Report

Table of Content

Part 1. Understanding The Problems

Problem 1:

The goal of this problem is to transport the cargo ‘C1’ which is present at airport ‘SFO’ to airport ‘JFK’ and cargo ‘C2’ which is present at airport ‘JFK’ to airport ‘SFO’. All of this task is to be done using planes ‘P1’ and ‘P2’ which are currently present at airport ‘SFO’ and ‘JFK’ respectively. A symbolic solution is to be created using three actions, which include ‘Load’, ‘Unload’ and ‘Fly’.

Symbolic Pre-Condition:

Cargo(C1) ∧ Cargo(C2) ∧
Plane(P1) ∧ Plane(P2) ∧
Airport(JFK) ∧ Airport(SFO) ∧
At(C1, SFO) ∧ At(C2, JFK) ∧ At(P1, SFO) ∧ At(P2, JFK)

Symbolic Solution:

At(C1, JFK) ∧ At(C2, SFO)

Optimal Solution:

The Optimal plan Length is 6 actions. These involves

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:

The goal of this problem is to transport the cargo ‘C1’ which is present at airport ‘SFO’ to airport ‘JFK’, cargo ‘C2’ which is present at airport ‘JFK’ to airport ‘SFO’ and cargo ‘C3’ which is present at airport ‘ATL’ to airport ‘SFO’. All of this task is to be done using planes ‘P1’, ‘P2’ and ‘P3’ which are currently present at airport ‘SFO’, ‘JFK’ and ‘ATL’ respectively. A symbolic solution is to be created using three actions, which include ‘Load’, ‘Unload’ and ‘Fly’.

Symbolic Pre-Condition:

Cargo(C1) ∧ Cargo(C2) ∧ Cargo(C3) ∧
Plane(P1) ∧ Plane(P2) ∧ Plane(P3) ∧
Airport(JFK) ∧ Airport(SFO) ∧ Airport(ATL) ∧
At(C1, SFO) ∧ At(C2, JFK) ∧ At(C3, ATL) ∧ At(P1, SFO) ∧ At(P2, JFK) ∧ At(P3, ATL)

Symbolic Solution:

At(C1, JFK) ∧ At(C2, SFO) ∧ At(C3, SFO)

Optimal Solution:

The Optimal plan Length is 9 actions. These involves

```
Load(C3, P3, ATL)
Fly(P3, ATL, SFO)
Unload(C3, P3, SFO)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
Unload(C2, P2, SFO)
Load(C1, P3, SFO)
Fly(P3, SFO, JFK)
Unload(C1, P3, JFK)
```

<u>Problem 3:</u>

The goal of this problem is to transport the cargo ‘C1’ which is present at airport ‘SFO’ to airport ‘JFK’, cargo ‘C2’ which is present at airport ‘JFK’ to airport ‘SFO’, cargo ‘C3’ which is present at airport ‘ATL’ to airport ‘JFK’ and cargo ‘C4’ which is present at airport ‘ORD’ to airport ‘SFO’. All of this task is to be done using planes ‘P1’ and ‘P2’ which are currently present at airport ‘SFO’ and ‘JFK’ respectively. A symbolic solution is to be created using three actions, which include ‘Load’, ‘Unload’ and ‘Fly’.

Symbolic Pre-Condition:

```
Cargo(C1) ∧ Cargo(C2) ∧ Cargo(C3) ∧ Cargo(C4) ∧
Plane(P1) ∧ Plane(P2)∧
Airport(JFK) ∧ Airport(SFO) ∧ Airport(ATL) ∧ Airport(ORD) ∧
At(C1, SFO) ∧ At(C2, JFK) ∧ At(C3, ATL) ∧ At(C4, ORD) ∧ At(P1, SFO) ∧ At(P2, JFK)
```

Symbolic Solution:

```
At(C1, JFK) ∧ At(C2, SFO) ∧ At(C3, JFK) ∧ At(C4, SFO)
```

Optimal Solution:

The Optimal plan Length is 12 actions. These involves

```
Load(C2, P2, JFK)
Fly(P2, JFK, ATL)
Load(C3, P2, ATL)
Fly(P2, ATL, ORD)
Load(C4, P2, ORD)
Fly(P2, ORD, SFO)
Unload(C4, P2, SFO)
Unload(C2, P2, SFO)
Load(C1, P2, SFO)
Fly(P2, SFO, JFK)
Unload(C3, P2, JFK)
Unload(C1, P2, JFK)
```

Problem 4:

The goal of this problem is to transport the cargo ‘C1’ which is present at airport ‘SFO’ to airport ‘JFK’, cargo ‘C2’ which is present at airport ‘JFK’ to airport ‘SFO’, cargo ‘C3’ which is present at airport ‘ATL’ to airport ‘JFK’, cargo ‘C4’ which is present at airport ‘ORD’ to airport ‘SFO’ and cargo ‘C5’ which is present at airport ‘ORD’ to airport ‘JFK’. All of this task is to be done using planes ‘P1’ and ‘P2’ which are currently present at airport ‘SFO’ and ‘JFK’ respectively. A symbolic solution is to be created using three actions, which include ‘Load’, ‘Unload’ and ‘Fly’.

Symbolic Pre-Condition:

Cargo(C1) ∧ Cargo(C2) ∧ Cargo(C3) ∧ Cargo(C4) ∧ Cargo(C5) ∧
Plane(P1) ∧ Plane(P2)∧
Airport(JFK) ∧ Airport(SFO) ∧ Airport(ATL) ∧ Airport(ORD) ∧
At(C1, SFO) ∧ At(C2, JFK) ∧ At(C3, ATL) ∧ At(C4, ORD) ∧ At(C5, ORD) ∧ At(P1, SFO) ∧ At(P2, JFK)

Symbolic Solution:

At(C1, JFK) ∧ At(C2, SFO) ∧ At(C3, JFK) ∧ At(C4, SFO) ∧ At(C5, JFK)

Optimal Solution:

The Optimal plan Length is 14 actions. These involves

Load(C2, P2, JFK)
Fly(P2, JFK, ATL)
Load(C3, P2, ATL)
Fly(P2, ATL, ORD)
Load(C4, P2, ORD)
Load(C5, P2, ORD)
Fly(P2, ORD, SFO)
Unload(C4, P2, SFO)
Unload(C2, P2, SFO)
Load(C1, P2, SFO)
Fly(P2, SFO, JFK)
Unload(C5, P2, JFK)
Unload(C3, P2, JFK)
Unload(C1, P2, JFK)

Part 2. Understanding Results

As highlighted in part 1, we are provided with four logistic planning problems for air cargo transportation system where the task is to find optimal path for the cargo delivery. We are to use agents to perform progressive search to solve these planning problems while using symbolic logic and classical search, and also perform comparison with each other. The various strategies that are used involves:

- (Uninformed Search) Breadth First Search
- (Uninformed Search) Depth First Search
- (Uninformed Search) Uniform Cost Search
- (Informed Search) Greedy Best First Graph Search with UnmetGoals heuristic
- (Informed Search) Greedy Best First Graph Search with LevelSum heuristic
- (Informed Search) Greedy Best First Graph Search with MaxLevel heuristic
- (Informed Search) Greedy Best First Graph Search with SetLevel heuristic
- (Informed Search) A* Search with UnmetGoals heuristic
- (Informed Search) A* Search with LevelSum heuristic
- (Informed Search) A* Search with MaxLevel heuristic
- (Informed Search) A* Search with SetLevel heuristic

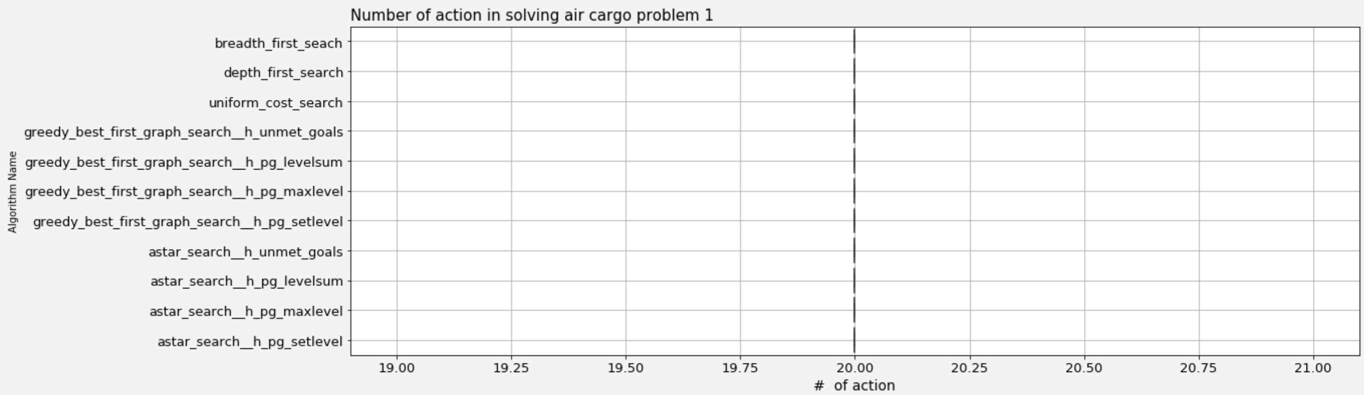
We compare the performance of these eleven strategies in terms of speed (execution time), memory usage (search node expansion), and optimality (is plan length optimal). The number of goal tests and number of new nodes are reported but won't be considered in analysis as they do not change the results.

