Heuristic Analysis For An Air Cargo Planner

Uirá Caiado

Given that human planning is the act of approaching goals ([2]), a planning agent should find a sequence of actions for transforming a given state into a state which fulfills a predefined set of goals. Problem-solving agents are used to solving similar problems. So, why not use these algorithms to solve planning tasks?

Because, as explained by [1], these algorithms require a good domain-specific heuristic to perform well. Also, in many real world applications, the atomic representation of states used by problem-solving techniques might not be feasible. On the other hand, a planning agent uses a collection of variables to represent the states of the environment, drastically reducing its state space. Also, even though a heuristic is still necessary, one can use a function derived from a relaxed version of the problem at hand.

In this project, we solved deterministic logistics planning problems for an Air Cargo transport system using a planning search agent with automatically generated heuristics. The first task is to transport the cargo $C1$ and $C2$ from the airports SFO and JFK to the airports JFK and SFO, respectively, using two airplanes, $P1$ and $P2$. Defining the problems in classical Planning Domain Definition Language (PDDL), we can write:

$$\begin{aligned}\text{Init}(&\text{At}(C1, SFO) \land \text{At}(C2, JFK)\\ &\land \text{At}(P1, SFO) \land \text{At}(P2, JFK)\\ &\land \text{Cargo}(C1) \land \text{Cargo}(C2)\\ &\land \text{Plane}(P1) \land \text{Plane}(P2)\\ &\land \text{Airport}(JFK) \land \text{Airport}(SFO))\\ \text{Goal}(&\text{At}(C1, JFK) \land \text{At}(C2, SFO))\end{aligned}$$

The second problem is transporting the cargo $C1$, $C2$ and $C3$ from the airports SFO, JFK and ATL to the airports JFK, SFO and SFO, respectively, using three airplanes, $P1$, $P2$ and $P3$.

$$\begin{aligned}\text{Init}(&\text{At}(C1, SFO) \land \text{At}(C2, JFK) \land \text{At}(C3, ATL)\\ &\land \text{At}(P1, SFO) \land \text{At}(P2, JFK) \land \text{At}(P3, ATL)\\ &\land \text{Cargo}(C1) \land \text{Cargo}(C2) \land \text{Cargo}(C3)\\ &\land \text{Plane}(P1) \land \text{Plane}(P2) \land \text{Plane}(P3)\\ &\land \text{Airport}(JFK) \land \text{Airport}(SFO) \land \text{Airport}(ATL))\\ \text{Goal}(&\text{At}(C1, JFK) \land \text{At}(C2, SFO) \land \text{At}(C3, SFO))\end{aligned}$$

The last problem is transporting the cargo $C1$, $C2$, $C3$ and $C4$ from the airports SFO, JFK, ATL and ORD to the airports JFK, JFK, SFO and SFO, respectively, using the airplanes $P1$ and $P2$.

$$\begin{aligned}\text{Init}(&\text{At}(C1, SFO) \land \text{At}(C2, JFK) \land \text{At}(C3, ATL) \land \text{At}(C4, ORD)\\ &\land \text{At}(P1, SFO) \land \text{At}(P2, JFK)\\ &\land \text{Cargo}(C1) \land \text{Cargo}(C2) \land \text{Cargo}(C3) \land \text{Cargo}(C4)\\ &\land \text{Plane}(P1) \land \text{Plane}(P2)\\ &\land \text{Airport}(JFK) \land \text{Airport}(SFO) \land \text{Airport}(ATL) \land \text{Airport}(ORD))\\ \text{Goal}(&\text{At}(C1, JFK) \land \text{At}(C3, JFK) \land \text{At}(C2, SFO) \land \text{At}(C4, SFO))\end{aligned}$$

In the table 1 are presented different non-heuristic search[1] methods that were tested to try to find an optimal solution to the air cargo logistics problems.

