Planning Graph Project Algorithm Analysis

This research aims to show which algorithms work best for finding the optimal plan to a logic based Planning and Search problem domain.

Table of Contents

Optimal Plans

The below optimal plans are the result of the solution determined as the most optimal in the conclusion, A* search with the Ignore Preconditions Heuristic.

Problem 1

$Load(C1, P1, SFO)$
$Fly(P1, SFO, JFK)$
$Unload(C1, P1, JFK)$
$Load(C2, P2, JFK)$
$Fly(P2, JFK, SFO)$
$Unload(C2, P2, SFO)$

Problem 2

$Load(C1, P1, SFO)$
$Fly(P1, SFO, JFK)$
$Unload(C1, P1, JFK)$
$Load(C2, P2, JFK)$
$Fly(P2, JFK, SFO)$
$Unload(C2, P2, SFO)$
$Load(C3, P3, ATL)$
$Fly(P3, ATL, SFO)$
$Unload(C3, P3, SFO)$

Problem 3

$Load(C1, P1, SFO)$
$Fly(P1, SFO, ATL)$
$Load(C3, P1, ATL)$
$Fly(P1, ATL, JFK)$
$Unload(C1, P1, JFK)$
$Load(C2, P2, JFK)$
$Fly(P2, JFK, ORD)$
$Load(C4, P2, ORD)$
$Fly(P2, ORD, SFO)$
$Unload(C2, P2, SFO)$
$Unload(C3, P1, JFK)$
$Unload(C4, P2, SFO)$

