Heuristic Analysis

This project to solve deterministic logistics planning problems for an Air Cargo transport system using a planning search agent (AI: A Modern Approach by Norvig And Russel) given classical PDDL. The goal of this project run both uninformed non heuristic and domain independent heuristic with A* search and obtain an optimal and fastest solution for this problem.

All problems are in the Air Cargo domain. They have the same action schema.

- 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))
```

An optimal sequence of actions:

Problem1:	**Problem2:**	**Problem3:**

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

Optimal plan Length:6	**Optimal plan Length:9**	**Optimal plan Length:12**

Uninformed Non-heuristic Search Analysis:

The Uninformed non-heuristic search agent ran with Breadth first search (BFS), and Depth first search (DFS) and Uniform cost search (UCS) . Here results table.

Problem	**Search Type**	**Expansions**	**Goal Tests**	**New Nodes**	**Path Length**	**Execution Time (s)**	**Optimal**
P1	**BFS**	43	56	180	6	0.029	YES
P1	**DFS**	101	271	414	50	0.074	NO
P1	**UCS**	55	57	224	6	0.030	YES
P2	**BFS**	3343	4609	30509	9	2.369	YES
P2	**DFS**	624	625	5602	619	0.033	NO
P2	**UCS**	4843	4855	44041	9	0.038	YES
P3	**BFS**	14663	18098	129631	12	90.39	YES
P3	**DFS**	408	409	3364	392	1.53	NO
P3	**UCS**	18223	18225	159618	12	44.59	YES

Comparison between search agents for Uninformed Non-heuristic

- Node Expansion: Depth First Search (DFS) taking less memory compare to BFS and UCS, because expanding less nodes. UCS > BFS > DFS
- Execution Time: Depth First Search (DFS) is faster than BFS and UCS.
UCS > BFS > DFS
- Optimal: Based on Path length (P1->6, P2->9, P3->12) Breadth First Search (BFS) And Uniform cost search (UCS) is provide optimal action plan. DFS not provide optimal action plan based on path lengths 50,619,392 instead of 6,9,12, because it does not consider better node due to once reach the goal, then not look other nodes even though better nodes available.

Based on factor (memory, Time, Optimal) **Breadth First Search (BFS)** is recommended uninformed non-heuristic search agent,

Domain independent heuristic with A* Search Analysis:

The domain independent heuristic with A* search agent ran with

- Greedy_best_first_graph_search with h_1
- A * Search h_1
- A* Search h_ignore_preconditions
- A* Search h_pg_levelsum

Here results table.

Problem	**Search Type**	**Expansions**	**Goal Tests**	**New Nodes**	**Path Length**	**Execution Time (s)**	**Optimal**
P1	**Greedy_best_first_graph_search with h_1**	7	9	28	6	0.004	YES
P1	**A * Search h_1**	55	57	224	6	0.033	YES
P1	**A* Search h_ignore_preconditions**	41	43	170	6	0.032	YES
P1	**A* Search h_pg_levelsum**	11	13	50	6	0.038	YES

P2	**Greedy_best_first_graph_search with h_1**	998	1000	8982	9	2.07	YES
P2	**A * Search h_1**	4853	4855	44041	9	10.07	YES
P2	**A* Search h_ignore_preconditions**	1450	1452	13303	9	3.63	YES
P2	**A* Search h_pg_levelsum**	86	88	841	9	29.32	YES
P3	**Greedy_best_first_graph_search with h_1**	5577	5579	49141	21	13.80	NO
P3	**A * Search h_1**	18223	18225	159618	12	44.70	YES
P3	**A* Search h_ignore_preconditions**	5040	5042	44994	12	14.39	YES
P3	**A* Search h_pg_levelsum**	328	330	3032	12	145.62	YES

Comparison between search agents for domain independent heuristic with A* search

- Node Expansion: A* Search h_pg_levelsum taking less memory compare to Greedy_best_first_graph_search with h_1, A * Search h_1 and A* Search h_ignore_preconditions, because expanding less nodes.