[1] Source: `http://cs.lmu.edu/~ray/notes/usearch/`

Non-Heuristic Search	Problem	Optimality	Goal Tests	Time (s)	Expansions	Plan Length
breadth first	1	Yes	56	0.06	43	6
breadth first	2	Yes	4609	21.98	3343	9
breadth first	3	Yes	18098	144.07	14663	12
breadth first tree	1	Yes	1459	1.52	1458	6
breadth first tree	2	No	-	> 600	-	-
breadth first tree	3	No	-	> 600	-	-
depth first graph	1	No	13	0.01	12	12
depth first graph	2	No	1670	12.34	1669	1444
depth first graph	3	No	593	2.86	592	571
depth limited	1	No	271	0.13	101	50
depth limited	2	No	-	> 600	-	-
depth limited	3	No	-	> 600	-	-
uniform cost	1	Yes	57	0.06	55	6
uniform cost	2	Yes	4854	21.17	4852	9
uniform cost	3	Yes	18237	99.01	18235	12

Table 1: Non-Heuristics Search performances

As can be seen, just the *breadth first search* and *uniform cost search* were able to find an optimal solution to the three problems. The first one was slightly faster than the second one and also used less node expasions and goal tests to find the solutions. The *breadth first tree search* and *depth limited search* have taken more than 10 minutes to solve the problemns 2 and 3 and were terminated before find a solution. The optimal solutions to these problems are described in the table 2.

Problem 1	Problem 2	Problem 3
Plan length 6	**Plan length 9**	**Plan length 12**
Load($C2, P2, JFK$)	Load($C2, P2, JFK$)	Load($C2, P2, JFK$)
Load($C1, P1, SFO$)	Load($C1, P1, SFO$)	Load($C1, P1, SFO$)
Fly($P2, JFK, SFO$)	Load($C3, P3, ATL$)	Fly($P2, JFK, ORD$)
Unload($C2, P2, SFO$)	Fly($P2, JFK, SFO$)	Load($C4, P2, ORD$)
Fly($P1, SFO, JFK$)	Unload($C2, P2, SFO$)	Fly($P1, SFO, ATL$)
Unload($C1, P1, JFK$)	Fly($P1, SFO, JFK$)	Load($C3, P1, ATL$)
	Unload($C1, P1, JFK$)	Fly($P1, ATL, JFK$)
	Fly($P3, ATL, SFO$)	Unload($C1, P1, JFK$)
	Unload($C3, P3, SFO$)	Unload($C3, P1, JFK$)
		Fly($P2, ORD, SFO$)
		Unload($C2, P2, SFO$)
		Unload($C4, P2, SFO$)

Table 2: Optimal Solutions

Neither the search methods tested are expected to be much efficient without using a heuristic. In the table 3 is presented the results to the tests performed using the algorithm A^* in conjunctions with two heuristic search strategies: *ignore preconditions* and *levelsum*.

As explained by [1], both heuristics are derived by defining a relaxed version of the problems, easier to solve. The cost of the solutions to this other problems becomes the heuristic for the original ones. Using *ignore preconditions*, all action becomes applicable in every state, implying that any goal fluent is achievable in one step (it is unsolvable otherwise). *Level sum* is derived using a data

A^* Heuristic Search	Problem	Optimality	Goal Tests	Time (s)	Expansions	Plan Length
Ignore preconditions	1	Yes	43	0.03	41	6
Ignore preconditions	2	Yes	1452	3.98	1450	9
Ignore preconditions	3	Yes	5042	14.35	5040	12
Levelsum	1	Yes	48	0.59	46	6
Levelsum	2	Yes	2897	247.71	2895	9
Levelsum	3	Yes	11395	1538.87	11393	12

Table 3: Heuristics Search performances

structure called planning graph that is divided into levels of states and actions. Each level is a step towards the solution of the problem. Then, the heuristic consists of counting the number of levels to achieve the goal.

Both heuristics were able to solve all the problems, as presented in the results. However, the *Levelsum* heuristic took much longer to solve the problems 2 and 3, as well as almost the double of node expansions and goals tests. With a clear advantage of the A^* Search using *ignore preconditions* heuristic in larger state spaces, when compared to the other methods tested, this algorithm is recommended.

References

[1] Stuart Russel and Peter Norvig. *Artificial Intelligence: A Modern Approach, 2003.*

[2] Mariya Ushshaque and Deependra Pandey. A study on artificial intelligence planning. *International Journal of Research in Engineering and Technology*, 20015.