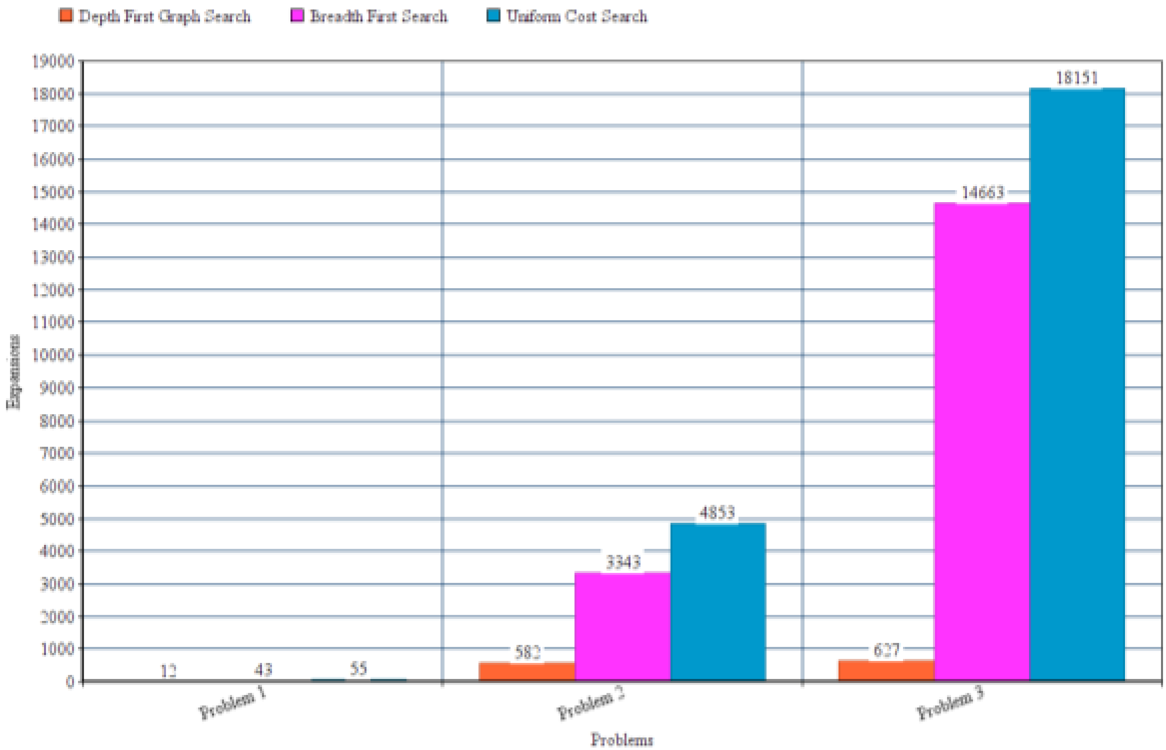
Non-Heuristic Search results

Table of Metric Results for each Problem using Non-Heuristic Search Algorithms

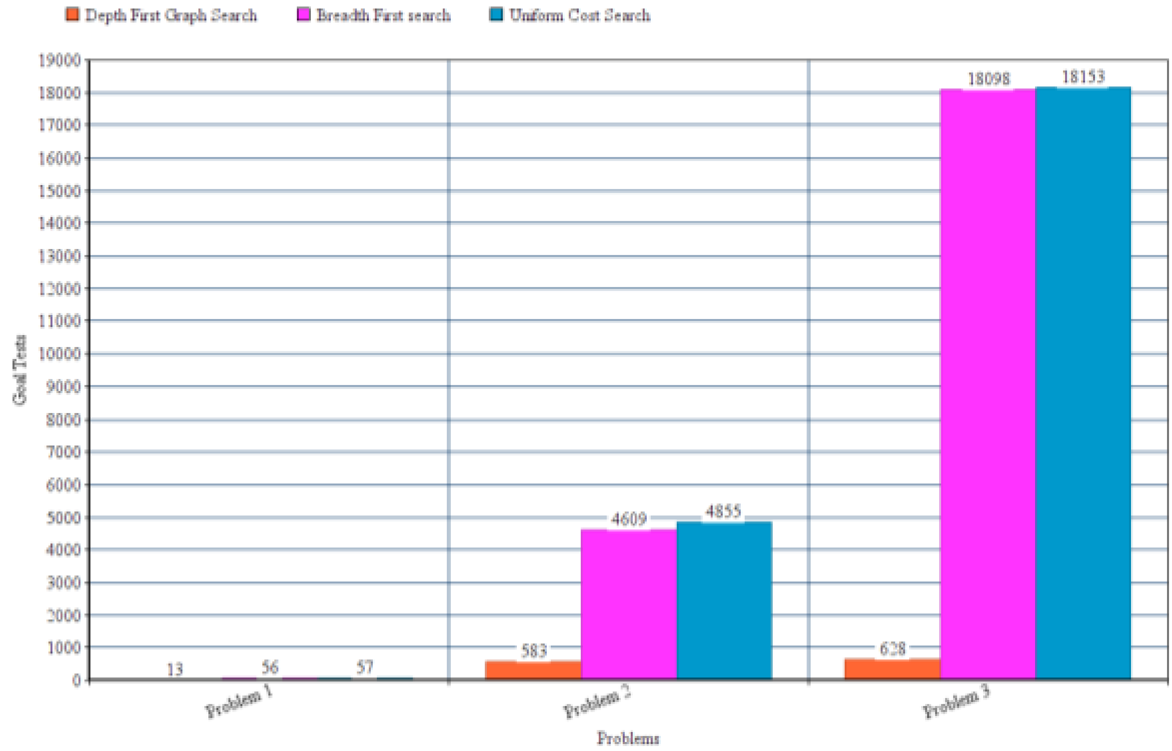
Problem	Test	Metric	Depth First Graph Search	Breadth First Search	Uniform Cost Search
Problem 1	Test 1	Expansions	12	43	55
		Goal Tests	13	56	57
		New Nodes	48	180	224
		Execution Time (s)	0,01	0,05	0,06
		Plan Length	12	6	6
	Test 2	Expansions	12	43	55
		Goal Tests	13	46	57
		New Nodes	48	180	224
		Execution Time (s)	0,01	0,04	0,06
		Plan Length	12	6	6
	Test 3	Expansions	12	43	55
		Goal Tests	13	56	57
		New Nodes	48	180	224
		Execution Time (s)	0.01	0.05	0.06
		Plan Length	12	6	6
Problem 2	Test 1	Expansions	582	3343	4853
		Goal Tests	583	4609	4855
		New Nodes	5211	30509	44041
		Execution Time (s)	3,72	17,17	57,42
		Plan Length	575	9	9
	Test 2	Expansions	582	3343	4853
		Goal Tests	583	4609	4855
		New Nodes	5211	30509	44041
		Execution Time (s)	3,77	16,99	56,60
		Plan Length	575	9	9
	Test 3	Expansions	582	3343	4853
		Goal Tests	583	4609	4855
		New Nodes	5211	30509	44041
		Execution Time (s)	3,72	17,05	62,14
		Plan Length	575	9	9
Problem 3	Test 1	Expansions	627	14663	18151
		Goal Tests	628	18098	18153

		New Nodes	5176	129631	159038
		Execution Time (s)	4,06	125,91	488,89
		Plan Length	596	12	12
	Test 2	Expansions	627	14663	18151
		Goal Tests	628	18098	18153
		New Nodes	5176	129631	159038
		Execution Time (s)	3,99	123,32	506,05
		Plan Length	596	12	12
	Test 3	Expansions	627	14663	18151
		Goal Tests	628	18098	18153
		New Nodes	5176	129631	159038
		Execution Time (s)	4,03	122,78	493,50
		Plan Length	596	12	12

