Planning Search

Analysis of Results

Robert Crowe – v1.0 19 July 2017

Overview

As an exploration of the field of planning search, 3 different search problems were defined in classical PDDL (Planning Domain Definition Language) for the air cargo domain. A suite of different planning and search algorithms were used to develop plans to solve these problems, and the results were compared. This included automatically generated domain-independent heuristics, including planning graph heuristics, as well as uniformed planning searches including breadth-first and depth-first. The results of different approaches were then compared for optimality of the resulting plans as well as performance and efficiency.

Results

Problem 1

Search Function	Expansions	Goal Tests	New Nodes	Elapsed Secs	Plan Length
Breadth First	43	56	180	0.028	6
Breadth First Tree	1458	1459	5960	0.77	6
Depth First Graph	12	13	48	0.008	12
Depth Limited	101	271	414	0.08	50
Uniform Cost	55	57	224	0.034	6
Recursive Best First h_1	4229	4230	17029	2.28	6
Greedy Best First Graph h_1	7	9	28	0.005	6
A* h_1	55	57	224	0.034	6
A* h_ignore_preconditions (stub)	55	57	224	0.036	6
A* h_ignore_preconditions (real)	41	43	170	0.03	6
A* h_pg_levelsum	55	57	224	1.59	6

Problem 2

Search Function	Expansions	Goal Tests	New Nodes	Elapsed Secs	Plan Length
Breadth First	3401	4672	31049	11.68	9
Breadth First Tree	(did not complete in 4 hours)				
Depth First Graph	350	351	3142	1.26	346
Depth Limited	254020	2344879	2345254	925.42	50
Uniform Cost	4761	4763	43206	10.48	9
Recursive Best First h_1	(did not complete in 4 hours)				
Greedy Best First Graph h_1	550	552	4950	1.2	9
A* h_1	4761	4763	43206	10.27	9
A* h_ignore_preconditions (stub)	4761	4763	43206	10.31	9

A* h_ignore_preconditions (real)	1450	1452	13303	3.9	9
A* h_pg_levelsum	4761	4763	43206	1843.45	9

Problem 3

Search Function	Expansions	Goal Tests	New Nodes	Elapsed Secs	Plan Length
Breadth First	14491	17947	128184	87.33	12
Breadth First Tree	(did not complete in 30 minutes)				
Depth First Graph	1948	1949	16253	16.51	1878
Depth Limited	(did not complete in 4 hours)				
Uniform Cost	17783	17785	155920	44.89	12
Recursive Best First h_1	(did not complete in 30 minutes)				
Greedy Best First Graph h_1	4031	4033	35794	10.29	22
A* h_1	17783	17785	155920	45.76	12
A* h_ignore_preconditions (stub)	17783	17785	155920	46.98	12
A* h_ignore_preconditions (real)	5003	5005	44586	17.05	12
A* h_pg_levelsum	17783	17785	155920	10346	12

Action Schema

The problem domain is air cargo – moving cargo from one airport to another using airplanes.

Action(Load(c, p, a),

PRECOND: At(c, a) ∧ At(p, a) ∧ Cargo(c) ∧ Plane(p) ∧ Airport(a)

EFFECT: ¬ At(c, a) ∧ In(c, p))

Action(Unload(c, p, a),

PRECOND: In(c, p) ∧ At(p, a) ∧ Cargo(c) ∧ Plane(p) ∧ Airport(a)

EFFECT: At(c, a) ∧ ¬ In(c, p))

Action(Fly(p, from, to),

PRECOND: At(p, from) ∧ Plane(p) ∧ Airport(from) ∧ Airport(to)

EFFECT: ¬ At(p, from) ∧ At(p, to))

Problem 1

Initial state and goal

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))

Goal(At(C1, JFK) ∧ At(C2, SFO))

Optimal Solution

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

Initial state and goal

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))

Goal(At(C1, JFK) ∧ At(C2, SFO) ∧ At(C3, SFO))

Optimal Solution

Load(C1, P1, SFO)

Load(C2, P2, JFK)

Load(C3, P3, ATL)

Fly(P1, SFO, JFK)

Fly(P2, JFK, SFO)

Fly(P3, ATL, SFO)

Unload(C3, P3, SFO)

Unload(C2, P2, SFO)

Unload(C1, P1, JFK)

Problem 3

Initial state and goal

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(C3, JFK) ∧ At(C2, SFO) ∧ At(C4, SFO))

Optimal Solution

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)

Non-Heuristic Search Comparison

Breadth-first search and breadth-first tree search both produced optimal plans, when they were tractable. For these problems breadth-first always completed in a reasonable time, however the growth shown in the amount of time to complete problem 3 suggests that for larger problems it would probably be too inefficient to use. Breadth-first tree search was far less efficient, and only completed for Problem 1. For breadth-first the space complexity was not bad, creating 128,184 new nodes for problem 3, compared to 44,586 for A* using the ignore preconditions heuristic while still finding optimal plans.

Depth-first graph search completed quickly (16.51s vs 87.33s for breadth-first on problem 3), but never found an optimal plan. It was always wildly off, for example finding a plan of length 1,878 for problem 3 compared to the optimal length of 12. Space and time complexity was however consistently good, see results.

Uniform cost search was not bad overall. It always found an optimal plan in a reasonable time, and was the fastest of the non-heuristic searches that found optimal plans on problems 2 and 3. The space complexity was always higher than breadth-first, but not wildly so, for example creating 155,920 new nodes for problem 3 compared to 128,184 for breadth-first.

Heuristic Search Comparison: A*

A* search was run using a constant heuristic (see results for “A* h_ignore_preconditions (stub)”) and an actual ignore preconditions heuristic (“A* h_ignore_preconditions (real)”). A level sum heuristic was also used with A*, and results gathered.

In all cases, using all heuristics, A* produced an optimal plan. However while there was no difference between the constant and the level sum heuristics in terms of space complexity, measured by new nodes created, the difference in time complexity was significant: for problem 3, the constant heuristic required 46.98s to reach an optimal plan, while level sum required 10,346s.

But the real star of A* was the ignore preconditions heuristic. This heuristic estimates the minimum number of actions that must be carried out from the current state in order to satisfy all of the goal conditions by ignoring the preconditions required for an action to be executed.

This implementation accounts for actions that achieve multiple positive goals, while ignoring delete lists. Space complexity was in the range of 33% of that for the other A* heuristics (3X reduction) and the time required was also in the range of 33% of that required for the constant heuristic. This suggests that the improved accuracy of the heuristic resulted in a significantly reduced traversal of the search space, resulting in vastly improved time and space complexity.

Conclusion

Of all the search methods tried, for those that were able to produce optimal plans, A* search using the ignore preconditions heuristic was by far the best performer in both time and space. This is probably because the improved accuracy of the heuristic resulted in a significantly reduced traversal of the search space, resulting in vastly improved time and space complexity.