  A * Search h_1 > A* Search h_ignore_preconditions > Greedy_best_first_graph_search > A* Search h_pg_levelsum

- Execution Time: A* Search h_pg_levelsum faster compare to Greedy_best_first_graph_search with h_1, A * Search h_1 and A* Search h_ignore_preconditions, because expanding less nodes.

  A * Search h_1 > A* Search h_ignore_preconditions > Greedy_best_first_graph_search > A* Search h_pg_levelsum

- Optimal: Based on Path length (P1->6, P2->9, P3->12) A * Search h_1 and A* Search h_ignore_preconditions and A* Search h_pg_levelsum is provide optimal action plan. Greedy_best_first_graph_search with h_1 not provide optimal action plan based on path lengths 6,9,21 instead of 6,9,12.

Based on factor (memory, Time, Optimal) **A* Search h_pg_levelsum** is recommended domain independent heuristic with A* search agent. This heuristic Search Agents perform better as the problem complexity increased.

Search Agents Comparison Between Individual Problems.

Problem1:

Problem	Search Type	Expansions	Goal Tests	New Nodes	Path Length	Execution Time (s)	Optimal
P1	**BFS**	43	56	180	6	0.029	YES
P1	**DFS**	101	271	414	50	0.074	NO
P1	**UCS**	55	57	224	6	0.030	YES
P1	**Greedy_best_first_graph_search with h_1**	7	9	28	6	0.004	YES
P1	**A * Search h_1**	55	57	224	6	0.033	YES
P1	**A* Search h_ignore_preconditions**	41	43	170	6	0.032	YES
P1	**A* Search h_pg_levelsum**	11	13	50	6	0.038	YES

Greedy_best_first_graph_search with h_1 using less memory, execution time and optimal solution compare to across all search agents in problem use case. Less complexity problems recommended **Greedy_best_first_graph_search with h_1** for Problem1**.**

Problem2:

Problem	Search Type	Expansions	Goal Tests	New Nodes	Path Length	Execution Time (s)	Optimal
P2	**BFS**	3343	4609	30509	9	2.369	YES
P2	**DFS**	624	625	5602	619	0.033	NO
P2	**UCS**	4843	4855	44041	9	0.038	YES
P2	**Greedy_best_first_graph_search with h_1**	998	1000	8982	9	2.07	YES
P2	**A * Search h_1**	4853	4855	44041	9	10.07	YES
P2	**A* Search h_ignore_preconditions**	1450	1452	13303	9	3.63	YES
P2	**A* Search h_pg_levelsum**	86	88	841	9	29.32	YES

A* Search h_ignore_preconditions using execution time and optimal solution compare to across all search agents in problem use case. Medium complexity problems recommended **A* Search h_ignore_preconditions** for Problem2**,** (Note:if memory is constrain A* Search h_pg_levelsum using less memory)

Problem3:

Problem	Search Type	Expansions	Goal Tests	New Nodes	Path Length	Execution Time (s)	Optimal
P3	**BFS**	14663	18098	129631	12	90.39	YES
P3	**DFS**	408	409	3364	392	1.53	NO
P3	**UCS**	18223	18225	159618	12	44.59	YES
P3	**Greedy_best_first_graph_search with h_1**	5577	5579	49141	21	13.80	NO
P3	**A * Search h_1**	18223	18225	159618	12	44.70	YES
P3	**A* Search h_ignore_preconditions**	5040	5042	44994	12	14.39	YES
P3	**A* Search h_pg_levelsum**	328	330	3032	12	145.62	YES

A* Search h_pg_levelsum using less memory, execution time and optimal solution compare to across all search agents in problem use case. More complexity problems recommended **A* Search h_pg_levelsum** for Problem3**.**

Conclusion:

Based on results domain independent heuristic with A* search agent good for complexity problems, and less complexity problems good for uninformed search agent. **A* Search h_pg_levelsum** using less memory, execution time and optimal solution compare to across all search agents.

Reference:

Stuart J. Russel, Peter Norvig, Artificial Intelligence A Modern Approach (3rd Edition)