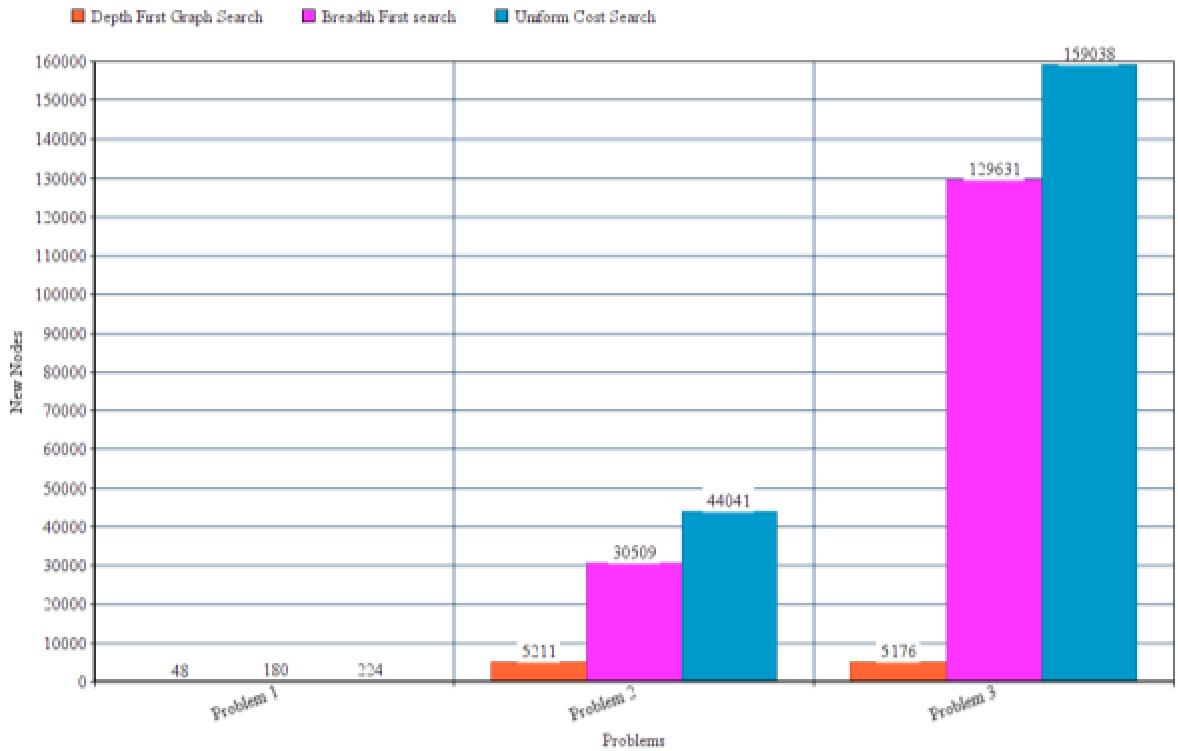
Graph of Node Expansion for each Problem using Non-Heuristic Search Algorithms



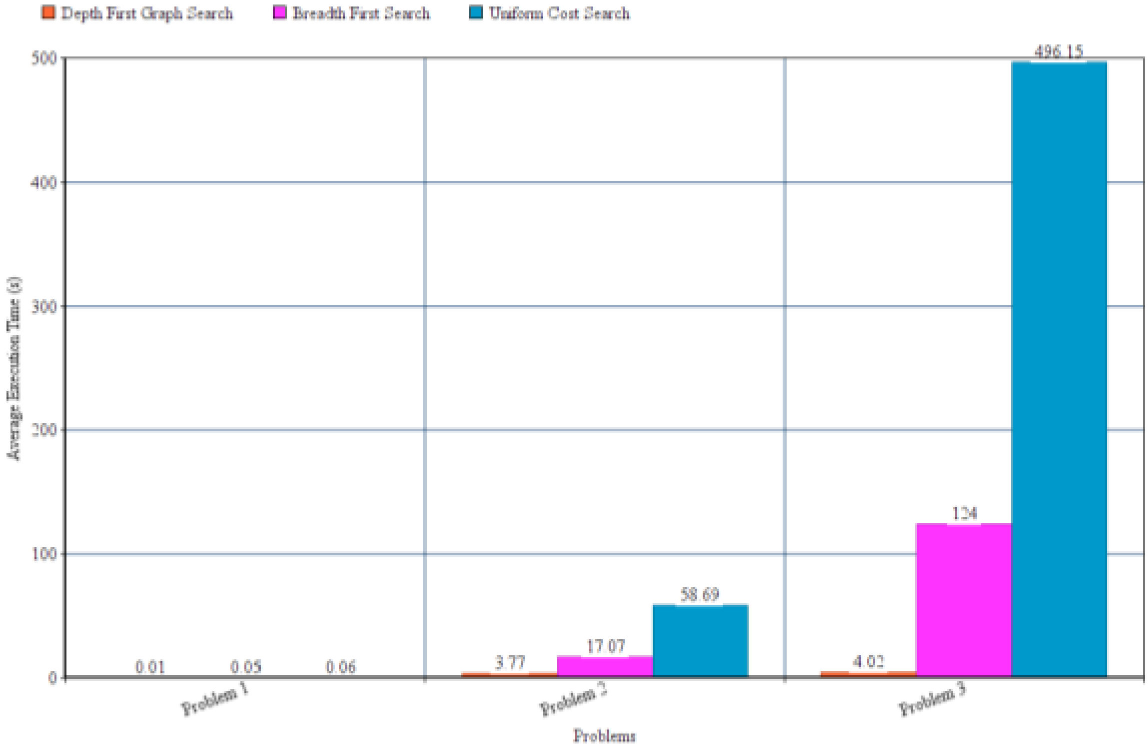
Graph of Goal Tests for each Problem using Non-Heuristic Search Algorithms



