PROJECT 3: PLANNING AGENT

Objective: The objective of this project is to implement an AI planning search agent to solve Air Cargo logistics planning problem. We use planning graph and domain-independent heuristics and compare the results with uninformed heuristic search methods [BFS, DFS, etc].

Problems: The following is the problem description of the Air cargo logistics problem:

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))
```

Metrics for non-heuristic planning search

For Un-informed search strategies are tested with following 7 algorithms and performance is assessed in terms of (1) execution time in seconds (2) memory usage in terms of node expansions required (3) whether solution is optimal or not. For Problem 2 and 3, data is not collected for BF Tree Search, Depth Limited Search and Recursive BFS as the execution time exceeded 10 minutes. Best of 3 performance indicators are highlighted as below:

Problem 1: Metrics for problem 1 with uninformed search strategies:

Search Strategy	Optimal	Path Length	Node Expansion	Execution Time	Goal Test
BF Search	Yes	6	43	0.03	56
BF Tree Search	Yes	6	1458	1.05	1459
DF Graph Search	No	12	12	0.01	13
Depth Limited Search	No	50	101	0.10	271
Uniform Cost Search	Yes	6	55	0.044	57
Recursive BFS	Yes	6	4229	3.09	4230
Greedy BF Graph Search	Yes	6	7	0.01	9

Problem 2 : Metrics for problem 2 with uninformed search strategies:

Search Strategy	Optimal	Path Length	Node Expansion	Execution Time	Goal Test
BF Search	Yes	9	3401	16.6	4672
BF Tree Search	-	-	-	-	-
DF Graph Search	No	346	350	1.77	351
Depth Limited Search	-	-	-	-	-
Uniform Cost Search	Yes	9	4761	53.53	4763
Recursive BFS	-	-	-	-	-
Greedy BF Graph Search	Yes	9	550	3.5	552

Problem 3 : Metrics for problem 3 with uninformed search strategies:

Search Strategy	Optimal	Path Length	Node Expansion	Execution Time	Goal Test
BF Search	Yes	12	14629	138.9	18072
BF Tree Search	-				
DF Graph Search	No	2200	2269	35.84	2270
Depth Limited Search	-	-	-	-	-
Uniform Cost Search	Yes	12	17783	493.3	17785
Recursive BFS	-	-	-	-	-
Greedy BF Graph Search	No	22	4031	88.4	4033

Analysis:

BFS and Uniform Cost Search are the only two un-informed search strategies which provide optimal action plan under 10 minutes. DF Graph search is the fastest (0.01 sec) and uses less memory in terms of performance, however it is not optimal solution. **BFS is optimal solution** as it performs faster and uses less memory. For problem 2 and 3, **Greedy BF Graph search** is best as solution is optimal for problem 1 and 2. For problem 3, path length is 22 which is better than DF Graph search path length 2200.

Heuristics Search Strategies (A*)

In informed search strategies, we compare performance of A* with 3 heuristic functions and assess them based on execution time, memory usage and optimality.

Problem 1:

Search Strategy	Optimal	Path Length	Node Expansion	Execution Time	Goal Test
H1 Heuristic	Yes	6	55	0.045	57
Ignore preconditions	Yes	6	41	0.052	43
Level Sum	Yes	6	11	3.81	13

Problem 2:

Time exceeded for Level Sum Heuristic more than 10 minutes and hence no data collected

Search Strategy	Optimal	Path Length	Node Expansion	Execution Time	Goal Test
H1 Heuristic	Yes	9	4761	49.30	4763
Ignore preconditions	Yes	9	1506	16.11	1508
Level Sum	-	-	-	-	-

Problem 3:

Time exceeded for Level Sum Heuristic more than 10 minutes and hence no data collected

Search Strategy	Optimal	Path Length	Node Expansion	Execution Time	Goal Test
H1 Heuristic	Yes	12	17783	472.17	17785
Ignore preconditions	Yes	12	5081	104.04	5083
Level Sum	-	-	-	-	-

Analysis:

Only H1 and Ignore preconditions heuristics returned within 10 minutes and as per table **Ignore preconditions** is the fastest

Informed Vs Un-Informed Search Comparison:

1. Un-informed: DFS Graph search is faster and uses less memory than Uniform cost search.
2. Informed: A* search with ignore pre-conditions uses less memory and faster than others.
3. Between (1) and (2) A* search with Ignore preconditions performs better for our problem

Problem 1:

Search Strategy	Path Length	Node Expansion	Execution Time	Goal Test
BFS Search	6	43	0.03	56
A* Search with Ignore preconditions	6	41	0.052	43

Problem 2:

Search Strategy	Path Length	Node Expansion	Execution Time	Goal Test
BFS Search	9	3401	16.6	4672
A* Search with Ignore preconditions	9	1506	**16.11**	1508

Problem 3:

Search Strategy	Path Length	Node Expansion	Execution Time	Goal Test
BFS Search	12	14629	138.9	18072
A* Search with Ignore preconditions	12	5081	**104.04**	5083

Conclusion:

1. **A* search with Ignore preconditions** is best overall for Air Cargo problem.
2. Informed Search strategies with custom heuristics perform better overall for optimal plan.
3. Informed search strategies perform better in terms of execution time and memory usage.