2.1. Results Problem Wise

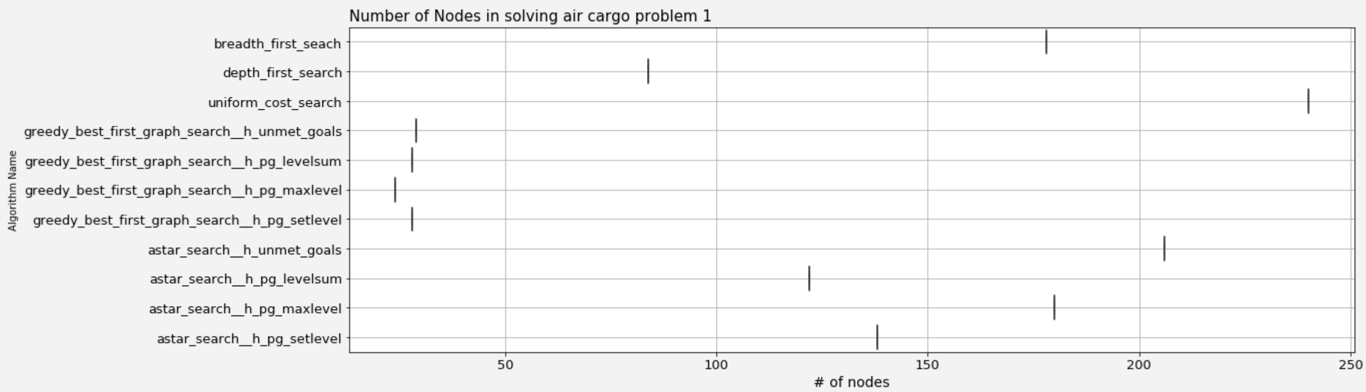
2.1.1. Problem 1:

Visualization

Graph1. Number of action vs Algorithm used

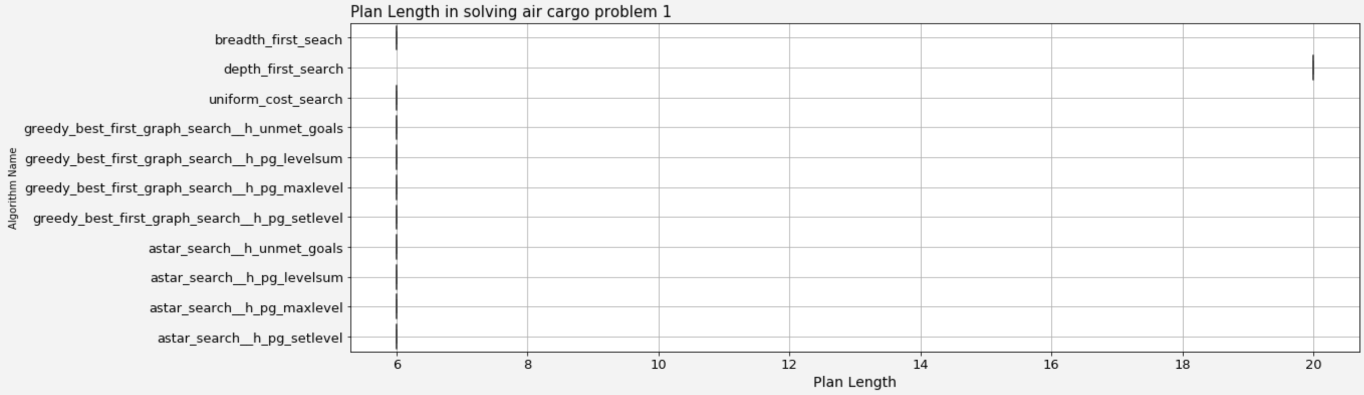


Graph2. Number of nodes vs Algorithm used

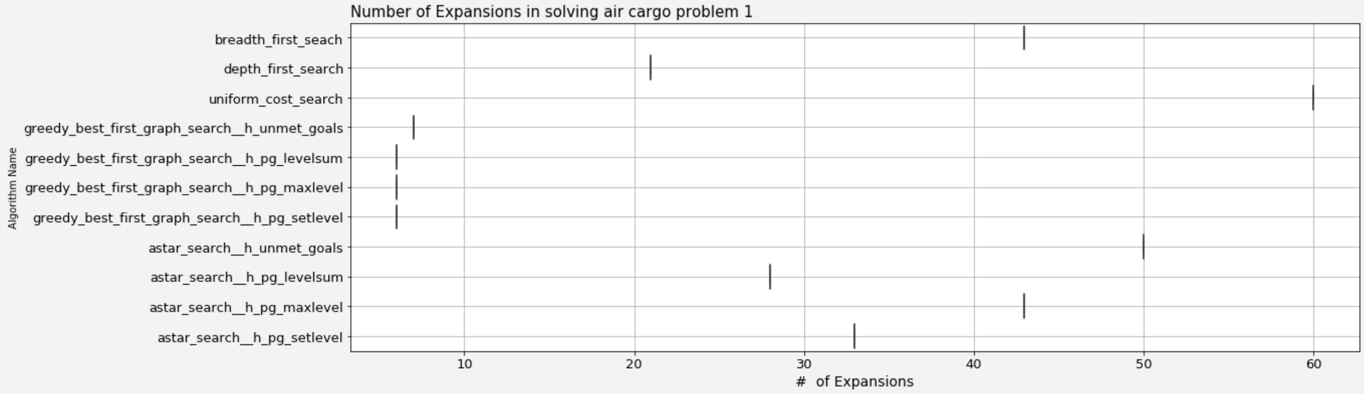


Graph3. Time Consumed (sec) vs Algorithm used

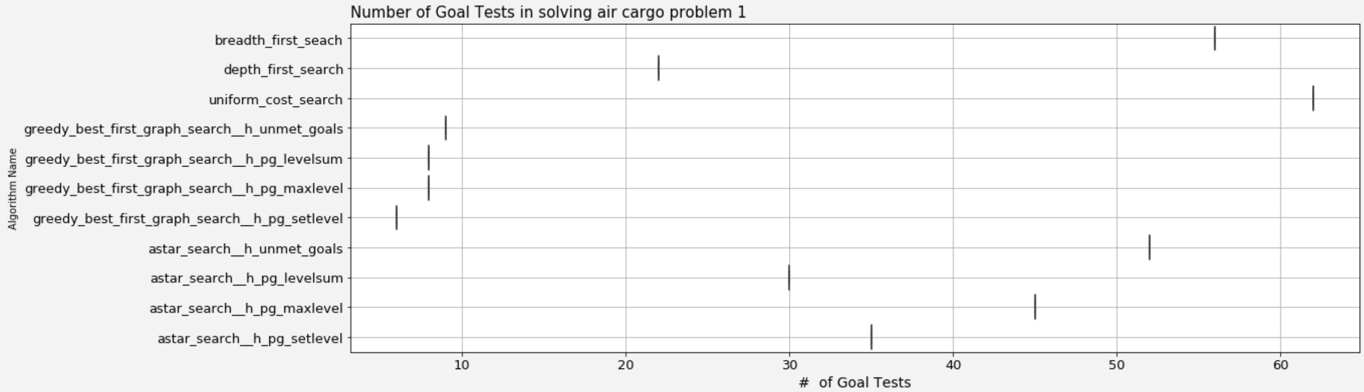
Graph4. Plan Length vs Algorithm used



Graph5. Number of Expansions vs Algorithm used



Graph6. Number of Goal tests vs Algorithm used



Tabulation

Algorithm Name	Actions	Expansions	Goal Tests	New Nodes	Plan Length	Avg Elap.Time
breadth_first_seach	20	43	56	178	6	0.007342
depth_first_search	20	21	22	84	20	0.004207
uniform_cost_search	20	60	62	240	6	0.009724
greedy_best_first_graph_search__h_unmet_goals	20	7	9	29	6	0.002685
greedy_best_first_graph_search__h_pg_levelsum	20	6	8	28	6	0.39906
greedy_best_first_graph_search__h_pg_maxlevel	20	6	8	24	6	0.295184
greedy_best_first_graph_search__h_pg_setlevel	20	6	6	28	6	0.522449
astar_search__h_unmet_goals	20	50	52	206	6	0.009492
astar_search__h_pg_levelsum	20	28	30	122	6	1.011474
astar_search__h_pg_maxlevel	20	43	45	180	6	1.028062
astar_search__h_pg_setlevel	20	33	35	138	6	1.26379

Comparison of Strategies

Between Uninformed Search Strategies

When comparing uninformed search strategies based on the above plots and table, it can be seen that 'Depth First Search' strategy is performing best in terms of speed i.e. time consumption and also based on memory utilized i.e. data storage. But the solution found by this strategy is not optimal. Among the strategies which found the optimal solution, 'Breadth First Search' strategy performs the best in terms of speed and memory consumed while 'Uniform Cost Search' lags in these criteria.

Between Informed Search Strategies

When comparing informed search strategies with multiple heuristics based on the above plots and table, it can be seen that both 'Greedy Best First Graph Search' and 'A* Search' strategies are able to find optimal solution. Though on overall basis 'Greedy Best First Graph Search' strategies are performing better than 'A* Search' strategies in terms on finding optimal solution, with lesser memory consumption and better searching speed.

Between Heuristics in Informed Search Strategies

Both 'Greedy Best First Graph Search' and 'A* Search' strategies with every heuristic, namely 'UnmetGoals', 'LevelSum', 'MaxLevel', and 'SetLevel' are able to find optimal solution. Additionally in both strategies heuristic 'UnmetGoals' is seeming to be performing best in terms of time consumption and worst in terms of memory consumption. While heuristic 'SetLevel' in comparison to other heuristics is seeming to perform worst in terms for time consumption and is also seeming to perform moderately in regarding memory consumption.

In regards to informed search in this problem, 'Greedy Best First Graph Search' with 'UnmetGoals' heuristic is seeming to perform best in terms of finding optimal solution and with least amount of time consumption with a bit larger memory consumption.

Between Uninformed and Informed Search Strategies

Since this problem is fairly small, uninformed search strategies were able to find the optimal solution relatively earlier than informed search strategies. While in general uninformed search strategies are consuming larger memory i.e. expanding larger number of nodes compared in comparison to 'Greedy Best First Graph Search' strategies. Additionally, all informed search strategies were able to find optimal solution, which wasn't the case with 'Depth First Search' search strategy.

Best Strategy

Summarizing the results in the table below

Algorithm Name	Memory Consumption	Optimal Solution	Time Consumption
breadth_first_seach		BU, OB	
depth_first_search	BU		BU
uniform_cost_search		BU, OB	
greedy_best_first_graph_search__h_unmet_goals	CBI, COB	BI, OB	BI, OB
greedy_best_first_graph_search__h_pg_levelsum	BI, OB	BI, OB	
greedy_best_first_graph_search__h_pg_maxlevel	BI, OB	BI, OB	
greedy_best_first_graph_search__h_pg_setlevel	BI, OB	BI, OB	
astar_search__h_unmet_goals		BI, OB	
astar_search__h_pg_levelsum		BI, OB	
astar_search__h_pg_maxlevel		BI, OB	
astar_search__h_pg_setlevel		BI, OB	
OB: Overall Best, COB: Close to Overall Best, BU: Best in Uninformed Search, CBU: Close to Best in Uninformed Search, BI: Best in Informed Search, CBI: Close to Best in Informed Search.			

