Udacity Artificial Intelligence Nanodegree Assignment 3

Planning Search Heuristic Analysis

Richard C. Stilborn

Introduction

This section of the nanodegree was about planning and the different search techniques and heuristics that can be used to find the optimal route or sequence of steps for a given problem domain. The chosen scenario was air cargo shipping. For the assignment we had to:

1. Implement methods of the air cargo problem
2. Implement methods of the planning graph
3. Implement and evaluate 2 different evaluation heuristics.

Air Cargo Domain PDDL

Air Cargo Action Schema:

```
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))
```

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))
```

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 Solutions

Problem 1 optimal Solution

```
Plan length: 6
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 optimal Solution

```
Plan length: 9
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 optimal Solution

```
Plan length: 12
Load(C1, P1, SFO)
```

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

Un-informed Planning Search Results

Here are the metrics from running three non-heuristic search algorithms on the three problems.

	Air Cargo Problem 1			
	Expansions	**Goal Tests**	**Time (s)**	**Plan length**
breadth first search	**43**	**56**	**0.103**	**6**
depth first graph search	**21**	**22**	**0.030**	**21**
uniform cost search	**55**	**57**	**0.072**	**6**

	Air Cargo Problem 2			
	Expansions	**Goal Tests**	**Time (s)**	**Plan length**
breadth first search	**3343**	**4609**	**19.582**	**9**
depth first graph search	**624**	**625**	**4.874**	**619**
uniform cost search	**4853**	**4855**	**62.093**	**9**

	Air Cargo Problem 3			
	Expansions	**Goal Tests**	**Time (s)**	**Plan length**
breadth first search	**14663**	**18098**	**144.467**	**12**
depth first graph search	**408**	**409**	**2.444**	**392**
uniform cost search	**18223**	**18225**	**528.180**	**12**

The data shows that on these simple problems breadth-first-search and uniform-cost-search both produce the optimal solution with BFS taking the least amount of time and expanding less nodes.

Depth-first-graph-search takes significantly less time (and expands significantly less nodes) than the other search algorithms but produces massively sub-optimal solutions.

Automatic Heuristic Planning Search Results

Here are the metrics from running A* search with three different heuristics on the three problems.
H_1 is not a true heuristic - it always returns a value of 1
Ignore preconditions simply counts the number of outstanding goals
Planning Graph Level Sum uses a planning graph to estimate the number of actions required to meet the goals.

	Air Cargo Problem 1			
	Expansions	Goal Tests	Time (s)	Plan length
A* search with h_1	55	57	0.109	6
A* search with h_ignore_pre	41	43	0.129	6
A* search with h_pg_levelsum	11	13	2.122	6

	Air Cargo Problem 2			
	Expansions	Goal Tests	Time (s)	Plan length
A* search with h_1	4853	4855	61.550	9
A* search with h_ignore_pre	1506	1508	20.925	9
A* search with h_pg_levelsum	86	88	217.124	9

	Air Cargo Problem 3			
	Expansions	Goal Tests	Time (s)	Plan length
A* search with h_1	18223	18225	537.286	12
A* search with h_ignore_pre	5118	5120	122.449	12
A* search with h_pg_levelsum	414	416	1510.956	12

All three heuristics used with A* search produced the optimal solution for each problem with a big trade off between time and space requirements. The ignore pre-conditions heuristic[1] was the quickest and expanded a reasonable number of nodes. The level sum heuristic took significantly longer but expanded far fewer nodes because it uses a planning graph to search the planning space not the state space.

[1] S. Russell and P Norvig. Artificial Intelligence A Modern Approach. 3rd Ed. P 382-383