Heuristic Analysis

Nathan Findley

March 2017

Contents

Abstract

Evaluation of different search algorithms in a planning problem involving movement of cargo between airports via plane.

1 Optimal Plan

The order of presidence when evaluating an optimal plan is perhaps debatable but I would consider time to find the path be the most essential constraint followed perhaps by the length of the path in the evaluation of a real world problem. Take for instance the actual planning of shipments, the time taken to plan routes is quite critical when it comes to customer satisfaction, but the actual plan itself could, unnecessarily long, result in extreme costs of shipping. Expansions and new nodes are more theoretically interesting it seems to me.

Fortunately given the data presented below, there really isn't much debate to be had about the best plan of action: Breadth First Search is a winner. Greedy Best First Graph Search H1 is a tempting choice if the calculation of the plan is extremely important, but given that it produced a route that was nearly twice as long a Breadth First Search it is doubtful that it would actually be useful in a shipping context.

2 Comparisons

Breadth First Search and Depth First Search have significant differences in results: DFS is extremely quick when it comes to finding a solution. Unfortunately the path that was found was generally an order of magnitude larger than the path found by BFS, leaving DFS as an unlikely candidate.

Breadth First Search and Uniform Cost Search appear to be about on par with one another. They are similar algothrithms so this shouldn't come as a

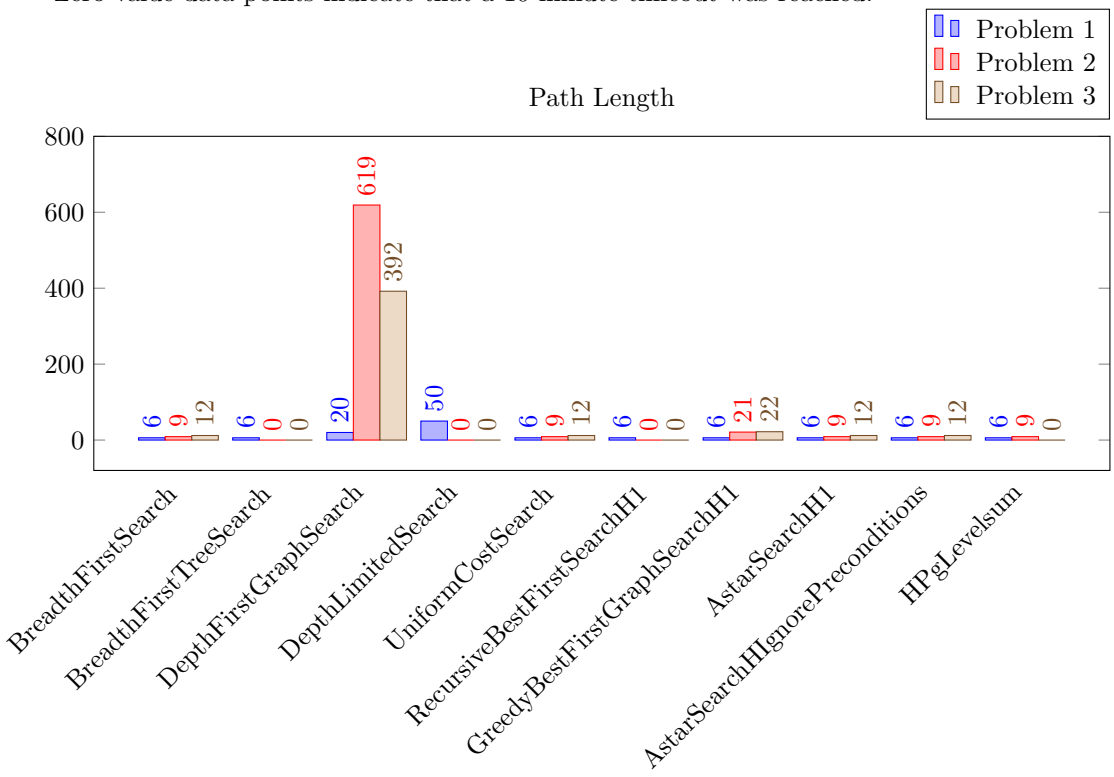
surprise. On all counts BFS is anywhere from moderately to significantly better than UCS. While their path lengths are the same, time elapsed makes BFS a clear winner.