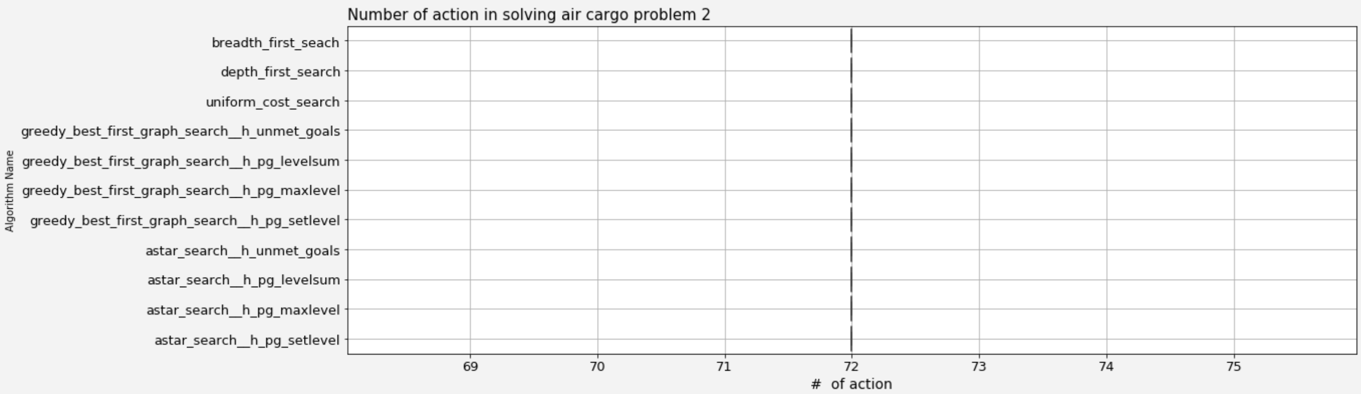
Assuming finding optimal solution is a must, the best strategy to use in the following setting

- In least memory possible with no bound on time consumed:
  ‘Greedy Best First Graph Search’ with ‘MaxLevel’ heuristic
- Fastest possible response with no constraint on memory:
  ‘Greedy Best First Graph Search’ with ‘UnmetGoals’ heuristic
- Best balance between time and memory consumed:
  ‘Greedy Best First Graph Search’ with ‘UnmetGoals’ heuristic

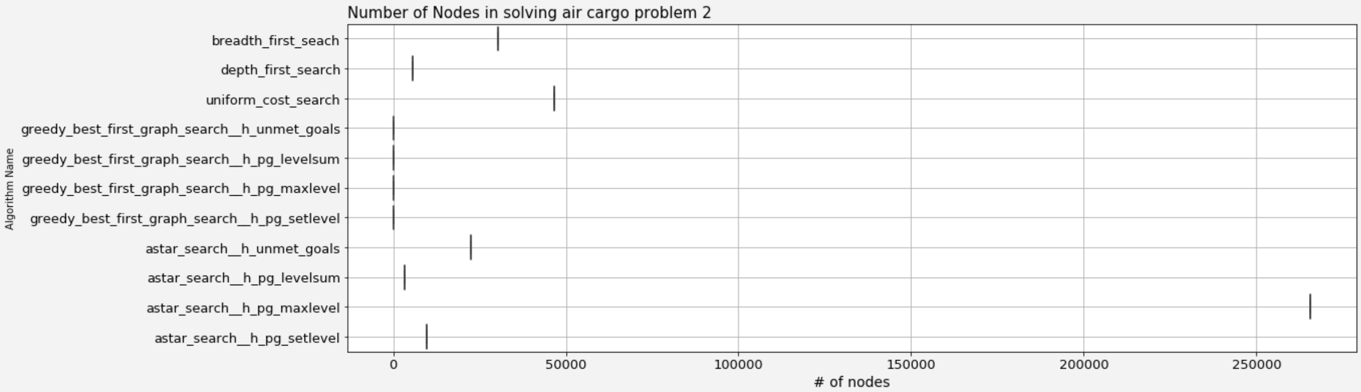
2.1.2. Problem 2:

Visualization

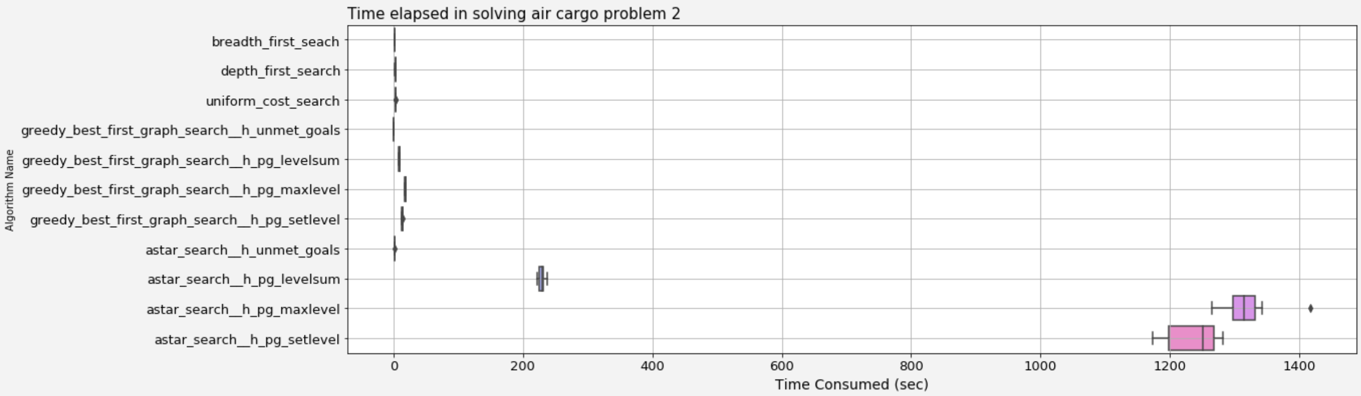
Graph1. Number of action vs Algorithm used



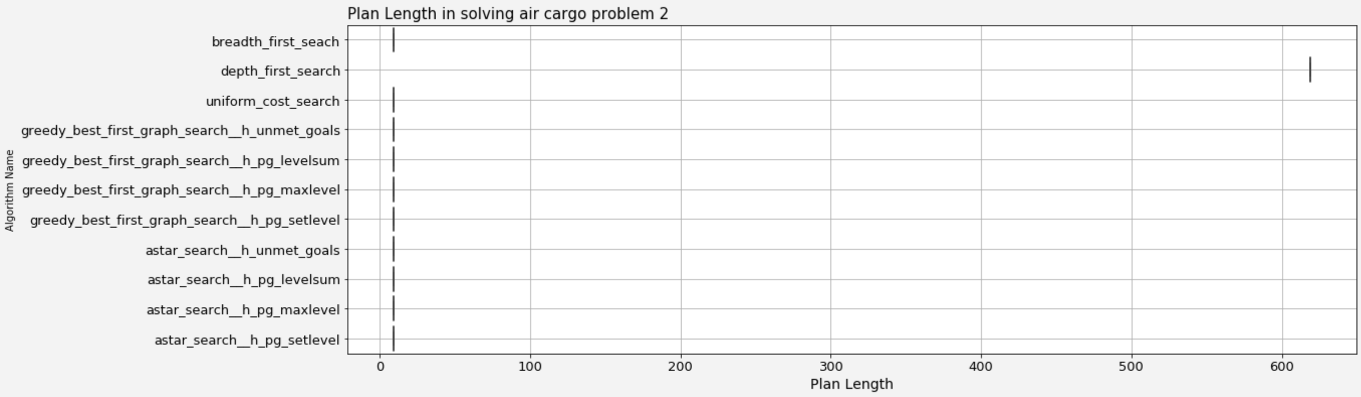
Graph2. Number of nodes vs Algorithm used



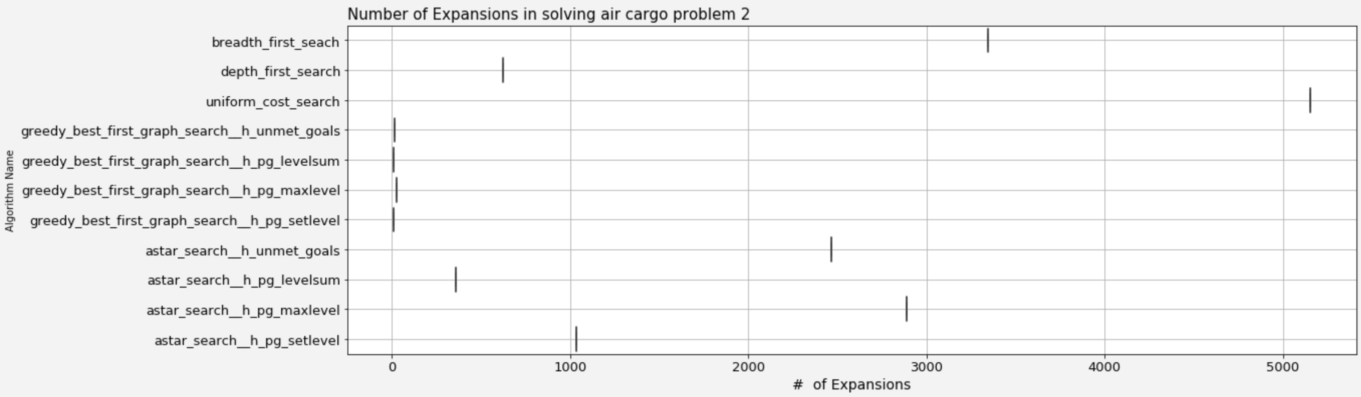
Graph3. Time Consumed (sec) vs Algorithm used



