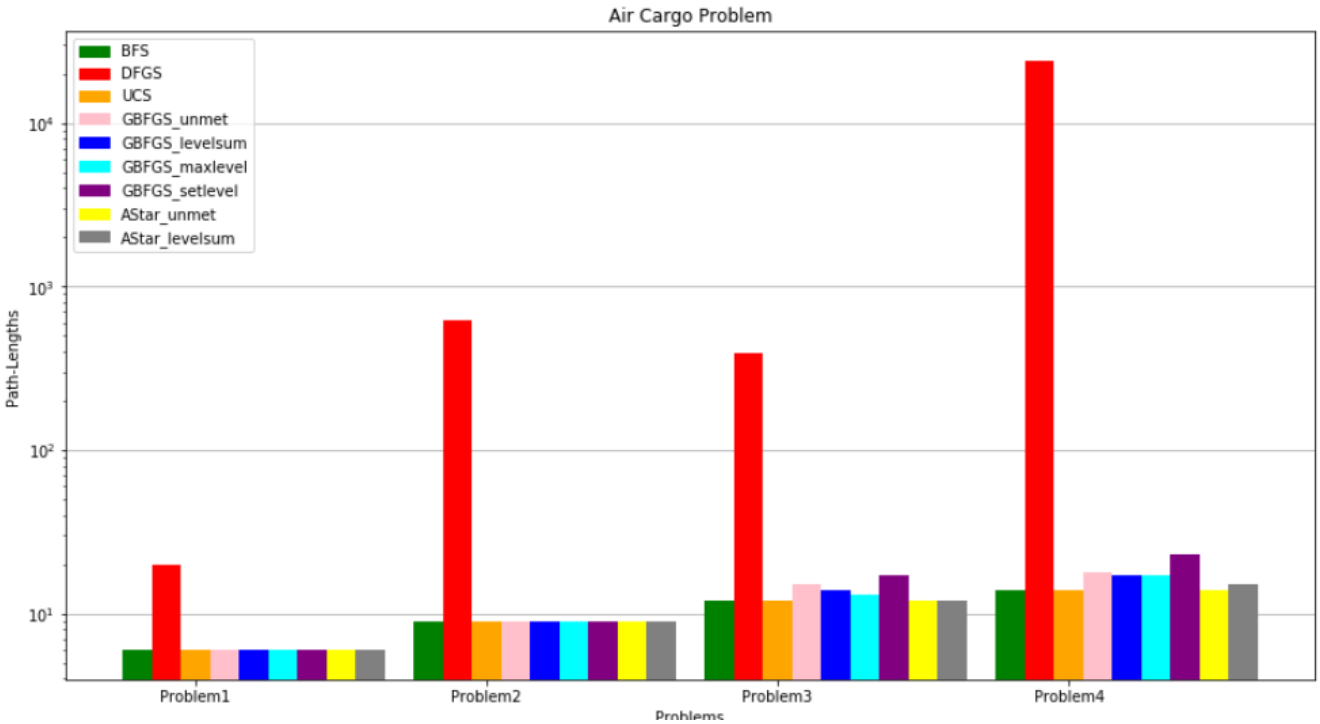




For search time performance, Greedy best first graph search with unmet goals heuristic seems to be performing best for all the domain sizes. Max level and levelsum heuristic variants of GBFGS are also ok, and seems to have predictable time estimate.
Although BFS and UCS are not great performers, their search time is predictable based on domain size.
DFGS seems to be unpredictable with time performance, although for large domain sizes it can take huge time and difficult to rely upon.
Maxlevel and setlevel variants of A* seems to be taking very high time as domain size increases, they were dropped for experiments when testing problem3 and problem4.

Analyze the optimality of solution as a function of domain size, search algorithm, and heuristic.

Search Algorithm	Heuristic	Air Cargo Problem Domain Sizes			
		Domain Size	Domain Size	Domain Size	Domain Size
		20	72	88	104
		Plan Length			
Breadth First	-	6	9	12	14
Depth First Graph	-	20	619	392	24132
Uniform Cost	-	6	9	12	14
Greedy Best First Graph	h_unmet_goals	6	9	15	18
Greedy Best First Graph	h_pg_levelsum	6	9	14	17
Greedy Best First Graph	h_pg_maxlevel	6	9	13	17
Greedy Best First Graph	h_pg_setlevel	6	9	17	23
A*	h_unmet_goals	6	9	12	14
A*	h_pg_levelsum	6	9	12	15
A*	h_pg_maxlevel	6	9	-	-
A*	h_pg_setlevel	6	9	-	-



For all the 4 problems, Breadth first search, Uniform cost search and A* search with unmet goals heuristic always seems to be giving optimal results. Although unmet goals heuristic might not be relied upon for optimal results as in case of GBFGS it did not give accurate optimal path lengths.
All variants of GBFGS and A* tested seems to be giving either optimal or close to optimal results.
As expected DFGS seems to be the most unreliable algorithm for getting optimal path length.

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?

Non-optimal choices

Greedy best first graph search with unmet goals heuristic can be good choice, if getting close to optimal solution is ok (it is not guaranteed to give optimal solution). Depth first graph search is also good in this case, but results can be far from optimal.

Optimal choices

Breadth first search and uniform cost search, which perform fairly well for small domain sizes and are guaranteed to be optimal. Since the problems are small in size, memory size is also not a problem for choosing BFS or UCS.

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)

Non-optimal choice

Greedy best first graph search with unmet goals heuristic can be excellent choice, if getting close to optimal solution is ok (it is not guaranteed to give optimal solution). This performs exceptionally well for large domain sizes. Another advantage of GBFGS with unmet goals heuristic here is small memory footprint.

Optimal choices

Breadth first search and uniform cost search, which perform fairly well for small domain sizes and are guaranteed to be optimal. Although it should not be a preferred choice if answers are required in real time, also for larger problem these algorithms have large memory footprints.

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

Definitely Breadth first search and uniform cost search are most appropriate for problem which only need optimal plans.