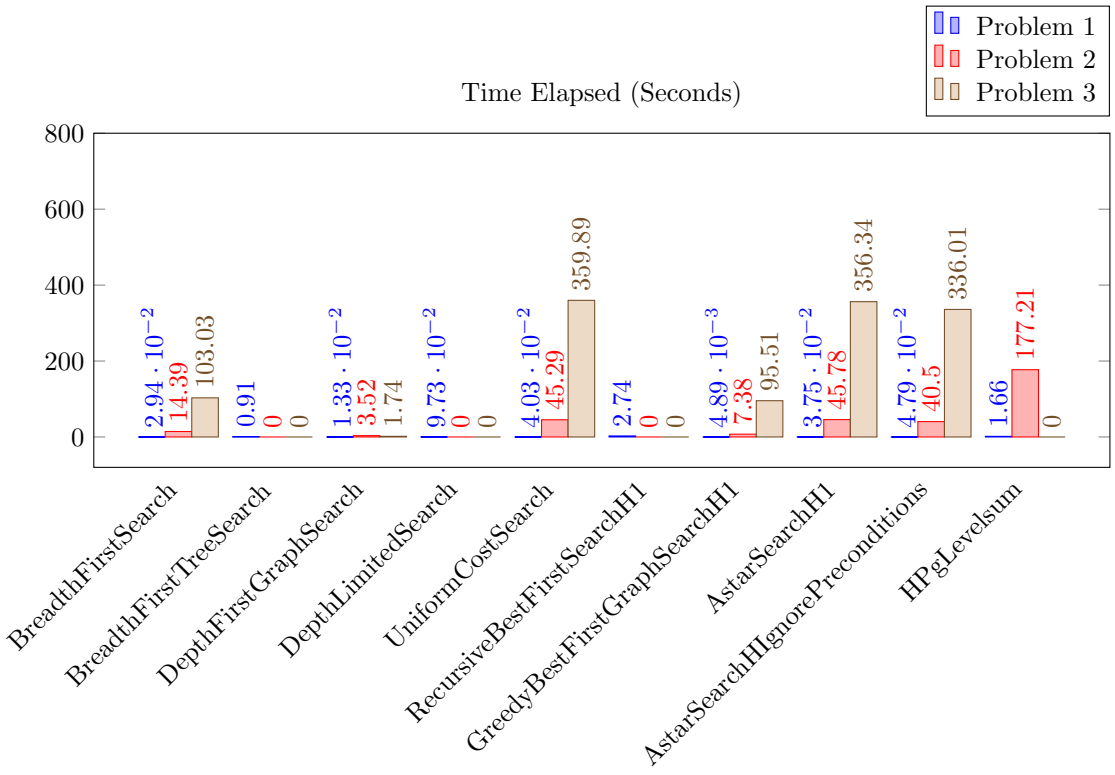
3 A* Heuristics

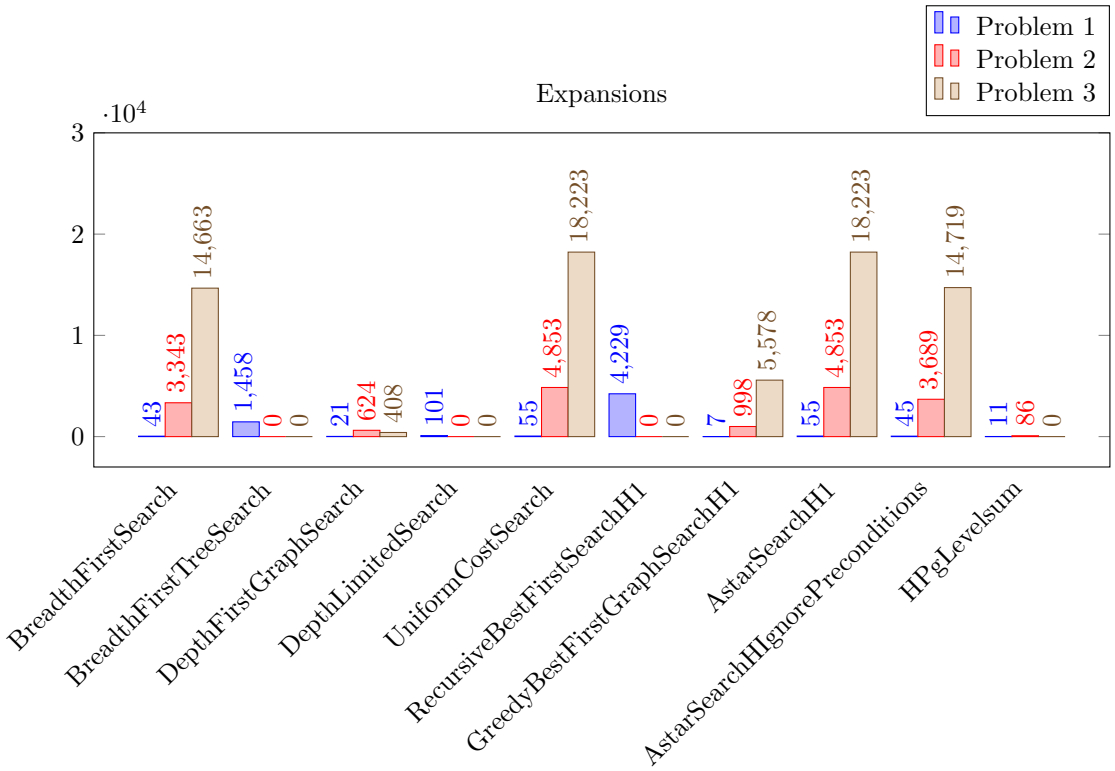
- Compare and contrast heuristic search result metrics using A* with the "ignore preconditions" and "level-sum" heuristics for Problems 1, 2, and 3. - What was the best heuristic used in these problems? Was it better than non-heuristic search planning methods for all problems? Why or why not?

