Heuristic Analysis

For Planning Search Agent

Prepared by Xinlu Tu
For Artificial Intelligence Nanodegree, Udacity
September 20th , 2017

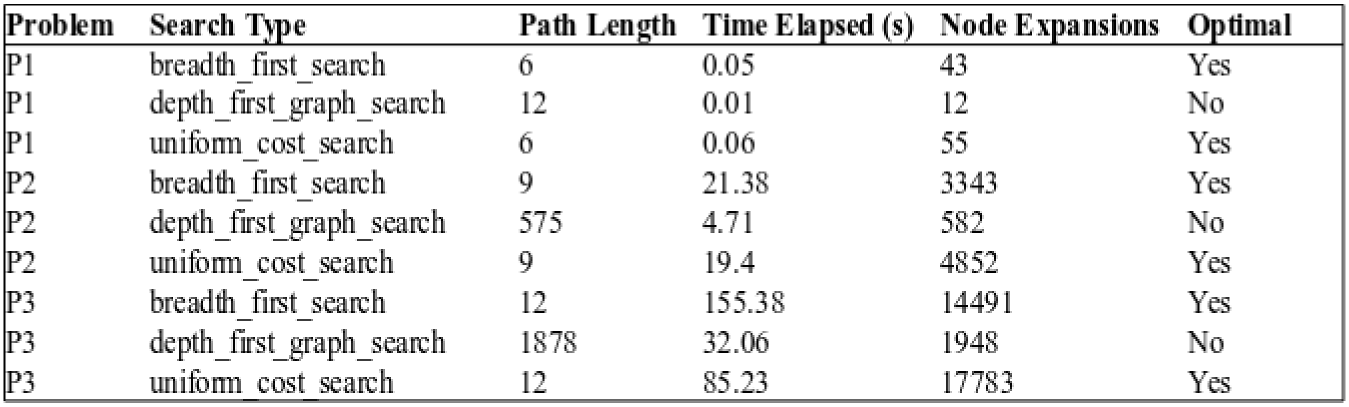
In this project, a planning search agent is implemented to solve deterministic logistics planning problems for Air Cargo transport system. The problems are defined in classical (Planning Domain Definition Language). Planning graph and A* search are implemented to compare with uninformed planning algorithms.

The three problems are:

```
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))
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))
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))
```

Uninformed Non-heuristic Search

Breadth First Search, Depth First Search, and Uniform Cost Search are selected for comparing in three problems. The table below is the result.



Problem	Search Type	Path Length	Time Elapsed (s)	Node Expansions	Optimal
P1	breadth_first_search	6	0.05	43	Yes
P1	depth_first_graph_search	12	0.01	12	No
P1	uniform_cost_search	6	0.06	55	Yes
P2	breadth_first_search	9	21.38	3343	Yes
P2	depth_first_graph_search	575	4.71	582	No
P2	uniform_cost_search	9	19.4	4852	Yes
P3	breadth_first_search	12	155.38	14491	Yes
P3	depth_first_graph_search	1878	32.06	1948	No
P3	uniform_cost_search	12	85.23	17783	Yes

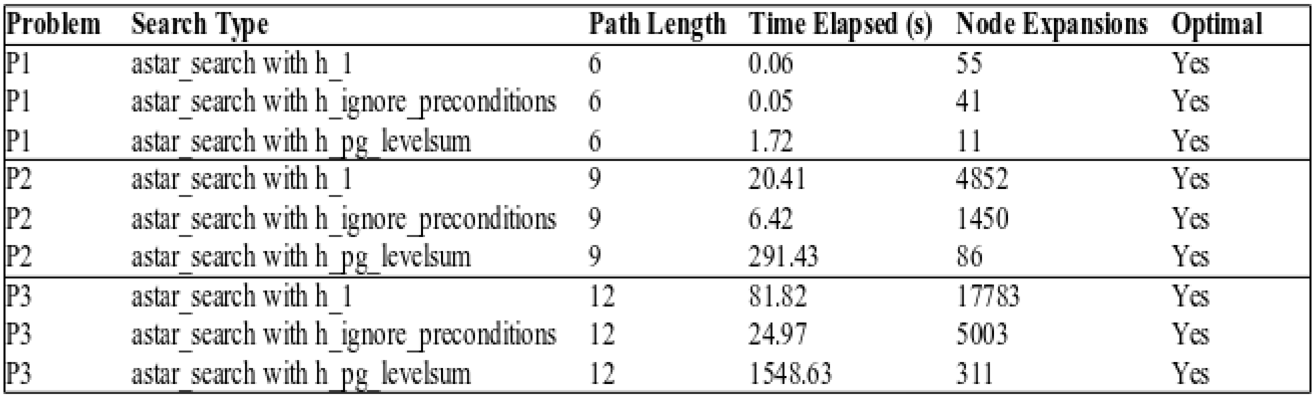
In Problem 1, time cost : DFS < BFS < UCS; node expansions: DFS < BFS < UCS. Although DFS has the lowest time cost and node expansions, it is not optimal. Thus, BFS seems a better choice for P1.

