Planning Historical Developments Research Review

July 18, 2017

Planning problems

	Plan
Problem1	Load(C1,P1,SFO) Load(C2,P2,JFK) Fly(P1,SFO,JFK) Fly(P2,JFK, SFO) Unload(C1,P1,JFK) Unload(C2,P2,SFO)
Problem2	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(C1,P1,JFK) Unload(C2,P2,SFO) Unload(C3,P3,SFO)
Problem3	Load(C1,P1,SFO) Load(C2,P2,JFK) Load(C3,P1,ATL) Load(C4,P2,ORD) Fly(P1,SFO,JFK) Fly(P2,JFK, ORD) Fly(P2,ORD, SFO) Fly(P1,ATL,JFK) Unload(C1,P1,JFK) Unload(C4,P2,SFO) Unload(C2,P2,SFO) Unload(C3,P1,JFK)

Results of search

	Expan.	Goal test	New Nodes	Plan len.	Time elapsed(sec)
Problem 1					
Breadth First Search	43	56	180	6	0.038
Depth First Graph Search	21	22	84	20	0.0142
Uniform Cost Search	55	57	224	6	0.0306
A* with h_ignore_precond.	41	43	170	6	0.047
A* with h_pg_levelsum	39	41	158	6	0.831
Problem 2					
Breadth First Search	3343	4609	30509	9	15.83
Depth First Graph Search	624	625	5602	619	3.193
Uniform Cost Search	4853	4855	44041	9	13.66
A* with h_ignore_precond.	1405	1452	13303	9	4.85
A* with h_pg_levelsum	1129	1131	10232	9	292.78
Problem 3					
Breadth First Search	14663	18098	129631	12	120.97
Depth First Graph Search	408	409	3364	392	2.09
Uniform Cost Search	18223	18225	159618	12	62.22
A* with h_ignore_precond.	5040	5042	44944	12	19.31
A* with h_pg_levelsum	2026	2028	17933	12	1038.45

Search Strategies discussion

The above table shows the results after applying the searches on all three problems in the project. Here all searches allowed to run with maximum of 600 second. Few are allowed more than that, which is recorded above.
Among the uninformed searches here, the breadth-first search consistently outperformed both depth-first graph and uniform cost searches. As explained in lectures, breadth first and cheapest-cost search are complete and optimal whereas depth-first search is not. The breadth first search has advantage on implementation which performs a goal test on the next frontier node before expanding its children. All of the breadth first searches performed return optimal plan. For the case of Depth first search, we know it is neither complete nor optimal. But Depth first graph search is complete but not optimal. Here it find the solution which is not optimal. It gives solution fast with large path length which is because it simply explores nodes without considering other better node.
The level sum heuristic A* search look slower runtimes than A* precondition because it need to construct complete PlanningGraph for the problem and which could also be computationally expensive.
According to the results obtained in this analysis, the best performacne among all the search choices is A* with preconditions heuristic. This algorithms perform well in time and also it reduces the amount of nodes to be expanded and hence the search space[1].

References

[1] Stuart J. Russell, Peter Norvig (2010), Artificial Intelligence: A Modern Approach (3rd Edition).