Implement a Planning Search.
Project results

Alexey Denisov

01/13/2018

Part 1. Planning problems

The following tables show the results obtained from seven non-heuristic search types (see the tables) applied to air cargo problems for this project. The goal of this analysis is to compare the outcomes of different search types and to find an optimal solution for each of the problems.

The optimal solution is the lowest path among all possible paths from start to goal.

For each problem the optimal solution is highlighted.

Problem 1

Algorithm	Heuristic	Optimality	Optimal & Best time	Expansions	Goal Tests	New Nodes	Plan length	Time elapsed in seconds
Breadth First Search		yes		43	56	180	6	0,10
Breadth First Tree Search		yes		1458	1459	5960	6	2,97
Depth First Graph Search		no		21	22	84	20	0,05
Depth Limited Search		no		101	271	414	50	0,29
Uniform Cost Search		yes		55	57	224	6	0,12
Recursive Best First Search	H1	yes		4229	4230	17023	6	8,45
Greedy Best First Graph Search	H1	yes	yes	7	9	28	6	0,02

The outcome of the GBFS is following:(aind) a_denisov@denisov-HP-ProBook-4310s:~/PycharmProjects/AI Udacity/AIND-Planning$ python run_search.py -p 1 -s 7

Solving Air Cargo Problem 1 using greedy_best_first_graph_search with h_1…

Expansions Goal Tests New Nodes
7 9 28

Plan length: 6 Time elapsed in seconds: 0.020334993999313156

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

Algorithm	Heuristic	Optimality	Optimal & Best time	Expansions	Goal Tests	New Nodes	Plan length	Time elapsed in seconds
Breadth First Search		yes	yes	3343	4609	30509	9	23,3
Breadth First Tree Search		no					too long	
Depth First Graph Search		no		624	625	5602	619	9,1
Depth Limited Search		no					too long	
Uniform Cost Search		yes		4852	4854	44030	9	32,8
Recursive Best First Search	H1	no					too long	
Greedy Best First Graph Search	H1	no		990	992	8910	17	6,6

The outcome of BFS is following:(aind) a_denisov@denisov-HP-ProBook-4310s:~/PycharmProjects/AI Udacity/AIND-Planning$ python run_search.py -p 2 -s 1

Solving Air Cargo Problem 2 using breadth_first_search...

Expansions Goal Tests New Nodes
3343 4609 30509

Plan length: 9 Time elapsed in seconds: 23.327612735000002
Load(C1, P1, SFO)
Load(C2, P2, JFK)
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)

Problem 3

Algorithm	Heuristic	Optimality	Optimal & Best time	Expansions	Goal Tests	New Nodes	Plan length	Time elapsed in seconds
Breadth First Search		yes	yes	14663	18098	129631	12	118,3
Breadth First Tree Search		no					too long	
Depth First Graph Search		no		408	409	3364	392	4,5
Depth Limited Search		no					too long	
Uniform Cost Search		yes		18235	18237	159716	12	140,7
Recursive Best First Search	H1	no					too long	
Greedy Best First Graph Search	H1	no		5614	5616	49429	22	43,5

The outcome of BFS is following:

(aind) a_denisov@denisov-HP-ProBook-4310s:~/PycharmProjects/AI Udacity/AIND-Planning$ python run_search.py -p 3 -s 1

Solving Air Cargo Problem 3 using breadth_first_search...

Expansions Goal Tests New Nodes
14663 18098 129631

Plan length: 12 Time elapsed in seconds: 118.2852048220002
Load(C1, P1, SFO)
Load(C2, P2, JFK)
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)

Part 2. Domain-independent heuristics

The following tables show the results obtained from three A* search types with different heuristics (see the tables) applied to air cargo problems for this project. The goal of this analysis is to compare the outcomes and to find an optimal solution for each of the problems.

The optimal solution is the lowest path among all possible paths from start to goal.

Problem 1

Algorithm	Heuristic	Optimality	Optimal & Best time	Expansions	Goal Tests	New Nodes	Plan length	Time elapsed in seconds
A* Search	H1	yes	yes	55	57	224	6	0,13
A* Search	Ignore Preconditions	yes		41	43	170	6	0,14
A* Search	The sum of the level costs of the individual goals	yes		11	13	50	6	1,61

For each problem the optimal solution is highlighted.

The outcome of the A*Search with H1 heuristic is following:

(aind) a_denisov@denisov-HP-ProBook-4310s:~/PycharmProjects/AI Udacity/AIND-Planning$ python run_search.py -p 1 -s 8

Solving Air Cargo Problem 1 using astar_search with h_1...