Graph of New Nodes for each Problem using Non-Heuristic Search Algorithms



Graph of Average Execution Time in Seconds for each Problem using Non-Heuristic Search Algorithms



Non-Heuristic Search Analysis

The table of results and graphs above clearly show that Depth First Graph Search expands the least nodes, goal tests the least nodes, creates the least nodes and executes far quicker than either Breadth First Search or Uniform Cost Search by a statistically significant value. However, from the above table of results and the **Graph of Plan Length for Each Problem using Both Heuristic and Non-Heuristic Search Algorithms**, it is evident that the plan produced by Depth First Graph Search is infeasibly long. The number of steps produced is not at all optimal to result in the goal states required. All other algorithms consistently produce more optimal paths. Thus, Depth First Graph Search is not a feasible solution to a logic based Planning or Search problem domain.

Breadth First Search and Uniform Cost Search produce the same plan length for all problems. A much lower and feasible plan length compared to Depth First Graph Search. Breadth First Search performs consistently better than Uniform Cost Search. Breadth First Search expands less nodes, goal tests less nodes and creates less nodes than Uniform Cost Search by a statistically significant amount. Breadth First Search also executes significantly quicker than Uniform Cost Search. From this it is clear that, for non-heuristic based search algorithms, Breadth First Search produces the most optimal path, in the quickest time and utilizes the least resources.

Non-Heuristic Conclusion

While Depth First Graph Search uses the least resources, the Plan length is completely infeasible, therefore Depth First Graph Search cannot be considered as a viable solution for logic based Planning and Search problem domains.

Breadth First Search consistently uses less resources, and executes statistically significantly quicker than Uniform Cost Search while producing the same plan length. It can be confidently concluded that Breadth First Search is the best solution - of the three chosen - for Non-Heuristic based Search algorithms in a logic based Planning and Search problem domain.

This is expected, as each action/node have the same cost, and Breadth First Search is known to be optimal in this case. [1]

