AIND – Implement a Planning Search (Heuristic Analysis)
By Param Shah

Problem 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)
```

- Optimal Plan (6 actions)

```
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P1, SFO, JFK)
Fly(P2, JFK, SFO)
Unload(C1, P1, JFK)
Unload(C2, P2, SFO)
```

➢ Uninformed Search Strategies Analysis:

Search Strategy	Optimality	Execution Time (s)	Path Length	Node Expansions
Breadth First Search	YES	0.029	6	43
Breadth First Tree Search	YES	0.83	6	1458
Depth First Graph Search	NO	0.009	12	12
Depth Limited Search	NO	0.09	50	101
Uniform Cost search	YES	0.035	6	55
Recursive Best first search	YES	2.379	6	4229
Greedy best first graph search	YES	0.0062	6	7

➢ Informed (Heuristic) Search Strategies Analysis:

Search Strategy	Optimality	Execution Time (s)	Path Length	Node Expansions
A* Search with h1 heuristic	YES	0.0340	6	55
A* Search with ignore preconditions heuristic	YES	0.0349	6	41
A* Search with Level Sum heuristic	YES	0.7974	6	11

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))
```

➢ Optimal Plan (9 actions)

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

```
Unload(C2, P2, SFO)
Unload(C1, P1, JFK)
```

- Uninformed Search Strategies Analysis:

Search Strategy	Optimality	Execution Time (s)	Path Length	Node Expansions
Breadth First Search	YES	7.2752	9	3401
Breadth First Tree Search	-	-	-	-
Depth First Graph Search	NO	1.327	346	350
Depth Limited Search	-	-	-	-
Uniform Cost search	YES	9.95545	9	4761
Recursive Best first search	-	-	-	-
Greedy best first graph search	YES	1.18532	9	550

- Informed (Heuristic) Search Strategies Analysis:

Search Strategy	Optimality	Execution Time (s)	Path Length	Node Expansions
A* Search with h1 heuristic	YES	10.033	9	4761
A* Search with ignore preconditions heuristic	YES	3.8381	9	1450
A* Search with Level Sum heuristic	YES	139.8651	9	86

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))
```

- Optimal Plan (12 actions)

```
Load(C1, P1, SFO)
Fly(P1, SFO, ATL)
Load(C3, P1, ATL)
Fly(P1, ATL, JFK)
```

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

- Uninformed Search Strategies Analysis:

Search Strategy	Optimality	Execution Time (s)	Path Length	Node Expansions
Breadth First Search	YES	35.04334	12	14491
Breadth First Tree Search	-	-	-	-
Depth First Graph Search	NO	16.713441	1878	1948
Depth Limited Search	-	-	-	-
Uniform Cost search	YES	42.267	12	17783
Recursive Best first search	-	-	-	-
Greedy best first graph search	No	9.809678	22	4031

- Informed (Heuristic) Search Strategies Analysis:

Search Strategy	Optimality	Execution Time (s)	Path Length	Node Expansions
A* Search with h1 heuristic	YES	41.51231	12	17783
A* Search with ignore preconditions heuristic	YES	14.3173	12	5003
A* Search with Level Sum heuristic	-	-	-	-

Search Strategies discussion:

- It is observed that in all the problems Breadth First Search, Depth First Graph Search and Uniform Cost Search, all give us a solution. But it is seen that Breadth First Search out and Uniform Cost search always gives us an Optimal Solution while Depth First Graph Search does not.

- Breadth First search always finds the shortest path first and because of that it is always able to find optimal solutions in a reasonable amount of time.

  But memory requirements are a huge problem and many other algorithms can solve the same problem with a lesser memory requirement. After Memory Requirement issue, time taken to solve a problem is also large. It's time complexity is $O(b^d)$. So this searching technique should not be used for problems with exponential complexities. But BFS performs extremely well for small instances
  [AIMA book, Peter Norvig, 3rd edition, pg 83]

- The Uniform Cost Search does find optimal solutions, but the computational time for the search is large. Hence making it slower than the Breadth First Search.

  Uniform Cost Search does not care about the number of path steps but cares only about their total cost. So this may take it into an infinite loop if there is a path with an infinite sequence of zero cost actions. Hence more computational time.
  [AIMA book, Peter Norvig, 3rd edition, pg 85]

- Depth first graph search finds a solution very quickly, but this solution is not optimal because it simply explores the nodes that take it as deep as possible in the graph (it can even ignore the goal if it is one step away).

  There is no advantage of Depth First Graph Search over the Breadth First Search. Depth First Search is not Optimal because it ignores the goal nodes even if they are right next to the current node being explored. But Depth

First TREE search is a little better because it has very little space complexity, O(bm); where b = branching factor and m = maximum depth of the tree.
[AIMA book, Peter Norvig, 3rd edition, pg 87]

- Hence it is clear that Breadth First Search is the best searching strategy in the category of Uninformed Search Strategies.

- In case of Informed or Heuristic Search strategies, the A* search with ignore preconditions performed very well as the complexity of the planning problem increased.
[AIMA book, Peter Norvig, 3rd edition, pg 376]

- The A* search with level sum heuristic did not perform well as the complexity of planning problem increased probably because of the heuristic being too complex and hence increasing computational time.
[AIMA book, Peter Norvig, 3rd edition, pg 382]

Uninformed vs Informed Searching

- Problem 1:

Search Strategy	Optimality	Execution Time (s)	Path Length	Node Expansions
Breadth First Search	YES	0.029	6	43
A* Search with ignore preconditions heuristic	YES	0.0349	6	41

- Problem 2:

Search Strategy	Optimality	Execution Time (s)	Path Length	Node Expansions
Breadth First Search	YES	7.2752	9	3401

A* Search with ignore preconditions heuristic	YES	3.8381	9	1450

- Problem 3:

Search Strategy	Optimality	Execution Time (s)	Path Length	Node Expansions
Breadth First Search	YES	35.04334	12	14491
A* Search with ignore preconditions heuristic	YES	14.3173	12	5003

It is clear from above that A* Search with ignore preconditions heuristic would be out best option for solving the air-cargo problem optimally, consuming minimum memory and executing in less time.

Conclusion:

The above analysis clearly indicates the advantage of using Informed Search Strategies with custom heuristics. Even so, Uninformed Search Strategies like Breath First search is very useful when the problem is relatively simpler having fewer literals.

One more thing that should be noted here is that in case of Informed Search Strategies, it is much better to select a cheaper to calculate heuristic in place of a smarter but computationally costly to calculate heuristic.

[AIND, Building A Game Playing Agent]