

Overview

This paper compares and discusses the test results for non-heuristic and heuristic based search solutions. Non-heuristic based searches include Breadth First, Depth First and Uniform Cost. Heuristic based searches include A* w/ Ignore Preconditions and A* w/ PG LevelSum. First, a comparison of non-heuristic based searches. Next, a comparison of heuristic based searches. Finally, a comparison of non-heuristic and heuristic based searches and discussion of the optimal search solution.

Non-Heuristic Search

	Breadth First Search				
	Expansions	Goal Tests	New Nodes	Plan Length	Elapsed Time (sec)
Air Cargo Problem					
1	43	56	180	6	0.06
2	3343	4609	30509	9	24.24
3	14663	18098	129631	12	173.41

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

Figure 1

	Depth First Search				
	Expansions	Goal Tests	New Nodes	Plan Length	Elapsed Time (sec)
Air Cargo Problem					
1	12	13	48	12	0.02
2	582	583	5211	575	5.36
3	627	628	5176	596	5.71

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

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

Figure 2

	Uniform Cost Search				
	Expansions	Goal Tests	New Nodes	Plan Length	Elapsed Time (sec)
Air Cargo Problem					
1	55	57	224	6	0.08
2	4853	4855	44041	9	22.44
3	18236	18238	159726	12	97.84

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

Figure 3

In Breadth First and Depth First, the data is organized in a tree whereas Uniform Cost organizes data in a graph. Based on the three figures above, Uniform Cost and Breadth First searches are similar in the number of node expanded, number of goal tests, discovery of new nodes and elapsed times. The question is which search provides the most optimal solution? What feature or criteria determines the most optimal search solution? At first glance, Depth First takes a shorter time to execute than Breadth First and Uniform Cost but Depth First search takes many more iterations to get to the goal than the other two search solutions. What is the problem that is needs to be solved?

Problem: Which route/path to take to minimize time and cost.

Test results, above, show Breadth First and Uniform Cost provide a good combination of duration and iterations to reach the goal(s). Breadth First has less expansions, tests, and nodes discovered than Uniform Cost. Surprisingly, Uniform Cost has lower elapsed time as the number of

preconditions increase. Let's explore Breadth First and Uniform Cost further. The AIMA text states "When all step costs are equal, breadth-first search is optimal because it always expands the *shallowest* unexpected node. By a simple extension, we can find an algorithm that is optimal with any step-cost function. Instead of expanding the shallowest node, **uniform-cost search** expands the node n with the *lowest path cost $g(n)$* ."¹ Uniform-cost applies the goal-test to a node when the node is selected for expansion as opposed to creation time. "The reason is that the first goal node that is generated may be on a suboptimal path. The second difference is that a test is added in case a better path is found to a node currently on the frontier"² Uniform Cost is the optimal solution amongst the non-heuristic based search solutions that were tested.

Heuristic Search

A* Search w/ Ignore Preconditions Heuristic					
	Expansions	Goal Tests	New Nodes	Plan Length	Elapsed Time (sec)
Air Cargo Problem					
1	41	43	170	6	0.08
2	1450	1452	13303	9	8.98
3	5040	5042	44944	12	34.62

Figure 4

A* Search w/ PG Level Sum Heuristic					
	Expansions	Goal Tests	New Nodes	Plan Length	Elapsed Time (sec)
Air Cargo Problem					
1	11	13	50	6	1.01
2	86	88	841	9	86.67
3	318	320	2934	12	432.88

Figure 5

Let's shift gears and examine another set of search solutions with additional insights. More specifically, A* Search and the use of heuristic(s) to provide additional insights to solve the problem in a cost efficient and time sensitive manner. A* Search evaluates a node by determining the cost to reach a node and the cost to get from the node to the goal. "It evaluates nodes by combining $g(n)$, the cost to

¹ Norvig, Peter and Russell, Stuart (2010), AIMA, Pg. 84

² Norvig, Peter and Russell, Stuart (2010), AIMA, Pg. 84-85

reach a node, and $h(n)$, the cost to get from the node to the goal: $f(n) = g(n) + h(n)$. Since $g(n)$ gives the path cost from the start node to node n , and $h(n)$ is the estimated cost of the cheapest path from n to the goal, we have $f(n) = \text{estimated cost of the cheapest solution through } n$.”³ How is A* Search different than Uniform-Cost Search? “The algorithm is identical to UNIFORM-COST-SEARCH except that A* uses $g + h$ instead of g .”⁴

What is a heuristic? Merriam Webster defines Heuristic as “of or relating to exploratory problem-solving techniques that utilize self-educating techniques (such as the evaluation of feedback) to improve performance.”⁵ The tests include two heuristics: 1) Ignore Preconditions and 2) Level Sum. As the name implies, Ignore Preconditions allows A* Search to explore each node without conditions or boundaries. “Every action becomes applicable in every state, and any single goal fluent can be achieved in one step (if there is an applicable action – if not, the problem is impossible).”⁶ Planning Graph is a special data structure that helps minimize inaccuracies. Level Sum is a planning graph that returns the sum of the level costs of the goals. As depicted in the two figures above, Ignore Preconditions heuristic along with A* Search yields better performance in a shorter period of time than Level Sum heuristic along with A* Search. Why? The ignore preconditions heuristic has no limitations and is able to evaluate more nodes in a shorter period of time. On the other hand, Level Sum needs to evaluate less nodes to get to the goal(s). “The level sum heuristic, following the subgoal independence assumption, returns the sum of the level costs of the goals; this can be inadmissible but works in practice for problems that are largely decomposable. It is much more accurate than the number-of-unsatisfied-goals heuristic from Section 10.2”⁷

Of the two heuristic based searches, A* Search w/ PG Level Sum performs better than A* Search w/ Ignore Preconditions because Level Sum surveys the levels for the goal(s) first and then attempts to determine the best route to get to goal(s). A* Search w/ PG Level Sum heuristic is the optimal solution.

Conclusion

A* Search w/ PG Level Sum heuristic is the optimal choice for solving planning problems when the number of conditions increase. A* Search w/ PG Level Sum heuristic is more efficient and reduces the number of nodes expanded with smaller number of plans requires to reach the goal(s). The ignore preconditions heuristic is best used for quick assessment and then level sum for more accurate results. Let’s take the solution a bit further – first use ignore preconditions heuristic to eliminate as many invalid paths as possible then use level sum heuristic for refined search and the reduced number of nodes will also reduce the elapsed time.

³ Norvig, Peter and Russell, Stuart (2010), AIMA, Pg. 93

⁴ Norvig, Peter and Russell, Stuart (2010), AIMA, Pg. 93

⁵ Merriam-Webster Online (<https://www.merriam-webster.com/dictionary/heuristic>)

⁶ Norvig, Peter and Russell, Stuart (2010), AIMA, Pg. 376

⁷ Norvig, Peter and Russell, Stuart (2010), AIMA, Pg. 382