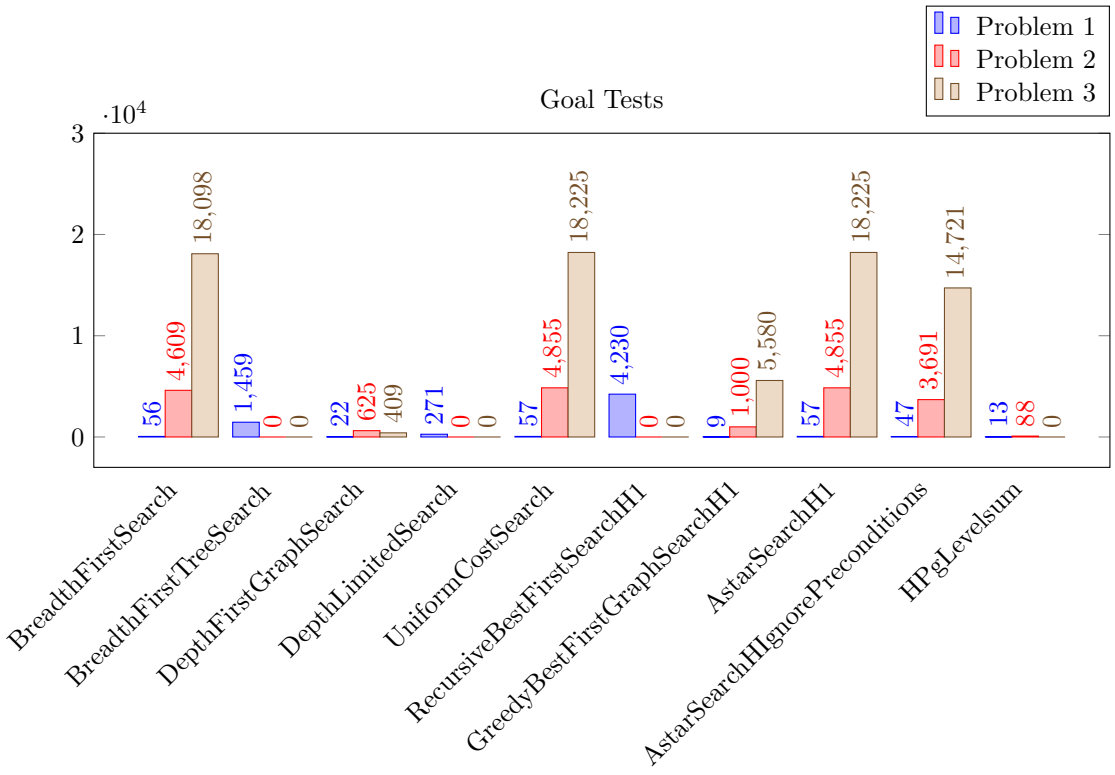
4 Data

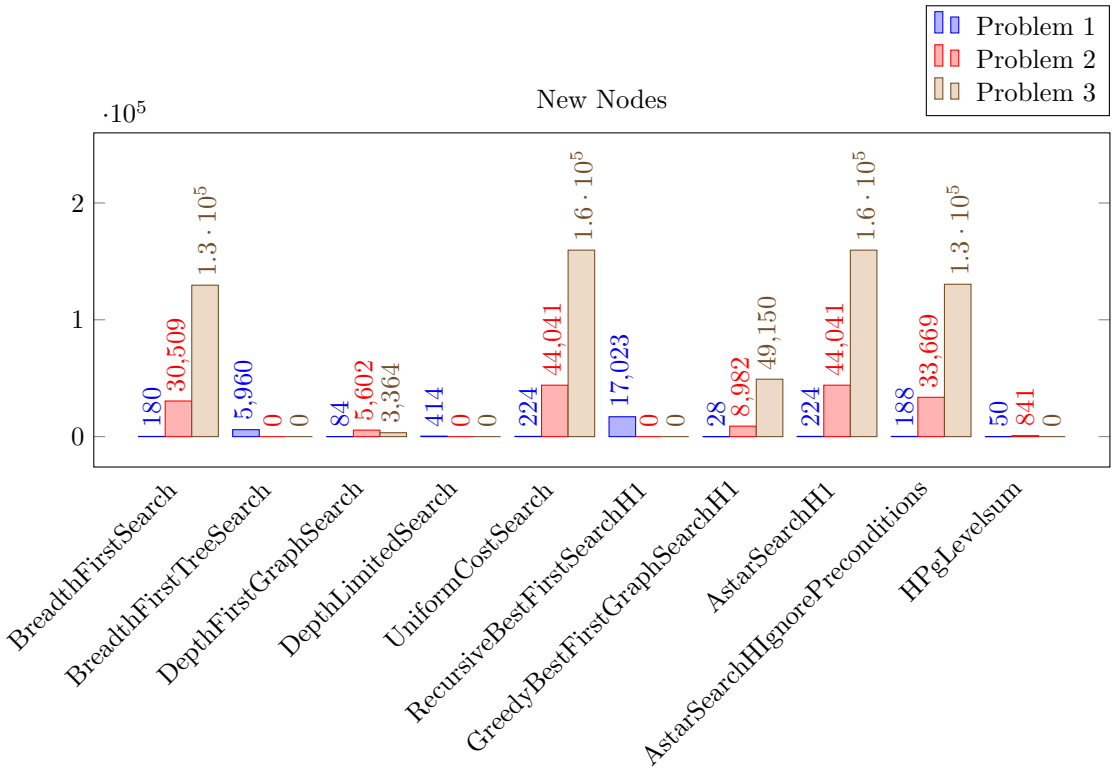
Zero value data points indicate that a 10 minute timeout was reached.




Time Elapsed (Seconds)
Problem 1
Problem 2
Problem 3
800
600
400
200
0
$2.94 \cdot 10^{-2}$
14.39
103.03
0.91
0
0
$1.33 \cdot 10^{-2}$
3.52
1.74
$9.73 \cdot 10^{-2}$
0
0
$4.03 \cdot 10^{-2}$
45.29
359.89
2.74
0
0
$4.89 \cdot 10^{-3}$
7.38
95.51
$3.75 \cdot 10^{-2}$
45.78
356.34
$4.79 \cdot 10^{-2}$
40.5
336.01
1.66
177.21
0
BreadthFirstSearch
BreadthFirstTreeSearch
DepthFirstGraphSearch
DepthLimitedSearch
UniformCostSearch
RecursiveBestFirstSearchH1
GreedyBestFirstGraphSearchH1
AstarSearchH1
AstarSearchHIgnorePreconditions
HPgLevelsum


Expansions
Problem 1
Problem 2
Problem 3
$\cdot 10^4$
3
2
1
0
43
3,343
14,663
1,458
0
0
21
624
408
101
0
0
55
4,853
18,223
4,229
0
0
7
998
5,578
55
4,853
18,223
45
3,689
14,719
11
86
0
BreadthFirstSearch
BreadthFirstTreeSearch
DepthFirstGraphSearch
DepthLimitedSearch