Expansions Goal Tests New Nodes
55 57 224

Plan length: 6 Time elapsed in seconds: 0.13157273299293593
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

Algorithm	Heuristic	Optimality	Optimal & Best time	Expansions	Goal Tests	New Nodes	Plan length	Time elapsed in seconds
A* Search	H1	yes		4852	4854	44030	9	35,66
A* Search	Ignore Preconditions	yes	yes	1450	1452	13303	9	14,05
A* Search	The sum of the level costs of the individual goals	yes		86	88	841	9	156,23

The outcome of the A*Search with the Ignore Preconditions heuristic is following:

(aind) a_denisov@denisov-HP-ProBook-4310s:~/PycharmProjects/AI Udacity/AIND-Planning$ python run_search.py -p 2 -s 9

Solving Air Cargo Problem 2 using astar_search with h_ignore_preconditions...

Expansions Goal Tests New Nodes
1450 1452 13303

Plan length: 9 Time elapsed in seconds: 14.049305744003505
Load(C3, P3, ATL)
Fly(P3, ATL, SFO)
Unload(C3, P3, SFO)
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 3

Algorithm	Heuristic	Optimality	Optimal & Best time	Expansions	Goal Tests	New Nodes	Plan length	Time elapsed in seconds
A* Search	H1	yes		18235	18237	159716	12	151,70
A* Search	Ignore Preconditions	yes	yes	5040	5042	44944	12	48,62
A* Search	The sum of the level costs of the individual goals	yes		318	320	2934	12	653,91

The outcome of the A*Search with the Ignore Preconditions heuristic is following:

(aind) a_denisov@denisov-HP-ProBook-4310s:~/PycharmProjects/AI Udacity/AIND-Planning$ python run_search.py -p 3 -s 9

Solving Air Cargo Problem 3 using astar_search with h_ignore_preconditions...

Expansions Goal Tests New Nodes
5040 5042 44944

Plan length: 12 Time elapsed in seconds: 48.623760232003406
Load(C2, P2, JFK)
Fly(P2, JFK, ORD)
Load(C4, P2, ORD)
Fly(P2, ORD, SFO)
Unload(C4, P2, SFO)
Load(C1, P1, SFO)
Fly(P1, SFO, ATL)
Load(C3, P1, ATL)
Fly(P1, ATL, JFK)
Unload(C3, P1, JFK)
Unload(C1, P1, JFK)
Unload(C2, P2, SFO)

Part 3. Written Analysis

The optimal plans for each of the problems are following:

Problem 1 (BFS)

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

Fly(P1, SFO, JFK)
Unload(C1, P1, JFK)

Problem 2 (BFS)

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

Problem 3 (BFS)

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

Non-heuristic test results (see Part 1)

Five out of seven non-heuristic searches resulted in optimal solutions for Problem 1: BFS, BFTS, UCS, RBFS and GBFGS. Two searches – DFGS and DLS – resulted in much longer plans, 20 and 50 instead of 6, respectively. GBFGS became the fastest out of all searches.

Only two out of seven non-heuristic searches resulted in optimal solutions for Problem 2: BFS and UCS. Two searches – GBFGS and DFGS – resulted in much longer plans, 17 and 619 instead of 9, respectively. Three searches – BFTS, DLS and RBFS – did not finish in a reasonable amount of time (took longer than 10 minutes). BFS became the fastest out of all searches that gave an optimal solution.

The same two non-heuristic searches - BFS and UCS - out of seven resulted in optimal solutions for Problem 3. Two searches – GBFGS and DFGS – resulted in much longer plans, 22 and 392 instead of 12, respectively. Three searches – BFTS, DLS and RBFS – did not finish in a reasonable amount of time (took longer than 10 minutes). BFS became the fastest out of all searches hat gave an optimal solution.

Based on these experiments, we may conclude that only BFS consistently gives an optimal and speedy solution among non-heuristic searches.

According to *Artificial Intelligence. A Modern Approach by Russel and Norvig,* the memory requirements are a bigger problem for BFS than is the execution time. We can see this in the *Expansions* and *New Nodes* columns. Moreover, exponential-complexity search problems cannot be solved by uniformed methods for any but the smallest instances.

Heuristics test results (see Part 2)

All three heuristic searches resulted in optimal solutions for Problem 1: H1, Ignore_Preconditions and LevelSum. H1 became the fastest out of all searches and gave slightly better result than Ignore_Preconditions.

All three heuristic searches resulted in optimal solutions for Problem 2: H1, Ignore_Preconditions and LevelSum. Ignore_Preconditions became the fastest out of all searches.

All three heuristic searches resulted in optimal solutions for Problem 3: H1, Ignore_Preconditions and LevelSum. Ignore_Preconditions became the fastest out of all searches.

Based on these experiments, we may conclude that A* Search with Ignore Preconditions heuristic consistently gives an optimal and speedy solution. The more complex heuristic – LevelSum - does not give solutions that are better than Ignore_Preconditions when applied to Problems 1,2 and 3 despite it is more accurate. It may be related to the size of the problems – they are small enough to use the better accuracy.

Overall, heuristic based search performs better as the complexity of the problem increased. This is more evident in the Problems 2 and 3, where A* search with Ignore_Preconditions heuristics outperformed all the other types of searches, non-heuristic and heuristic.