Planning Historical Developments Research Review

July 19, 2017

Huristic Analysis

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	0.034
Uniform Cost Search	55	57	224	6	0.039
Greedy Best First Graph Search	7	9	28	6	0.0051
A* search h_1	55	57	224	6	0.043
A* with h_ignore_precond	41	43	170	6	0.043

	Expan.	Goal test	New Nodes	Plan len.	Time elapsed(sec)
Problem 3					
Breadth First Search	3343	4609	40960	9	18.68
Uniform Cost Search	4853	4855	58964	6	15.99
Greedy Best First Graph Search	998	1000	11976	21	3.38
A* search h_1	4853	4855	58964	9	16.26
A* with h_ignore_precond.	1450	1452	17906	9	5.70

	Expan.	Goal test	New Nodes	Plan len.	Time elapsed(sec)
Problem 2					
Breadth First Search	414663	18096	209742	12	176.73
Uniform Cost Search	18236	18238	256119	12	82.28
Greedy Best First Graph Search	5623	5625	79884	26	25.53
A* search h_1	18236	18238	22256119	12	82.32
A* with h_ignore_precond.	5037	5039	73767	12	26.11

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.

1. 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.

2. 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.

3. 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.

4. The level sum heuristic *A* search* look slower runtimes than *A* precondition* because it need to construct complete Planning Graph for the problem and which could also be computationally expensive.

5. **Best result**: According to the results obtained in this analysis, the best performacne among all the search choices is *A* with preconditions heuristic*. In case of problem 2, *Greedy Best First Graph Search* out performs the *A* with h_ignore_precond*. 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).