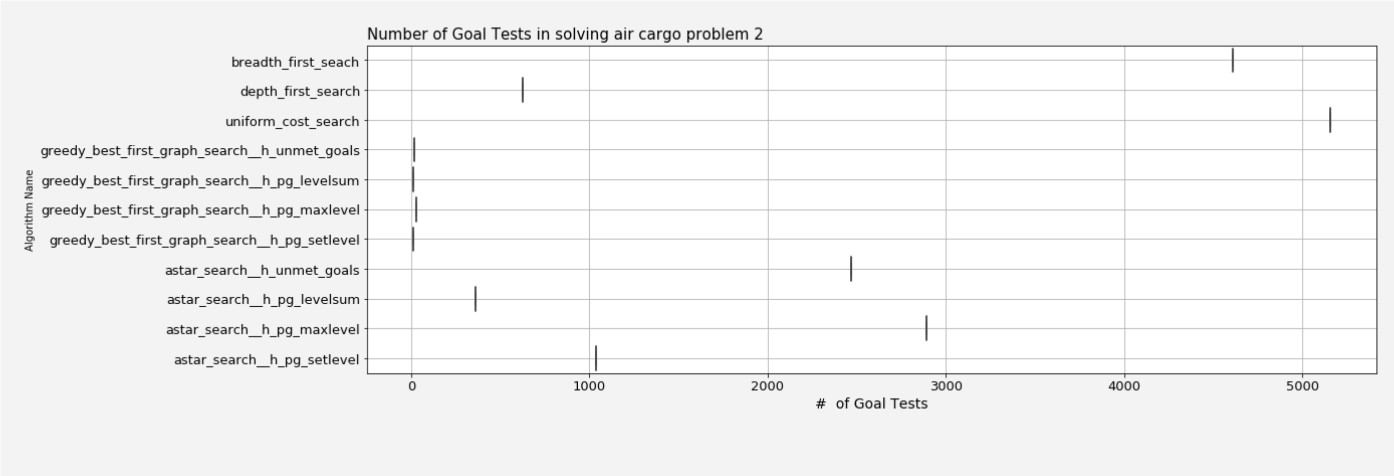
Graph4. Plan Length vs Algorithm used



Graph5. Number of Expansions vs Algorithm used



Graph6. Number of Goal tests vs Algorithm used



Tabulation

Algorithm Name	Actions	Expansions	Goal Tests	New Nodes	Plan Length	Avg Elap.Time
breadth_first_seach	72	3343	4609	30503	9	1.735581
depth_first_search	72	624	625	5602	619	2.660274
uniform_cost_search	72	5154	5156	46618	9	3.007047
greedy_best_first_graph_search__h_unmet_goals	72	17	19	170	9	0.018661
greedy_best_first_graph_search__h_pg_levelsum	72	9	11	86	9	8.914722
greedy_best_first_graph_search__h_pg_maxlevel	72	27	29	249	9	18.162882
greedy_best_first_graph_search__h_pg_setlevel	72	9	11	84	9	13.633953
astar_search__h_unmet_goals	72	2467	2469	22522	9	1.999412
astar_search__h_pg_levelsum	72	357	359	3426	9	229.055023
astar_search__h_pg_maxlevel	72	2887	2889	265594	9	1318.754408
astar_search__h_pg_setlevel	72	1037	1039	9605	9	1236.158324

Comparison of Strategies

Between Uninformed Search Strategies

When comparing uninformed search strategies based on the above plots and table, it can be seen that among the strategies which are able to find the optimal solution 'Breadth First Search' strategy is performing best in terms of speed i.e. time consumption and also based on memory utilized i.e. data storage. Among the strategies which found the optimal solution 'Uniform Cost Search' lags in memory utilized and time consumption. Additionally, 'Depth First Search' is performing the best in terms of memory utilization but is performing worst in terms of finding optimal solution.

Between Informed Search Strategies

When comparing informed search strategies with multiple heuristics based on the above plots and table, it can be seen that both 'Greedy Best First Graph Search' and 'A* Search' strategies are able to find optimal solution. Though on overall basis 'Greedy Best First Graph Search' strategies are performing better than 'A* Search' strategies in terms on finding optimal solution, with lesser memory consumption and better searching speed.

Between Heuristics in Informed Search Strategies

Both 'Greedy Best First Graph Search' and 'A* Search' strategies with every heuristic, namely 'UnmetGoals', 'LevelSum', 'MaxLevel', and 'SetLevel' are able to find optimal solution. Additionally in both strategies heuristic 'UnmetGoals' is seeming to be performing best in terms of time consumption and moderately bad in terms of memory consumption. Heuristic 'LevelSum' is performing the best in terms of memory consumption, while also finding the optimal solution but this heuristic take more time compared to 'UnmetGoals' heuristic. While heuristic 'MaxLevel' in

comparison to other heuristics is seeming to perform worst in terms of memory and time consumption.
In regards to informed search in this problem, ‘Greedy Best First Graph Search’ with ‘UnmetGoals’ heuristic is able to find optimal solution while consuming least amount of time with a bit larger memory consumption. While ‘Greedy Best First Graph Search’ with ‘LevelSum’ and ‘SetLevel’ heuristics are also able to find optimal solution and this while having least memory consumption with somewhat larger time consumption. ‘LevelSum’ heuristic consume less time compared to ‘SetLevel’ heuristic with ‘Greedy Best First Graph Search’. So based on the need for storage and runtime, the decision for the best heuristics could be different.

Between Uninformed and Informed Search Strategies

In this problem, ‘Greedy Best First Graph Search’ strategy with ‘UnmetGoals’ heuristic was able to find the optimal solution relatively much earlier than any other informed or uninformed search strategies. Additionally, ‘Greedy Best First Graph Search’ with ‘LevelSum’ heuristic and ‘Greedy Best First Graph Search’ with ‘SetLevel’ heuristic are also able to find the optimal solution and they achieve this with lower memory consumption but taking longer time. In uninformed strategies ‘Breadth First Search’ and ‘Uniform Cost Search’ are also able to find the optimal solution and they do this in lesser time but they utilise large amount of memory in achieving this feat. So based on the need for storage and runtime, the decision for the best heuristics could be different.

<u>Best Strategy</u>

Summarizing the results in the table below

Algorithm Name	Memory Consumption	Optimal Solution	Time Consumption
breadth_first_seach		BU, OB	BU
depth_first_search	BU		
uniform_cost_search		BU, OB	
greedy_best_first_graph_search__h_unmet_goals	CBI, COB	BI, OB	BI, OB
greedy_best_first_graph_search__h_pg_levelsum	BI, OB	BI, OB	
greedy_best_first_graph_search__h_pg_maxlevel		BI, OB	
greedy_best_first_graph_search__h_pg_setlevel	BI, OB	BI, OB	
astar_search__h_unmet_goals		BI, OB	
astar_search__h_pg_levelsum		BI, OB	
astar_search__h_pg_maxlevel		BI, OB	
astar_search__h_pg_setlevel		BI, OB	
OB: Overall Best, COB: Close to Overall Best, BU: Best in Uninformed Search, CBU: Close to Best in Uninformed Search, BI: Best in Informed Search, CBI: Close to Best in Informed Search.			

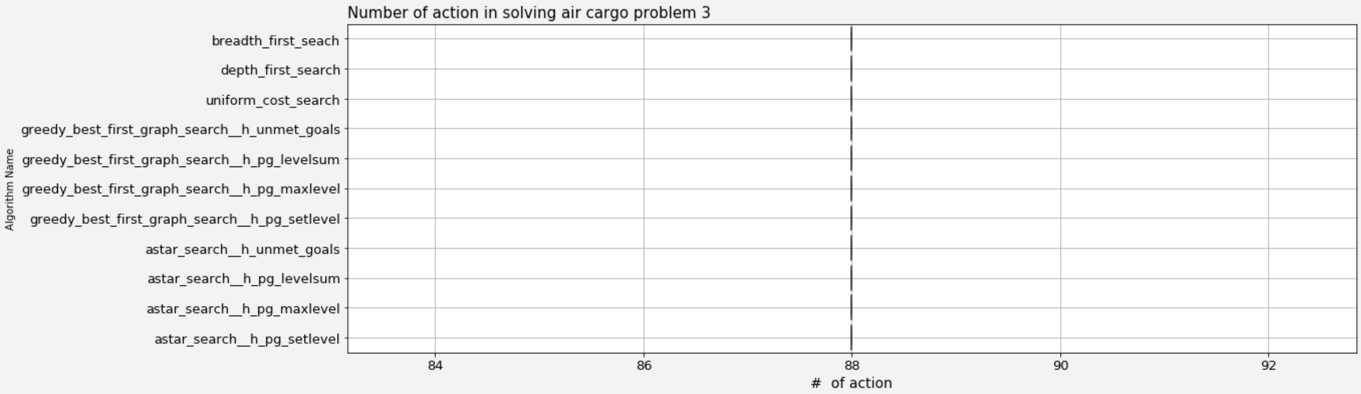
Assuming finding optimal solution is a must, the best strategy to use in the following setting

- In least memory possible with no bound on time consumed:
  ‘Greedy Best First Graph Search’ with ‘LevelSum’ heuristic
- Fastest possible response with no constraint on memory:
  ‘Greedy Best First Graph Search’ with ‘UnmetGoals’ heuristic
- Best balance between time and memory consumed:
  ‘Greedy Best First Graph Search’ with ‘UnmetGoals’ heuristic

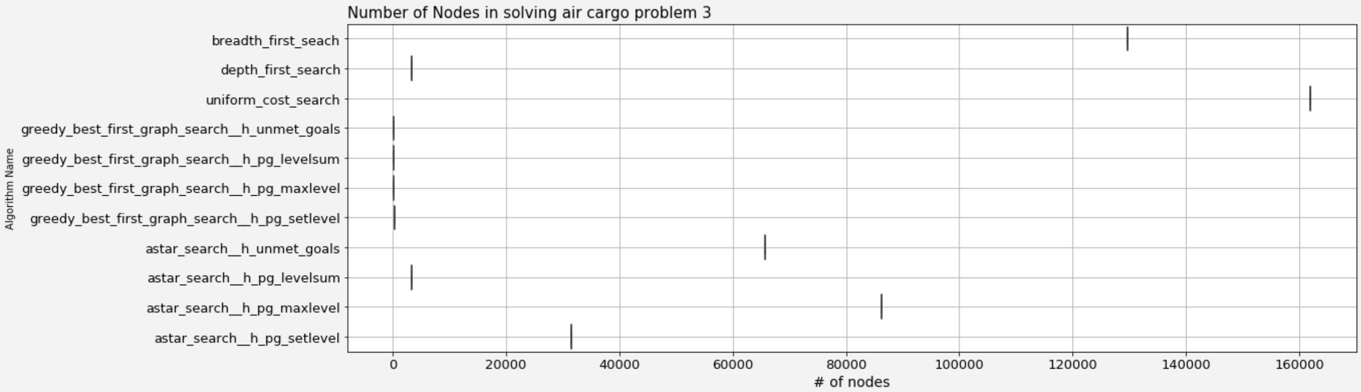
2.1.3. Problem 3:

Visualization

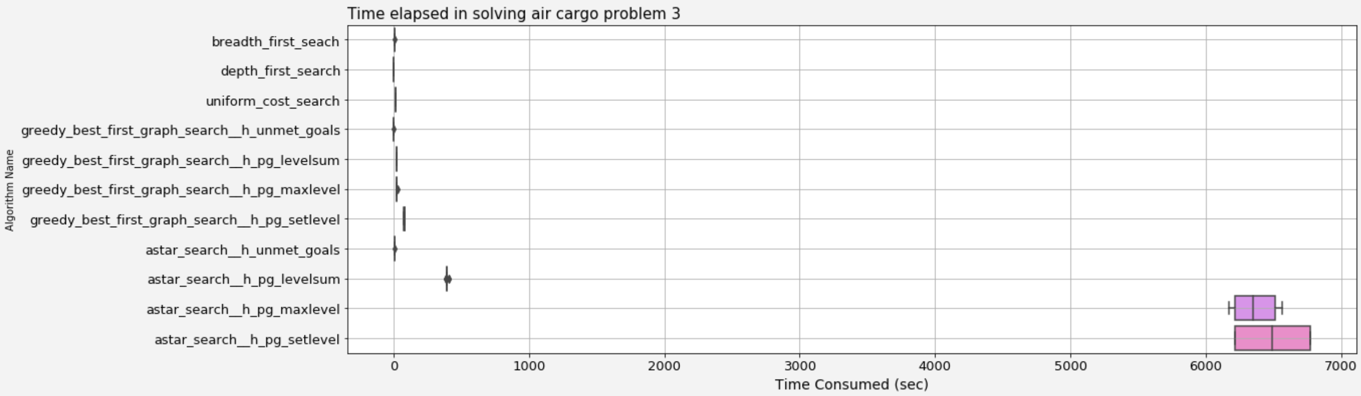
Graph1. Number of action vs Algorithm used



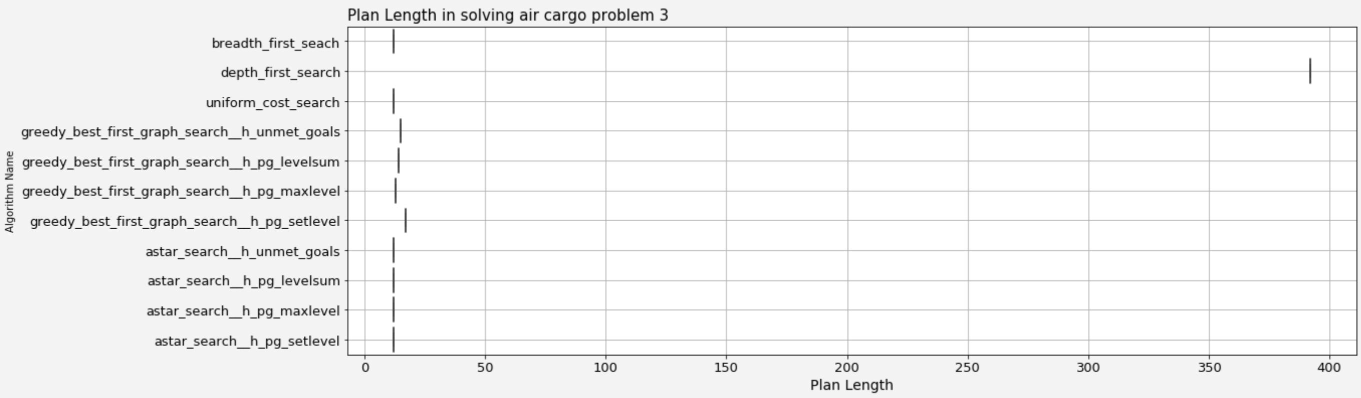
Graph2. Number of nodes vs Algorithm used



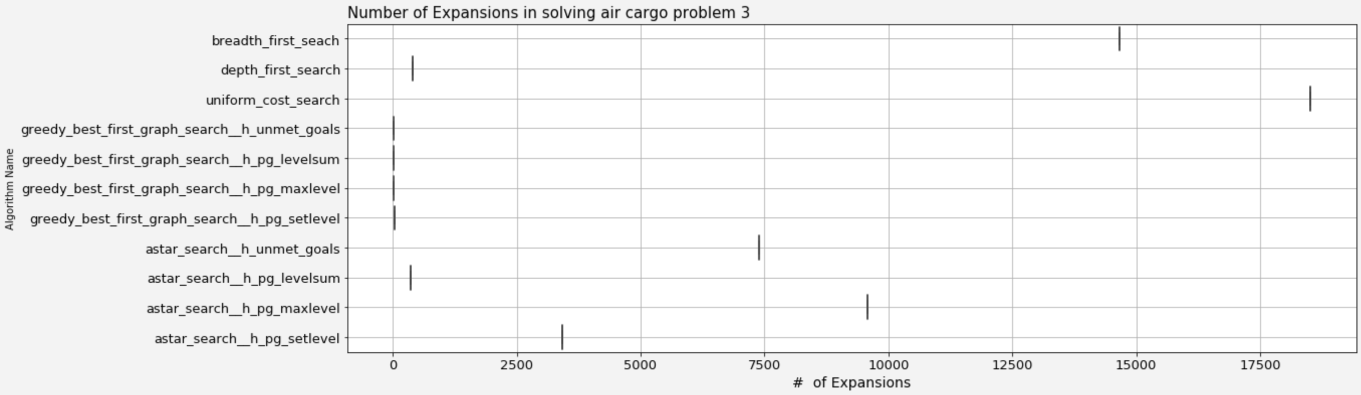
Graph3. Time Consumed (sec) vs Algorithm used



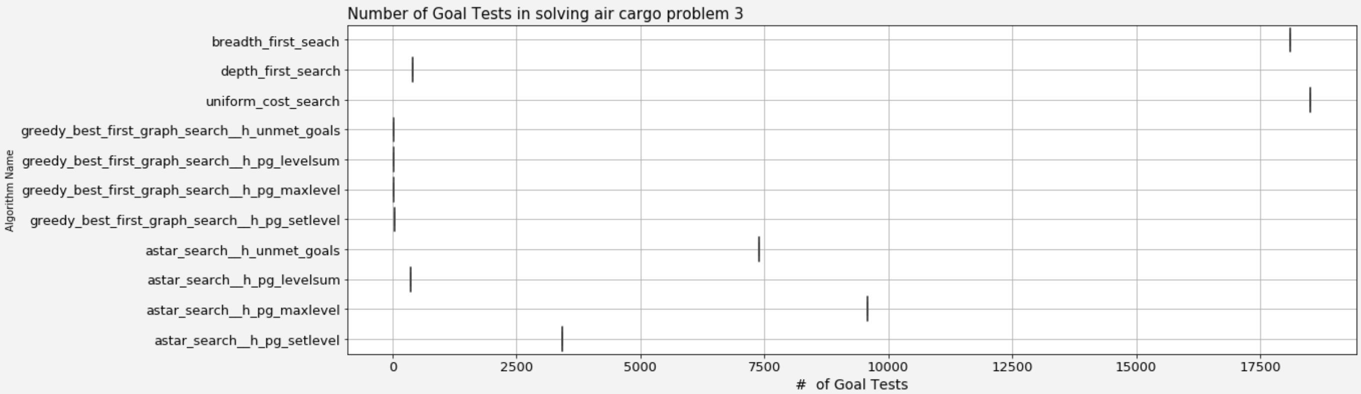
Graph4. Plan Length vs Algorithm used



Graph5. Number of Expansions vs Algorithm used



Graph6. Number of Goal tests vs Algorithm used



Tabulation

Algorithm Name	Actions	Expansions	Goal Tests	New Nodes	Plan Length	Avg Elap.Time
breadth_first_seach	88	14663	18098	129625	12	10.202076
depth_first_search	88	408	409	3364	392	1.168774
uniform_cost_search	88	18510	18512	161936	12	13.964497
greedy_best_first_graph_search__h_unmet_goals	88	25	27	230	15	0.045546
greedy_best_first_graph_search__h_pg_levelsum	88	14	16	126	14	20.98265
greedy_best_first_graph_search__h_pg_maxlevel	88	21	23	195	13	26.511476
greedy_best_first_graph_search__h_pg_setlevel	88	35	37	345	17	79.100955
astar_search__h_unmet_goals	88	7388	7390	65711	12	8.048166
astar_search__h_pg_levelsum	88	369	371	3403	12	395.114345
astar_search__h_pg_maxlevel	88	9580	9582	86312	12	6358.946978
astar_search__h_pg_setlevel	88	3423	3425	31596	12	6493.967158

Comparison of Strategies

Between Uninformed Search Strategies

When comparing uninformed search strategies based on the above plots and table, it can be seen that among the strategies which are able to find the optimal solution ‘Breadth First Search’ strategy is performing best in terms of speed i.e. time consumption and also based on memory utilized i.e. data storage. Among the strategies which found the optimal solution ‘Uniform Cost Search’ lags in memory utilized and time consumption. Additionally, ‘Depth First Search’ is performing the best in terms of memory and time utilization but is performing worst in terms of finding optimal solution.

Between Informed Search Strategies

When comparing informed search strategies with multiple heuristics based on the above plots and table, it can be seen that only ‘A* Search’ strategies are able to find optimal solution while ‘Greedy

Best First Graph Search' isn't. Though on basis time and memory consumption 'Greedy Best First Graph Search' strategies are performing better than 'A* Search' strategies but it isn't able to find an optimal solution.