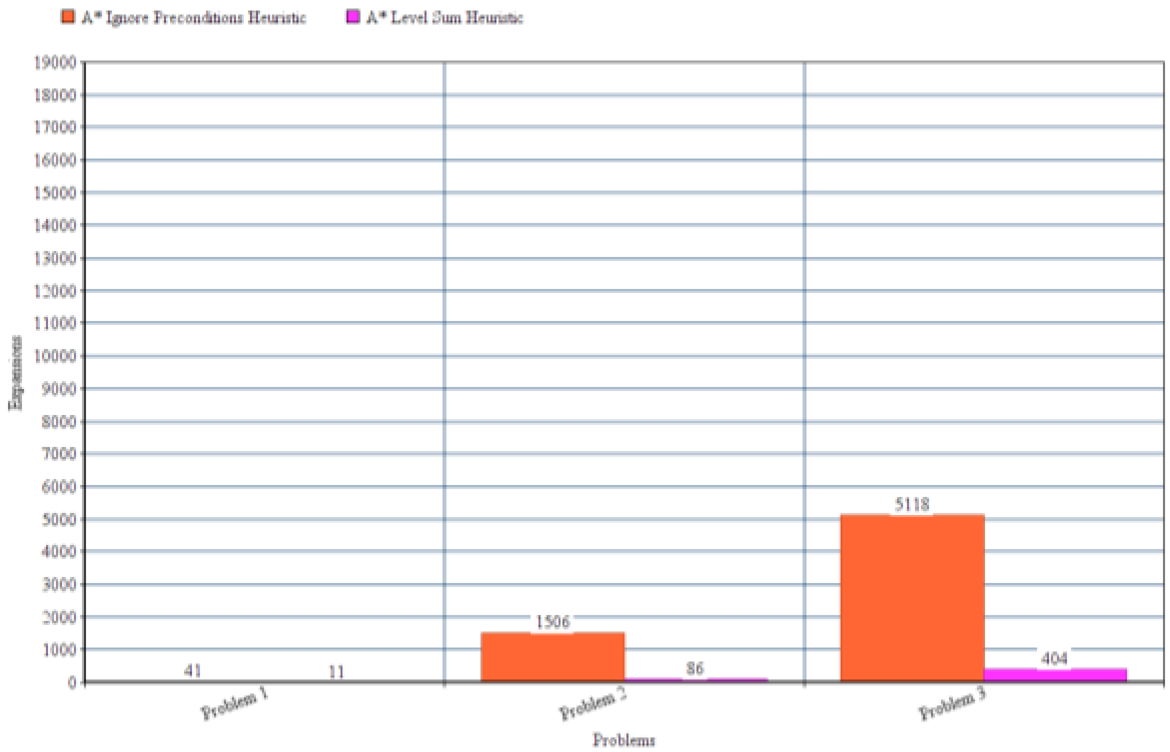
Heuristic Search results

Table of Metric Results for each Problem using Heuristic Search Algorithms

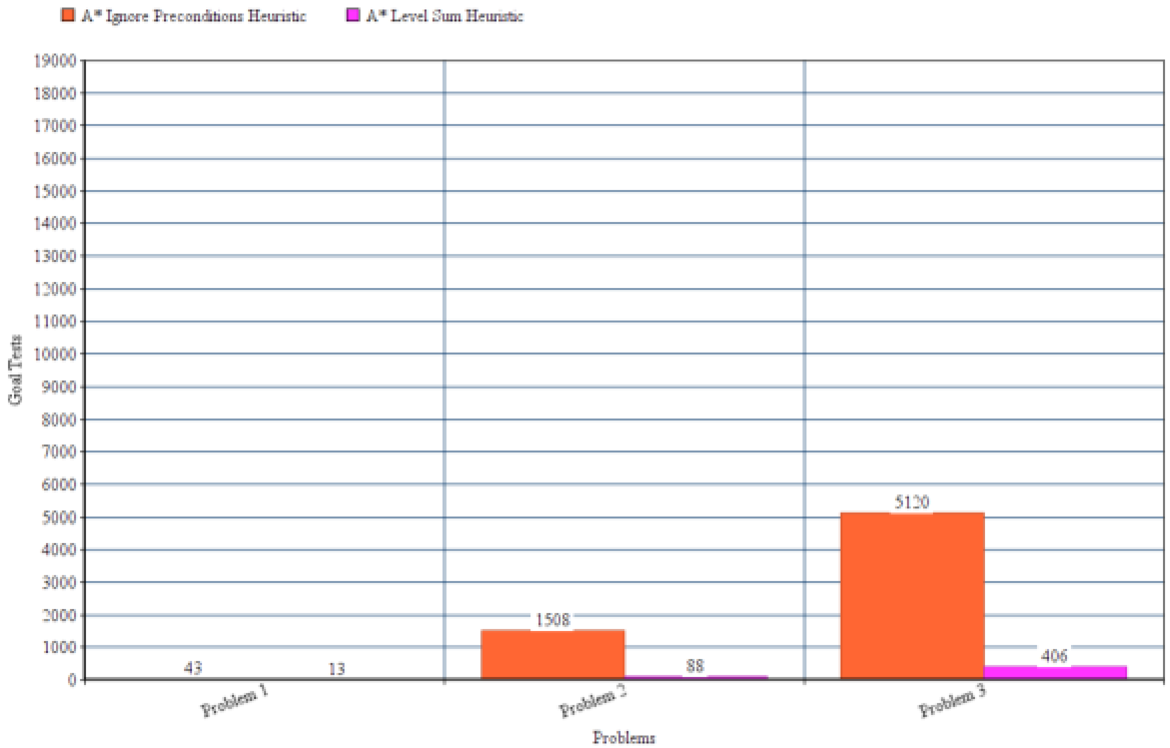
Problem	Test	Metric	A* Ignore Precondition Heuristic	A* Level Sum Heuristic
Problem 1	Test 1	Expansions	41	11
		Goal Tests	43	13
		New Nodes	170	50
		Execution Time (s)	0,05	1,98
		Plan Length	6	6
	Test 2	Expansions	41	11
		Goal Tests	43	13
		New Nodes	170	50
		Execution Time (s)	0,05	1,93
		Plan Length	6	6
	Test 3	Expansions	41	11
		Goal Tests	43	13
		New Nodes	170	50
		Execution Time (s)	0,04	1,95
		Plan Length	6	6
Problem 2	Test 1	Expansions	1506	86
		Goal Tests	1508	88
		New Nodes	13820	841
		Execution Time (s)	16,08	208,02
		Plan Length	9	9
	Test 2	Expansions	1506	86