UniformCostSearch
RecursiveBestFirstSearchH1
GreedyBestFirstGraphSearchH1
AstarSearchH1
AstarSearchHIgnorePreconditions
HPgLevelsum


Goal Tests
Problem 1
Problem 2
Problem 3
$\cdot 10^4$
3
2
1
0
56
4,609
18,098
1,459
0
0
22
625
409
271
0
0
57
4,855
18,225
4,230
0
0
9
1,000
5,580
57
4,855
18,225
47
3,691
14,721
13
88
0
BreadthFirstSearch
BreadthFirstTreeSearch
DepthFirstGraphSearch
DepthLimitedSearch
UniformCostSearch
RecursiveBestFirstSearchH1
GreedyBestFirstGraphSearchH1
AstarSearchH1
AstarSearchHIgnorePreconditions
HPgLevelsum


New Nodes
Problem 1
Problem 2
Problem 3
$\cdot 10^5$
2
1
0
180
30,509
$1.3 \cdot 10^5$
5,960
0
0
84
5,602
3,364
414
0
0
224
44,041
$1.6 \cdot 10^5$
17,023
0
0
28
8,982
49,150
224
44,041
$1.6 \cdot 10^5$
188
33,669
$1.3 \cdot 10^5$
50
841
0
BreadthFirstSearch
BreadthFirstTreeSearch
DepthFirstGraphSearch
DepthLimitedSearch
UniformCostSearch
RecursiveBestFirstSearchH1
GreedyBestFirstGraphSearchH1
AstarSearchH1
AstarSearchHIgnorePreconditions
HPgLevelsum