Between Heuristics in Informed Search Strategies

Only 'A* Search' strategies with every heuristic, namely 'UnmetGoals', 'LevelSum', 'MaxLevel', and 'SetLevel' is able to find the optimal solution. Though in both strategies heuristic 'UnmetGoals' is seeming to be performing best in terms of time consumption and moderately bad in terms of memory consumption. Heuristic 'LevelSum' is performing the best in terms of memory consumption, while also finding the optimal solution with 'Greedy Best First Graph Search' strategy but this heuristic take more time compared to 'UnmetGoals' heuristic. While heuristic 'SetLevel' in comparison to other heuristics is seeming to perform worst in terms of memory and time consumption.

In regards to informed search in this problem, 'A* Search' with 'UnmetGoals' heuristic is able to find optimal solution while consuming least amount of time with a bit larger memory consumption. While 'A* Search' with 'LevelSum' heuristics is also able to find optimal solution and this while having least memory consumption with somewhat larger time consumption. So based on the need for storage and runtime, the decision for the best heuristics could be different.

Between Uninformed and Informed Search Strategies

In this problem, 'A* Search' with 'UnmetGoals' heuristic was able to find the optimal solution earlier than any other informed or uninformed search strategies. Additionally, 'A* Search' with 'LevelSum' heuristic is also able to find the optimal solution and it achieves this with relatively much lower memory consumption but taking longer time. In uninformed strategies 'Breadth First Search' and 'Uniform Cost Search' are also able to find the optimal solution but they do this larger time and also utilise large amount of memory in achieving this feat. So based on the need for storage and runtime, the decision for the best heuristics could be different.

<u>Best Strategy</u>

Summarizing the results in the table below

Algorithm Name	Memory Consumption	Optimal Solution	Time Consumption
breadth_first_seach		BU, OB	BU
depth_first_search	BU		
uniform_cost_search		BU, OB	
greedy_best_first_graph_search__h_unmet_goals			BI, OB
greedy_best_first_graph_search__h_pg_levelsum	BI, OB		
greedy_best_first_graph_search__h_pg_maxlevel			
greedy_best_first_graph_search__h_pg_setlevel			
astar_search__h_unmet_goals		BI, OB	
astar_search__h_pg_levelsum		BI, OB	
astar_search__h_pg_maxlevel		BI, OB	
astar_search__h_pg_setlevel		BI, OB	
OB: Overall Best, COB: Close to Overall Best, BU: Best in Uninformed Search, CBU: Close to Best in Uninformed Search, BI: Best in Informed Search, CBI: Close to Best in Informed Search.			

Assuming finding optimal solution is a must, the best strategy to use in the following setting

- In least memory possible with no bound on time consumed:
  'A* Search' with 'LevelSum' heuristic
- Fastest possible response with no constraint on memory:

‘A* Search’ with ‘UnmetGoals’ heuristic

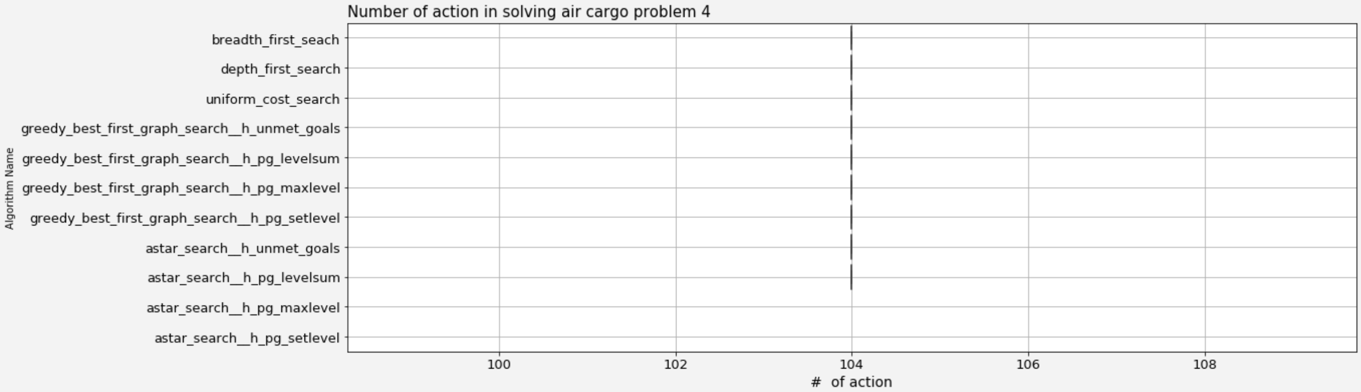
- Best balance between time and memory consumed:

  Both Time and Memory difference is sufficient to approximate this. Hence, based on need.

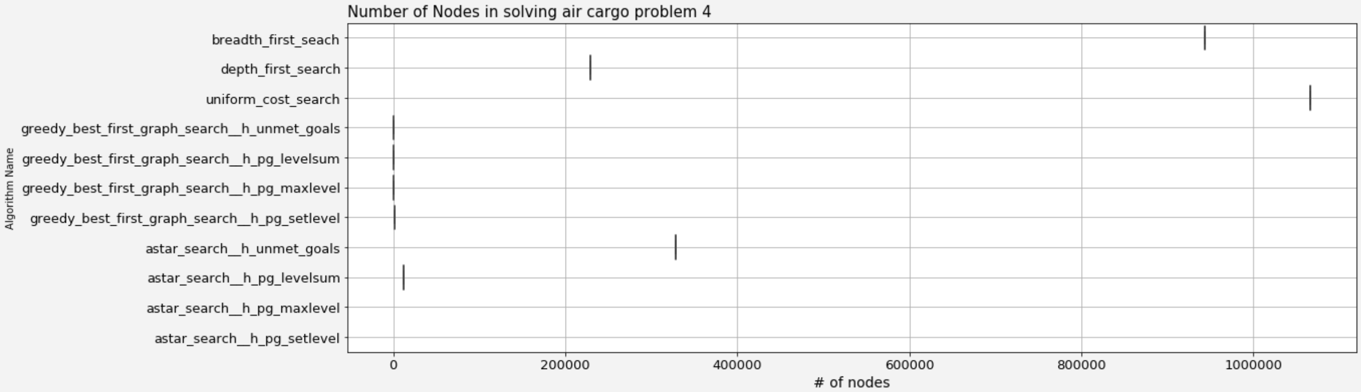
2.1.4. Problem 4:

Visualization

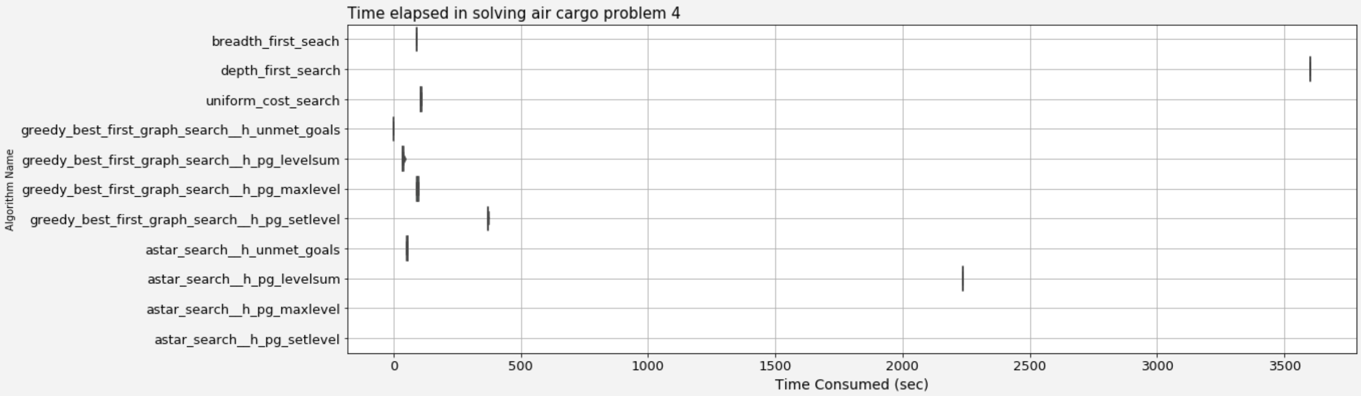
Graph1. Number of action vs Algorithm used



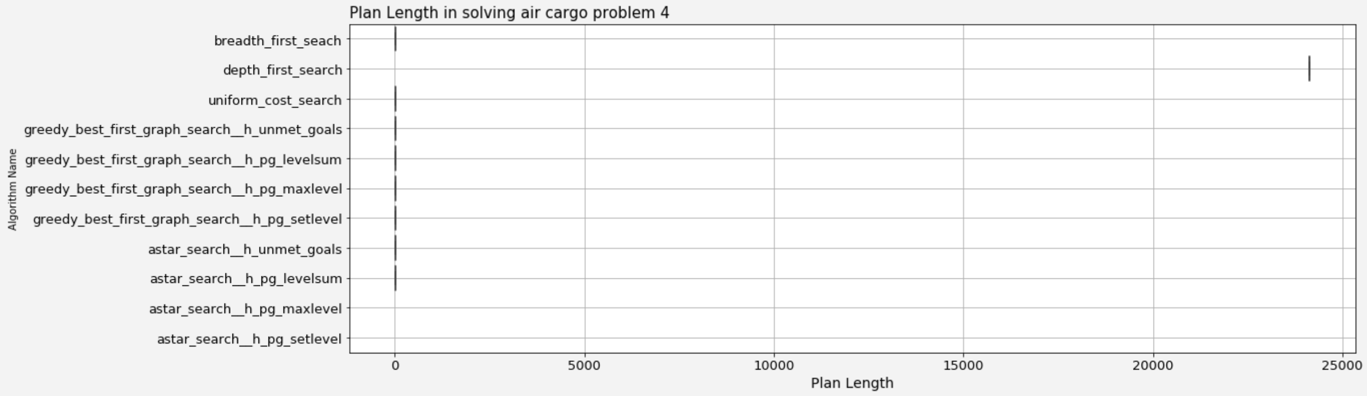
Graph2. Number of nodes vs Algorithm used