		Goal Tests	1508	88
		New Nodes	13820	841
		Execution Time (s)	16,72	194,49
		Plan Length	9	9
	Test 3	Expansions	1506	86
		Goal Tests	1508	88
		New Nodes	13820	841
		Execution Time (s)	16,69	200,16
		Plan Length	9	9
Problem 3	Test 1	Expansions	5118	404
		Goal Tests	5120	406
		New Nodes	45650	3718
		Execution Time (s)	101,44	1396,69
		Plan Length	12	12
	Test 2	Expansions	5118	404
		Goal Tests	5120	406
		New Nodes	45650	3718
		Execution Time (s)	120,17	1292,05
		Plan Length	12	12
	Test 3	Expansions	5118	404
		Goal Tests	5120	406
		New Nodes	45650	3718
		Execution Time (s)	116,03	1337,49
		Plan Length	12	12

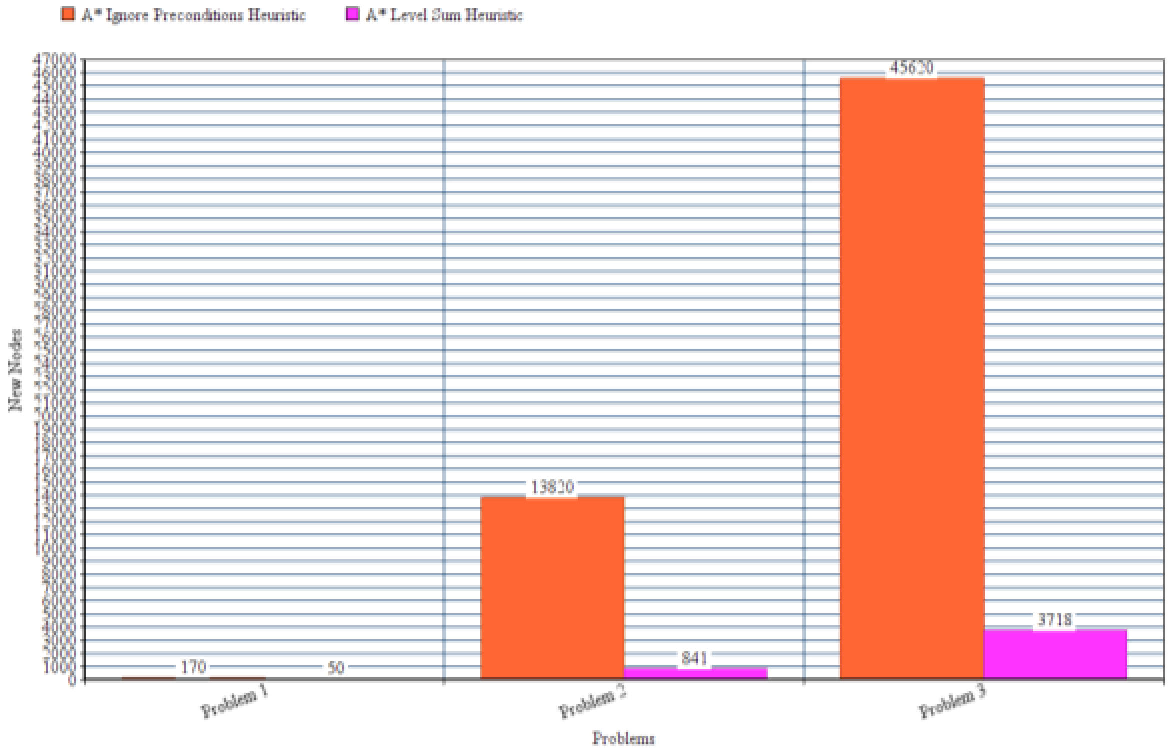
Graph of Node Expansion for each Problem using Heuristic Search Algorithms