In Problem 2, time cost : DFS < UCS < BFS; node expansions: DFS < BFS < UCS. Although DFS has the lowest time cost and node expansions, it is not optimal. Thus, BFS seems a better choice for P2 because the time elapsed does not differ much between UCS and BFS but BFS has 1000 less node expansions than UCS.

In Problem 3, time cost : DFS < UCS < BFS; node expansions: DFS < BFS < UCS. Although DFS has the lowest time cost and node expansions, it is not optimal. Thus, UCS seems a better choice for P3 because the node expansions do not differ much between UCS and BFS but UCS has much less time cost than BFS.

Informed Heuristic Search

A* with h1 heuristic, A* with ignore preconditions heuristic, and A* with level sum heuristic are selected for comparing in three problems. The table below is the result.



Problem	Search Type	Path Length	Time Elapsed (s)	Node Expansions	Optimal
P1	astar_search with h_1	6	0.06	55	Yes
P1	astar_search with h_ignore_preconditions	6	0.05	41	Yes
P1	astar_search with h_pg_levelsum	6	1.72	11	Yes
P2	astar_search with h_1	9	20.41	4852	Yes
P2	astar_search with h_ignore_preconditions	9	6.42	1450	Yes
P2	astar_search with h_pg_levelsum	9	291.43	86	Yes
P3	astar_search with h_1	12	81.82	17783	Yes
P3	astar_search with h_ignore_preconditions	12	24.97	5003	Yes
P3	astar_search with h_pg_levelsum	12	1548.63	311	Yes

In Problem 1, time cost : h_ignore_preconditions < h_1 < h_pg_levelsum; node expansions: h_pg_levelsum < h_ignore_preconditions < h_1. All of them are optimal. Thus, h_ignore_preconditions seems to be a better choice for P1 as it is the fastest and with relatively less node expansions.

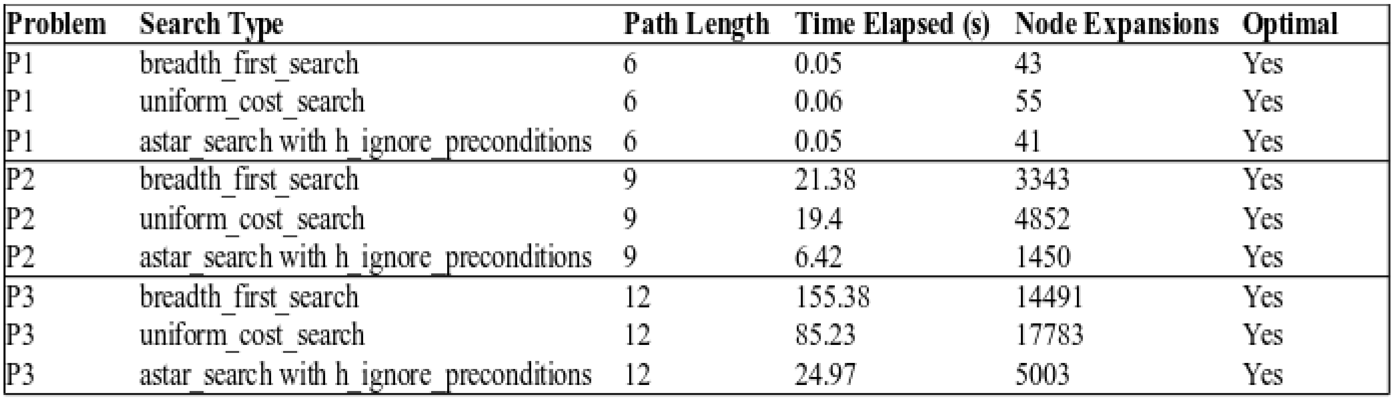
In Problem 2, time cost : h_ignore_preconditions < h_1 < h_pg_levelsum; node expansions: h_pg_levelsum < h_ignore_preconditions < h_1. All of them are optimal. Although h_pg_levelsum has lowest nodes expansions, it takes longest time. Thus, h_ignore_preconditions seems to be a better choice for P2 as it is the fastest and with relatively less node expansions.