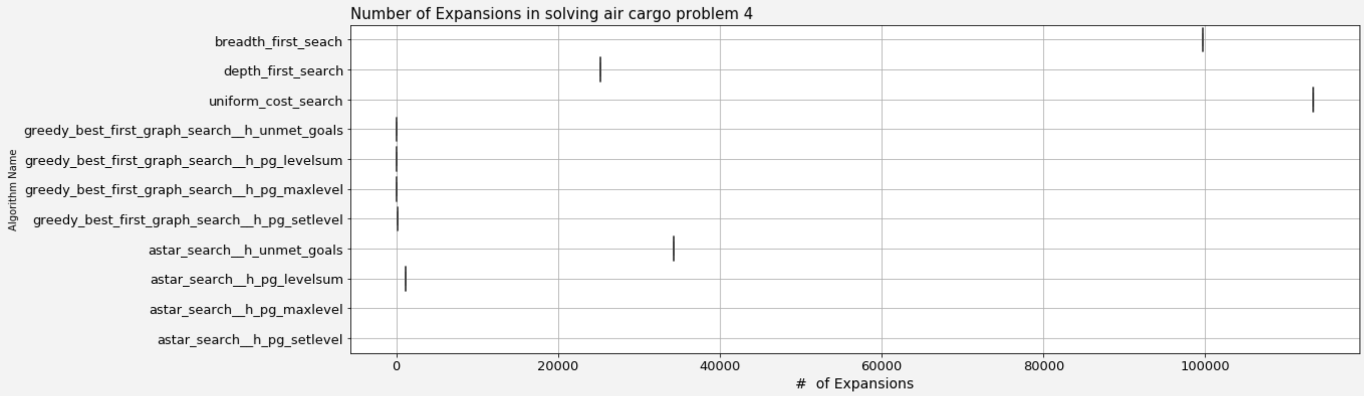
Graph3. Time Consumed (sec) vs Algorithm used



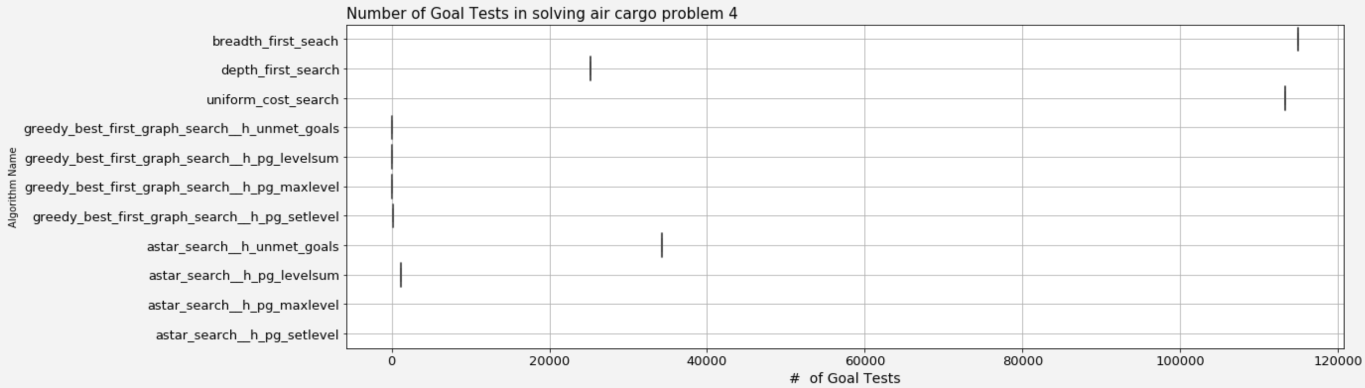
Graph4. Plan Length vs Algorithm used



Graph5. Number of Expansions vs Algorithm used



Graph6. Number of Goal tests vs Algorithm used



Tabulation

Algorithm Name	Actions	Expansions	Goal Tests	New Nodes	Plan Length	Avg Elap.Time
breadth_first_seach	104	99736	114953	944130	14	93.138974
depth_first_search	104	25174	25175	228849	24132	3602.291031
uniform_cost_search	104	113339	113341	1066413	14	109.207186
greedy_best_first_graph_search__h_unmet_goals	104	29	31	280	18	0.053888
greedy_best_first_graph_search__h_pg_levelsum	104	17	19	165	17	37.933342
greedy_best_first_graph_search__h_pg_maxlevel	104	56	58	580	17	95.145388
greedy_best_first_graph_search__h_pg_setlevel	104	107	109	1164	23	372.422547
astar_search__h_unmet_goals	104	34330	34332	328509	14	53.627757
astar_search__h_pg_levelsum	104	1208	1210	12210	15	2238.20014
astar_search__h_pg_maxlevel	NA	NA	NA	NA	NA	NA
astar_search__h_pg_setlevel	NA	NA	NA	NA	NA	NA

** NA represents the run that took longer than an 2 hours and was aborted.

Comparison of Strategies

Between Uninformed Search Strategies

When comparing uninformed search strategies based on the above plots and table, it can be seen that among the strategies which are able to find the optimal solution 'Breadth First Search' strategy is performing best in terms of speed i.e. time consumption and also based on memory utilized i.e. data storage. Among the strategies which found the optimal solution 'Uniform Cost Search' lags in memory utilized and time consumption. Additionally, 'Depth First Search' is performing the best in terms of memory and time utilization but is performing worst in terms of finding optimal solution.

Between Informed Search Strategies

When comparing informed search strategies with multiple heuristics based on the above plots and table, it can be seen that only 'A* Search' strategies are able to find optimal solution while 'Greedy Best First Graph Search' isn't. Though on the basis time and memory consumption 'Greedy Best First Graph Search' strategies are performing way much better than 'A* Search' strategies but it isn't able to find an optimal solution.

Between Heuristics in Informed Search Strategies

Only 'A* Search' with 'UnmetGoals' heuristic is able to find the optimal solution and we don't have any knowledge on 'MaxLevel' and 'SetLevel' heuristics with 'A* Search' strategy as their execution time exceeded 2 hours, hence these execution were stopped. Though in both strategies heuristic 'UnmetGoals' is seeming to be performing best in terms of time consumption and moderately bad in terms of memory consumption. Heuristic 'LevelSum' is performing the best in terms of memory consumption but this heuristic take more time compared to 'UnmetGoals' heuristic. While heuristic 'SetLevel' in comparison to other heuristics is seeming to perform worst in terms of memory and time consumption at least with 'Greedy Best First Graph Search' strategy.
In regards to informed search in this problem, 'A* Search' with 'UnmetGoals' heuristic is able to find optimal solution while consuming least amount of time with a bit larger memory consumption.

Between Uninformed and Informed Search Strategies

In this problem, among the informed search only 'A* Search' with 'UnmetGoals' heuristic was able to find the optimal solution. In uninformed strategies 'Breadth First Search' and 'Uniform Cost Search' are also able to find the optimal solution but they do this larger time and also utilise large amount of memory in achieving this feat. So based on this the only strategy we can take to find the optimal solution is 'A* Search' with 'UnmetGoals' heuristic.

Best Strategy

Summarizing the results in the table below

Algorithm Name	Memory Consumption	Optimal Solution	Time Consumption
breadth_first_seach		BU, OB	BU
depth_first_search	BU		
uniform_cost_search		BU, OB	
greedy_best_first_graph_search__h_unmet_goals			BI, OB
greedy_best_first_graph_search__h_pg_levelsum	BI, OB		
greedy_best_first_graph_search__h_pg_maxlevel			
greedy_best_first_graph_search__h_pg_setlevel			
astar_search__h_unmet_goals		BI, OB	
astar_search__h_pg_levelsum			
astar_search__h_pg_maxlevel	NA	NA	NA
astar_search__h_pg_setlevel	NA	NA	NA

OB: Overall Best, COB: Close to Overall Best, BU: Best in Uninformed Search, CBU: Close to Best in Uninformed Search, BI: Best in Informed Search, CBI: Close to Best in Informed Search.

** NA represents the run that took longer than an 2 hours and was aborted.

Assuming finding optimal solution is a must, the best strategy to use in the following setting

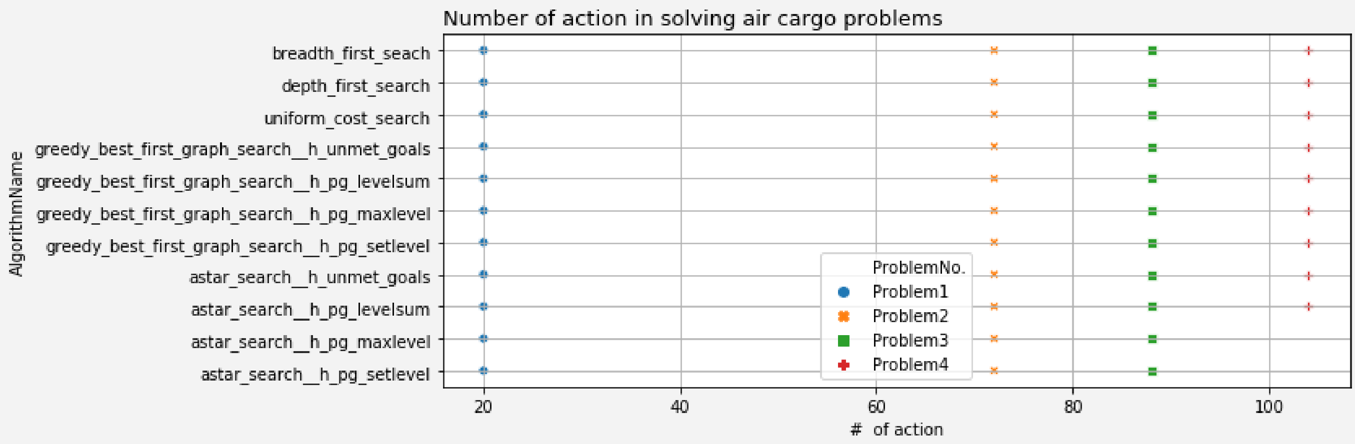
- In least memory possible with no bound on time consumed:
  'A* Search' with 'UnmetGoals' heuristic
- Fastest possible response with no constraint on memory:
  'A* Search' with 'UnmetGoals' heuristic
- Best balance between time and memory consumed:
  'A* Search' with 'UnmetGoals' heuristic