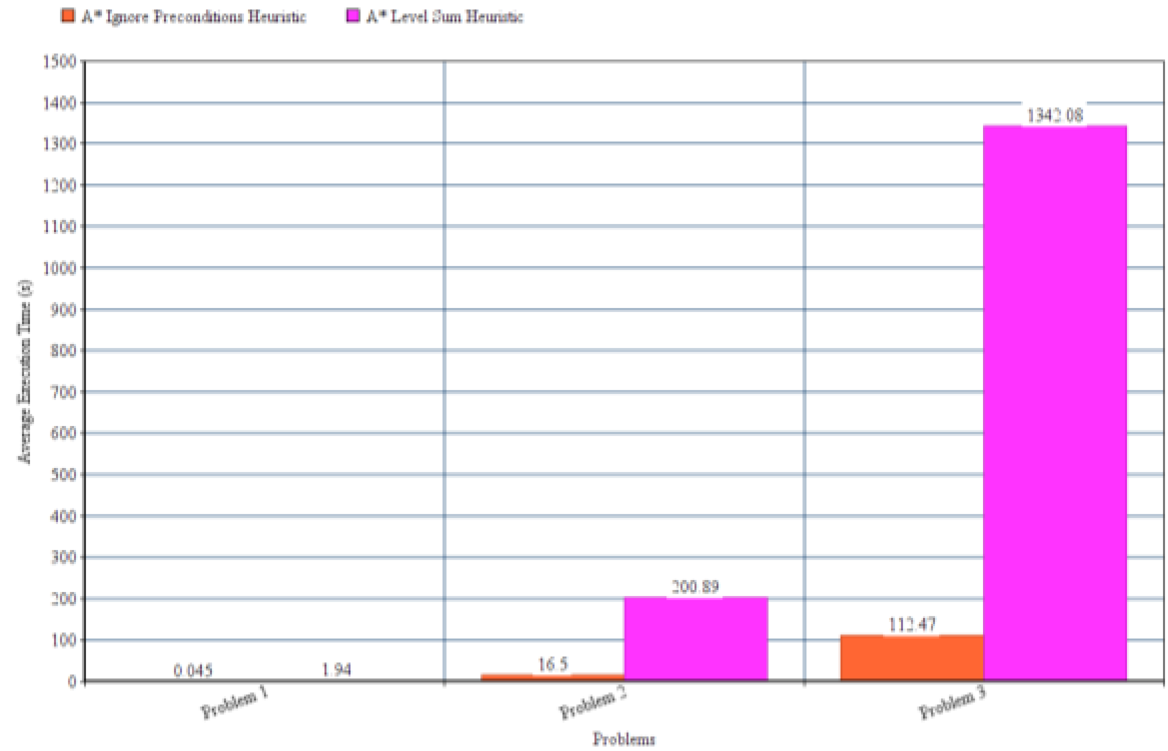
Graph of Goal Tests for each Problem using Heuristic Search Algorithms



Graph of New Nodes for each Problem using Heuristic Search Algorithms



Graph of Average Execution Time in Seconds for each Problem using Heuristic Search Algorithms



Heuristic Search Analysis

From the above table of results and the **Graph of Plan Length for Each Problem using Both Heuristic and Non-Heuristic Search Algorithms**, it is evident that A* search with the Level Sum Heuristic expands less nodes, goal tests less nodes and creates the least new nodes by a statistically significant difference when compared to A* search with the Ignore Preconditions Heuristic. However, the A* search with the Level Sum Heuristic takes far longer to provide a feasible path.

In such a case, the solution of choice would be decided by speed against resources. A* search with the Level Sum Heuristic will use far less memory and perhaps even storage space (depending on whether the planning graph will be stored permanently). The A* search with the Ignore Preconditions Heuristic will execute far quicker. If the goal is to determine a plan within the quickest amount of time then A* search with the Ignore Preconditions Heuristic should be considered, but if memory and perhaps even storage efficiency is required then the A* search with the Level Sum Heuristic should be considered.

Heuristic Search Conclusion

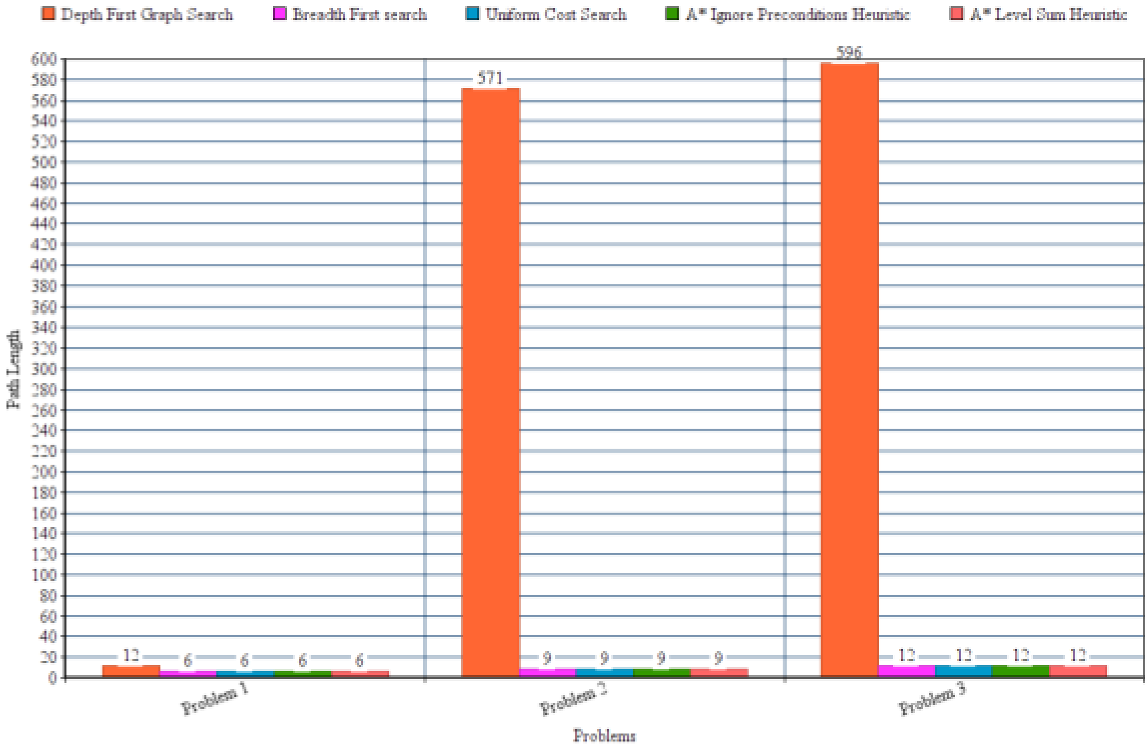
It is difficult to conclude which A* search heuristic is the better, as it greatly depends on the requirements of the solution. With that said, the likelihood of having to store the Planning Graph, and all its steps, is low. The plan itself would more likely be stored. This implies that A* search with the Ignore Preconditions Heuristic may be considered. Memory sizes constantly increase and prices decrease. Thus, using memory shouldn’t be a concern for large scale servers. However, with AI moving to small scale devices, such as drones, etc., memory should be used efficiently. This implies that large scale processing on adequate servers would benefit from A* search using the Ignore Preconditions Heuristic, while the small scale devices may benefit from A* search with the Level Sum Heuristic as they would require their resources be used efficiently.

