Planning Search - Heuristic Analysis

Author: Alexander Ponamarev

Summary

The following document presents an overview of a problem, solutions, and an analysis of various classical planning algorithms used to obtain solutions. Planning Search project implements a search agent for a fully observable, discrete, deterministic problem of logistics planning problems for an Air Cargo transport system. Project problems are described as following:

EXHIBIT 1:

Problem 1	Problem 2	Problem 3
Init(At(C1, SFO) ∧ At(C2, JFK) ∧ At(P1, SFO) ∧ At(P2, JFK) ∧ Cargo(C1) ∧ Cargo(C2) ∧ Plane(P1) ∧ Plane(P2) ∧ Airport(JFK) ∧ Airport(SFO))	Init(At(C1, SFO) ∧ At(C2, JFK) ∧ At(C3, ATL) ∧ At(P1, SFO) ∧ At(P2, JFK) ∧ At(P3, ATL) ∧ Cargo(C1) ∧ Cargo(C2) ∧ Cargo(C3) ∧ Plane(P1) ∧ Plane(P2) ∧ Plane(P3) ∧ Airport(JFK) ∧ Airport(SFO) ∧ Airport(ATL))	Init(At(C1, SFO) ∧ At(C2, JFK) ∧ At(C3, ATL) ∧ At(C4, ORD) ∧ At(P1, SFO) ∧ At(P2, JFK) ∧ Cargo(C1) ∧ Cargo(C2) ∧ Cargo(C3) ∧ Cargo(C4) ∧ Plane(P1) ∧ Plane(P2) ∧ Airport(JFK) ∧ Airport(SFO) ∧ Airport(ATL) ∧ Airport(ORD))
Goal(At(C1, JFK) ∧ At(C2, SFO))	Goal(At(C1, JFK) ∧ At(C2, SFO) ∧ At(C3, SFO))	Goal(At(C1, JFK) ∧ At(C3, JFK) ∧ At(C2, SFO) ∧ At(C4, SFO))

The following search algorithms were used to find a solution:

EXHIBIT 2:

Uninformed Search	Informed (Heuristic) Search
Breadth-First Search (BFS)	A* H1 (Uniform Cost Search)
Depth-First Search (DFS)	A* Ignore Preconditions (relaxed problem)
Uniform Cost Search (UCS)	A* LevelSum (level cost of remaining conditions)

All search algorithms found a solution within a pre-defined time limit (10 minutes). The performance of algorithms was compared based on memory usage (measured in expanded nodes) and execution time (measured in seconds). Please note that Uniform Cost Search and A* H1 search perform exactly the same algorithm, therefore results reported only once as Uniform Cost Search. A summary of algorithm performance presented below:

EXHIBIT 3:

UNINFORMED SEARCH

	Air Cargo Problem 1		Problem 2		Problem 3	
	Expansions	Time Elapsed (sec)	Expansions	Time Elapsed (sec)	Expansions	Time Elapsed (sec)
BFS (Optimal)	43	0.028	3343	7.46	14663	35.7
UCS (Optimal)	55	0.038	4853	10.59	18223	42.6
DFS (Not Optimal)	21	0.011	624	3.06	408	1.55

INFORMED SEARCH

Air Cargo Problem 1.

	Expansions	Time Elapsed (sec)
A* Ignore Preconditions (Optimal)	41	0.036
A* Level Sum (Optimal)	11	0.482

Air Cargo Problem 2

	Expansions	Time Elapsed (sec)
A* Ignore Preconditions (Optimal)	1450	3.7
A* Level Sum (Optimal)	86	36.1

Air Cargo Problem 3

	Expansions	Time Elapsed (sec)
A* Ignore Preconditions (Optimal)	5040	14.3
A* Level Sum (Optimal)	325	185.9

Analysis

All uninformed (non-heuristic) search algorithm found problem solutions. Out of all tested algorithms, depth-first search (DFS) demonstrated the best speed on all three problems. However, DFS generally is not guaranteed to find an optimal solution, which was confirmed in practice. In case of Problem 2, a solutions provided by DFS algorithm is more than 67 times longer than the optimal solution.

The table below compares the length of a final solution provided by the algorithm:

EXHIBIT 4

Solution Path Length (Air Cargo Problems)

	Problem 1	Problem 2	Problem3
BFS (Optimal)	6	9	12
UCS (Optimal)	6	9	12
DFS (Not Optimal)	20	619	392
A* Ignore Preconditions (Optimal)	6	9	12
A* Level Sum (Optimal)	6	9	12

Breadth-first search (BFS) was overall the second fastest algorithm on lower complexity problems (Problem 1 and 2). Given enough time, BFS algorithm finds an optimal solution, and it should be considered as the main choice for low-complexity problems.

Uniform Cost Search and A* H1 search algorithm use the same heuristic for counting path length. This heuristic did not provide useful information for the search algorithm resulting in worse performance as compared to BFS both in terms of speed and memory requirement on all three problems (irrespective of a problem complexity).

A* search algorithms that used an automatic heuristic (ignore precondition) and a domain knowledge based heuristic (level sum) incurred a substantial computational cost resulting in slower performance as compared to BFS algorithm. However, both of heuristics proofed to be useful in optimizing memory use. Suggesting that A* search algorithm could outperform BFS when looking for a solution for more complex problems with a greater number of actions and environment states. This finding was further supported by the performance of A* search on a more complex problem - Air Cargo Problem 3, where A* Ignore Preconditions search was the fastest optimal algorithm finding a solution in 14.3 seconds (vs. BFS - 35.7 seconds).

Optimal Solutions

Table below provides optimal solutions for Air Cargo Problems 1-3.

EXHIBIT 5

Air Cargo Solutions

Problem	Solution
Problem 1	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	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	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(C2, P2, SFO), Unload(C1, P1, JFK)