Aman Ahluwalia
Artificial Intelligence Nanodegree
29 July 2017

Planning Search Report

Part 1 - Planning Problems

Progression planning problems can be solved with graph searches such as breadth-first, depth-first, and A*, where the nodes of the graph are "states" and edges are "actions". A "state" is the logical conjunction of all boolean ground "fluents", or state variables, that are possible for the problem using Propositional Logic.

Here we will be dealing with three planning problems in the air cargo domain. The basic air cargo action schema is,

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))

The three planning problems we will be dealing with are,

air_cargo_p1,

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))

The optimal plan for this problem would be,

Load(C2, P2, JFK)

Load(C1, P1, SFO)

Fly(P2, JFK, SFO)

Unload(C2, P2, SFO)

Fly(P1, SFO, JFK)

Unload(C1, P1, JFK)

Resultant metrics on running uninformed planning searches for air_cargo_p1,

	Expansions	Goal Tests	New Nodes	Time elapsed
breadth_first_search	43	56	180	0.0450
depth_first_graph_search	12	13	48	0.01449
uniform_cost_search	55	57	224	0.05921

air_cargo_p2,

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))

The optimal plan for this problem would be,

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)

Resultant metrics on running uninformed planning searches for air_cargo_p2,

	Expansions	Goal Tests	New Nodes	Time elapsed
breadth_first_search	3343	4609	30509	17.7448
depth_first_graph_search	582	583	5211	4.1363
uniform_cost_search	4853	4855	44041	14.5444

air_cargo_p3,

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))

The optimal plan for this problem would be,

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)

Resultant metrics on running uninformed planning searches for air_cargo_p3,

	Expansions	Goal Tests	New Nodes	Time elapsed
breadth_first_search	14663	18098	129631	135.040
depth_first_graph_search	627	628	5176	4.3369
uniform_cost_search	18151	18153	159038	67.7834

Part 1 - Domain-independent heuristics

The planning graph is somewhat complex, but is useful in planning because it is a polynomial-size approximation of the exponential tree that represents all possible paths. The planning graph can be used to provide automated admissible heuristics for any domain.

Here we will compare and contrast heuristic search result for the three air cargo problem, with simple astar search, astar search with ignore preconditions and level-sum.

Resultant metrics on running the three algorithms on air_cargo_p1,

	Expansions	Goal Tests	New Nodes	Time elapsed
astar_search h_1	55	57	224	0.06536
astar_search h_ignore_preconditions	57	59	230	0.05844
astar_search h_pg_levelsum	11	13	50	0.77926

Resultant metrics on running the three algorithms on air_cargo_p2,

	Expansions	Goal Tests	New Nodes	Time elapsed
astar_search h_1	4853	4855	44041	18.3726
astar_search h_ignore_preconditions	3824	3826	34866	16.0598
astar_search h_pg_levelsum	86	88	841	69.0608

Resultant metrics on running the three algorithms on air_cargo_p3,

	Expansions	Goal Tests	New Nodes	Time elapsed
astar_search h_1	18151	18153	159038	73.3452
astar_search h_ignore_preconditions	11265	11267	100437	45.1349
astar_search h_pg_levelsum	314	316	2894	317.9294

According to the above results and the problem in hand definitely we can comment level sum is the best heuristic which can be used in the problem. We were able to make this strict statement on the basis of expansion, goal tests and new nodes result as shown in the above tables. The results were drastic. The heuristics help to reduce the size of the problem dramatically, though it took some time to do so, but still the benefits it gave overcome this negativeness it brought.

Using the heuristics definitely help in reducing the problem sample space and concentrating more or on the required nodes. We can definitely see from the above results using heuristics with a-star search definitely eased the problem by reduced the number of new and explored nodes otherwise the normal a-star does. The use of heuristics actually brings in some restriction, which avoids visiting some unnecessary nodes.