2.2. Overall Results

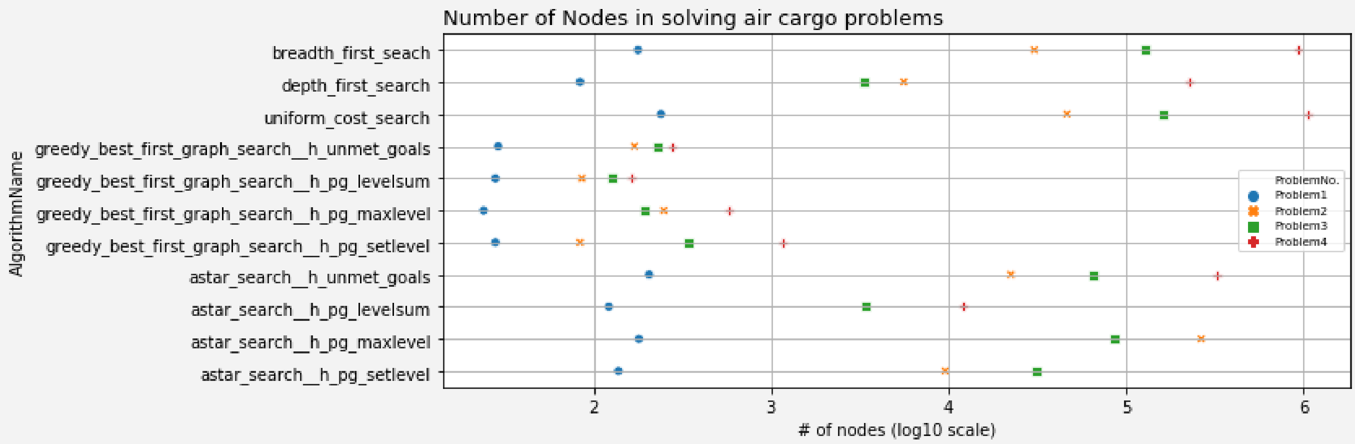
2.2.1. Problem 1:

Visualization

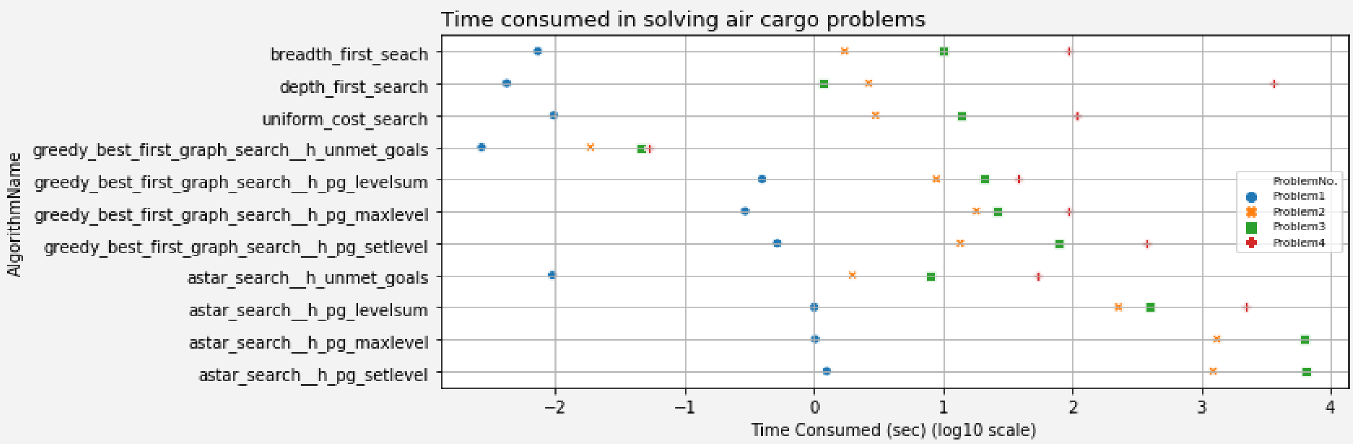
Graph1. Number of action vs Algorithm used



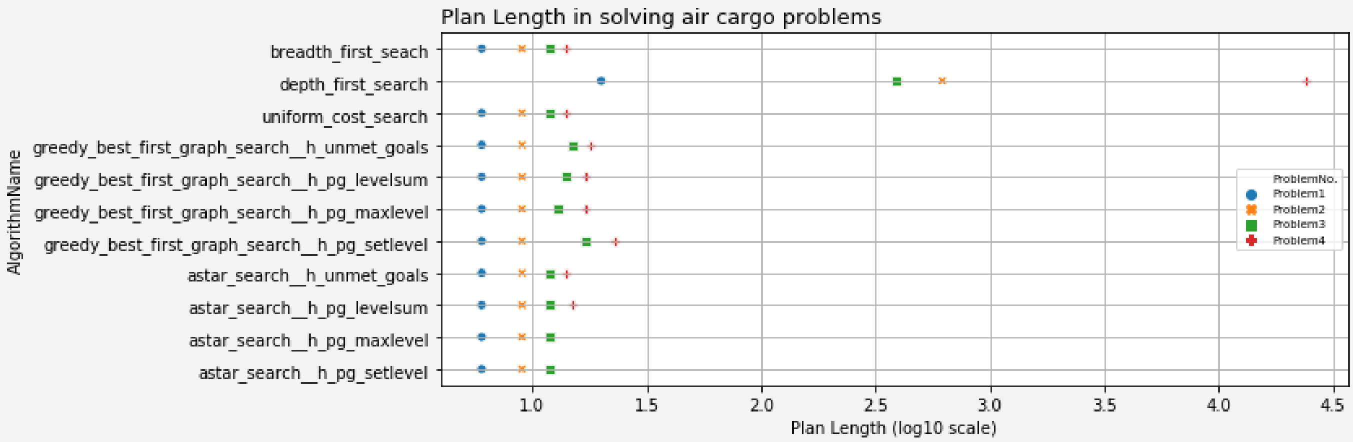
Graph2. Log10(Number of nodes) vs Algorithm used



Graph3. Log10(Time Consumed (sec)) vs Algorithm used



Graph4. Log10(Plan Length) vs Algorithm used



Note: Variables value is not available with ‘astar_search__h_pg_maxlevel’ and ‘‘astar_search__h_pg_setlevel’ with problem 4.

<u>Understanding Strategies behaviour as Problem size Increases</u>

As the size of problem increases, These were the following observation related to

Number of Nodes:

> ‘Greedy Best First Graph Search’ strategy seems to be most optimised strategy with almost every heuristic present here, followed by ‘A* Search’ strategy with every heuristic present here and this closely followed by uninformed search strategies. Uninformed search strategies have lesser nodes comparatively to ‘A* Search’ when smaller problems are considered but this reverses as the problem size increases.

Time Consumed:

> Comparatively ‘Greedy Best First Graph Search’ strategy is performing the best in regards to time consumed as the problem size increases. Also ‘Greedy Best First Graph Search’ with ‘UnmetGoals’ heuristic performance is the best in terms of speed. Uninformed search algorithm are also able to have somewhat balance time consumption as the problem size increases. But with ‘A* Search’ strategy only ‘UnmetGoals’ heuristic comparatively seems to have reasonable time consumption, and rest all heuristic with this strategy perform worse than any other strategy.

Path Length:

> Among all algorithm that are present ‘Depth First Search’ performs the worst and it has never been able to find the optimal path. Except from this rest uninformed search namely ‘Breadth First Search’ and ‘Uniform Cost Search’ have been able to find the optimal path with all the problems. Among informed search ‘Greedy Best First Graph Search’ strategy is not able to find the optimal solution as the problem size increases, though it takes less amount of time. While ‘A* Search’ strategy is

able to find the optimal path in all these problem with 'UnmetGoals' heuristic, but isn't able to do so in problem 4 with other heuristics.

Conclusion

Among all strategies 'Depth First Search' strategy is performing worst, as it is unable to find optimal path in problem of any size while it also consume large amount of time. Apart from this uninformed strategy, 'Breadth First Search' and 'Uniform Cost Search' have been able to find the the optimal path in every problem, while comparatively consuming moderate time but they consumed large amount of memory. 'Greedy Best First Graph Search' with 'UnmetGoals' heuristic strategy performed the best in terms of consuming the least amount of time and memory but it wasn't able to find the optimal solution as the problem size increased. Other heuristics with 'Greedy Best First Graph Search' strategy did consume less memory and comparatively moderate amount of time, but they were unable to find optimal solution as problem size became bigger. A* Search' with 'UnmetGoals' heuristic strategy performed the best as the problem size became bigger in terms of consuming the least amount of time and finding the optimal solution, and still it consumed less memory compared to uninformed search strategies. Other heuristics with 'A* Search' didn't perform well with bigger problem, neither they were able to find the optimal solution nor they consumed lesser time, though memory consumption wise they were good compared to uninformed search strategies.

These results have shown the benefits of using informed search strategies with custom heuristics over uninformed search techniques. These benefits are significant both in terms of time and memory consumption. Another, benefit is that one can customize the trade-off between speed and memory usage by using different custom heuristics and this benefit is not possible with uninformed search strategies.

Part 3. Formally Answering the Questions

Question 1.

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?

Answer

'Greedy Best First Graph Search' with 'UnmetGoals' heuristic strategy, as it has the best performance in terms of response time, while Greedy Best First Graph Search' consumes least amount of memory compared to any other search strategies present in this discussion and doing so while finding an optimal solution in a very restricted domain.

Question 2.

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)

Answer

'A* Search' with 'UnmetGoals' heuristic strategy, as it is the fastest and gives an optimal solution, while consuming less memory compared to 'Breadth First Search' and 'Uniform Cost Search'. 'Breadth First Search' and 'Uniform Cost Search' are also able to find optimal solution but they consume large time and memory

Question 3.

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

Answer

'Breadth First Search', as it performs faster and utilizes less memory than other search strategies. As 'Breadth First Search' is a complete and optimal search strategy with only downside being large memory usage, if the problem's branching factor is high.

References

- Artificial Intelligence: A Modern Approach, 3rd Ed, S. Russel and P. Norvig (2010)
- Course videos

END