In most cases, whether on small or large scale, a result or plan needs to be provided quickly. A person requiring GPS navigation cannot wait 23 minutes for a path to be found, nor can a drone. Ultimately, speed is favoured over efficient use of resources.

From the above, it can be concluded that speed is favoured over resource efficiency. Thus, A* search with Ignore Preconditions Heuristic can be considered the most optimal heuristic based search algorithm for solving logic based Planning and Search problem domains.

Heuristic against Non-Heuristic Based Search Algorithms Analysis

Graph of Plan Length for Each Problem using Both Heuristic and Non-Heuristic Search Algorithms



A* search with the Ignore Preconditions Heuristic was concluded as the optimal heuristic based search algorithm and Breadth First Search was concluded as the optional non-heuristic based search algorithm. Comparing the two optimal choices, A* search with the Ignore Preconditions heuristic expands significantly less nodes, goal tests significantly less nodes and creates significantly less nodes than Breadth First Search. A* search with the Ignore Preconditions heuristic executes faster than Breadth First Search for all problems on average, but not by a statistically significant value. The path lengths produced by both algorithms are the same. Therefore, mostly due to resource efficiency, A* search with the Ignore Preconditions Heuristic is the optimal choice.

The general results between Heuristic and Non-Heuristic based solutions show that non-heuristic based solutions execute faster than heuristic based solutions - except for A* search with the Ignore Preconditions Heuristic - but at a significantly higher resource cost. This suggests that, generally, heuristic based searches use less resources, however, the execution performance depends heavily on the heuristic implemented. This implies that implementing a good heuristic for the requirement will produce the optimal solution.

Heuristic against Non-Heuristic Based Search Algorithms Conclusion

As concluded before, Depth First Search provides an infeasible path, thus it cannot be considered as an optimal solution, despite its great performance in terms of time and efficiency in resources. Breadth First Search and Uniform Cost Search both produced feasible paths, however, Breadth First Search proved to be more efficient and faster than Uniform Cost Search. Thus, Breadth First Search is analyzed against the more optimal Heuristic based search.

A situational analysis concluded that A* search with the Ignore Preconditions Heuristic is more optimal than A* search with the Level Sum Heuristic, as the speed is favoured over resource efficiency. Thus, A* search with the Ignore Preconditions Heuristic is analyzed against Breadth First Search.

Analysis between A* search with the Ignore Preconditions Heuristic and Breadth First Search shows that both produce feasible paths, A* search with the Ignore Preconditions Heuristic executes slightly faster than Breadth First Search and uses significantly less resources (i.e. less nodes expanded, less nodes goal tested and less nodes created). Thus, A* search with the Ignore Preconditions Heuristic is the chosen optimal solution to a logic based Planning and Search problem domain.

The general analysis shows that heuristic based solutions use less resources (i.e. less nodes expanded, less nodes goal tested and less nodes created). However, the execution time is greatly dependent on the heuristic for the situation. If resources are not a concern and an appropriate heuristic is difficult to implement, then non-heuristic based solutions are valid solutions to logic based Planning and Search problem domains. If resources are not a concern or are a concern and an appropriate heuristic can be implemented, then A* search with the heuristic should be implemented as it will use less resources and execute faster than the non-heuristic solutions.

Bibliography

1. Russel, Norvig, S.J, P, 2010. *Artificial Intelligence A Modern Approach*. 3rd ed. University of California, Berkeley: Prentice Hall, Pearson.