Alexey Ganin

Planning Search Heuristic Analysis

The purpose of this paper is to compare and contrast performance of heuristic and non-heuristic searches in the domain of air cargo planning problems.

Problems definitions and optimal plans examples:

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))
```

Optimal plan for this task is **six steps** long, for example:

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

Optimal plan for this task is **nine steps** long, for example:

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

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 for this task is **12 steps** long, for example:

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

Non-heuristic search result metrics and analysis:

Search Results for Problem 1:

Search method	Optimality	Path length	Time elapsed (sec)	Node expansions
Breadth First Search	Yes	6	0.08	43
Breadth First Tree Search	Yes	6	1.56	1458
Depth First Graph Search	No	20	0.02	21
Depth Limited Search	No	50	0.12	101
Uniform Cost Search	Yes	6	0.05	55
Recursive Best First Search	Yes	6	4.13	4229
Greedy Best First Graph Search	**Yes**	**6**	**0.01**	**7**

Greedy best first graph search demonstrated the best performance: shortest path length, lowest timing and node expansions.

Search Results for Problem 2:

Search method	Optimality	Path length	Time elapsed (sec)	Node expansions
Breadth First Search	**Yes**	**9**	**14.54**	**3343**
Breadth First Tree Search	-	-	-	-
Depth First Graph Search	No	619	5.57	624
Depth Limited Search	No	50	1672	222719
Uniform Cost Search	Yes	9	21.35	4853
Recursive Best First Search	-	-	-	-
Greedy Best First Graph Search	No	21	4.62	998

Breadth first graph search demonstrated the best performance: shortest path length, lowest timing and node expansions.

Breadth First Tree Search and Recursive Best First Search were terminated after two hours of pending execution.

Search Results for Problem 3:

Search method	Optimality	Path length	Time elapsed (sec)	Node expansions
Breadth First Search	**Yes**	**12**	**70.54**	**14663**
Breadth First Tree Search	-	-	-	-
Depth First Graph Search	No	392	3.13	408
Depth Limited Search	-	-	-	-
Uniform Cost Search	Yes	12	89.53	18223
Recursive Best First Search	-	-	-	-
Greedy Best First Graph Search	No	22	27.24	5578

Breadth first graph search demonstrated the best performance: shortest path length, lowest timing and node expansions.

Breadth First Tree Search, Depth Limited Search, and Recursive Best First Search were terminated after an hour of pending execution.

Two search algorithms demonstrated consistent ability to find optimal solution: Breadth First Search and Uniform Cost Search. Their disadvantages are high execution time and number of explored nodes. This data is in line with information provided in the book "Artificial Intelligence: A Modern Approach" 3rd edition section 3.4.7:

Criterion	Breadth-First	Uniform-Cost	Depth-First	Depth-Limited	Iterative Deepening	Bidirectional (if applicable)
Complete?	Yes[a]	Yes[a,b]	No	No	Yes[a]	Yes[a,d]
Time	$O(b^d)$	$O(b^{1+\lfloor C^*/\epsilon \rfloor})$	$O(b^m)$	$O(b^\ell)$	$O(b^d)$	$O(b^{d/2})$
Space	$O(b^d)$	$O(b^{1+\lfloor C^*/\epsilon \rfloor})$	$O(bm)$	$O(b\ell)$	$O(bd)$	$O(b^{d/2})$
Optimal?	Yes[c]	Yes	No	No	Yes[c]	Yes[c,d]

Figure 3.21 Evaluation of tree-search strategies. b is the branching factor; d is the depth of the shallowest solution; m is the maximum depth of the search tree; l is the depth limit. Superscript caveats are as follows: [a] complete if b is finite; [b] complete if step costs $\geq \epsilon$ for positive ϵ; [c] optimal if step costs are all identical; [d] if both directions use breadth-first search.

Greedy Best First Graph Search provided sub-optimal path length, but it is significantly better than optimal algorithms in terms of execution time and node expansions. Consequently, it can be preferred in cases when computation space and time is limited.

Heuristic search result metrics:

Search Results for Problem 1:

Search method	Optimality	Path length	Time elapsed (sec)	Node expansions
A* Search with h1 heuristic	Yes	6	0.06	55
A* Search with ignore preconditions heuristic	Yes	6	0.07	41
A* Search with level sum heuristic	Yes	6	1.63	11

Search Results for Problem 2:

Search method	Optimality	Path length	Time elapsed (sec)	Node expansions
A* Search with h1 heuristic	Yes	9	30.6	4853
A* Search with ignore preconditions heuristic	Yes	9	11.32	1450
A* Search with level sum heuristic	Yes	9	479	86

Search Results for Problem 3:

Search method	Optimality	Path length	Time elapsed (sec)	Node expansions
A* Search with h1 heuristic	Yes	12	124	18223
A* Search with ignore preconditions heuristic	Yes	12	38	5040
A* Search with level sum heuristic	Yes	12	2225	325

All heuristic search algorithms were able to find optimal solution, however timing and space consumption varied a lot. A* search with ignore preconditions heuristic is consistently demonstrates better timing. A* with level sum heuristic performs better in terms of the number of expanded nodes.

Conclusion

Data provided above demonstrates the benefits of usage heuristic methods in the domain of air cargo planning problems. For problems 2 and 3 **A* search with ignore preconditions heuristic** performed significantly better in terms of time and space consumption.