In Problem 3, time cost : h_ignore_preconditions < h_1 < h_pg_levelsum; node expansions: h_pg_levelsum < h_ignore_preconditions < h_1. All of them are optimal. Although h_pg_levelsum has lowest nodes expansions, it takes longest time. Thus, h_ignore_preconditions seems to be a better choice for P3 as it is the fastest and with relatively less node expansions.

A* with ignore preconditions heuristic is most optimal search type for all three problems.

Uninformed v.s. Informed Heuristic Search

Breadth First Search, Uniform Cost Search, and A* with ignore preconditions heuristic are selected to compare with three problems because they perform good with less time and memory cost. The table below is the result.



Problem	Search Type	Path Length	Time Elapsed (s)	Node Expansions	Optimal
P1	breadth_first_search	6	0.05	43	Yes
P1	uniform_cost_search	6	0.06	55	Yes
P1	astar_search with h_ignore_preconditions	6	0.05	41	Yes
P2	breadth_first_search	9	21.38	3343	Yes
P2	uniform_cost_search	9	19.4	4852	Yes
P2	astar_search with h_ignore_preconditions	9	6.42	1450	Yes
P3	breadth_first_search	12	155.38	14491	Yes
P3	uniform_cost_search	12	85.23	17783	Yes
P3	astar_search with h_ignore_preconditions	12	24.97	5003	Yes

A* with ignore preconditions heuristic is the best choice with lowest time cost and node expansions for all the three questions even compared with uninformed search.

The optimal plan given by A* with ignore preconditions heuristic is the following:

P1:

Load(C1, P1, SFO)

Fly(P1, SFO, JFK)

Unload(C1, P1, JFK)

Load(C2, P2, JFK)

Fly(P2, JFK, SFO)

Unload(C2, P2, SFO)

P2:

Load(C3, P3, ATL)

Fly(P3, ATL, SFO)

Unload(C3, P3, SFO)

Load(C2, P2, JFK)

Fly(P2, JFK, SFO)

Unload(C2, P2, SFO)

Load(C1, P1, SFO)

Fly(P1, SFO, JFK)

Unload(C1, P1, JFK)

```
P3:
Load(C2, P2, JFK)
Fly(P2, JFK, ORD)
Load(C4, P2, ORD)
Fly(P2, ORD, SFO)
Unload(C4, P2, SFO)
Load(C1, P1, SFO)
Fly(P1, SFO, ATL)
Load(C3, P1, ATL)
Fly(P1, ATL, JFK)
Unload(C3, P1, JFK)
Unload(C1, P1, JFK)
Unload(C2, P2, SFO)
```

Conclusion

Informed search strategy with heuristics performs better than uninformed search strategy. And A* with ignore preconditions heuristic performs the best in the Air Cargo problems. Informed search strategy with heuristic functions are beneficial because it is fast and use less memory.

For uninformed search strategy, there is no information to determine preference of one child over another, thus it will be less efficient than informed search strategy. Below is the detailed comparison of uninformed and informed search:

"**1. Uninformed Search –** This is also known as 'Blind Search'. This type of search means that they have no additional information about the states other than the information provided by the problem definition. This type of search can generate successor states and also make a distinction between a goal state and a non-goal state.

2. Informed Search – This type of search strategy is more advanced than uninformed search. It is also known as 'heuristic search'. This strategy has the ability to determine whether a non-goal state is much better than another non-goal state in arriving at the goal state effectively and efficiently, thus it uses heuristics based information on top of the information provided by the problem definition about the states in the state space." [1]

As a result, uninformed search uses extra problem-specific knowledge which is beyond the actual problem itself in finding the best solution. And this leads it to find better problem-specific solutions in an efficient manner when compared to the uninformed search.

Reference:

[1] "Artificial Intelligence – Part 14" *digit.lk*. Aug 1, 2010.