Planning Search Heuristic Analysis

In this analysis, we will compare and contrast the performance of uninformed non-heuristic search result metrics and informed heuristic search result metrics for the three cargo planning problems. These metrics will be analyzed in terms of (1) finding the optimal path length, (2) speed, and (3) memory usage. I have listed the three criteria in order of importance (in my opinion) because we want our search to be:

- Economical - cargo should be moved as quickly/cheaply as possible.
- Quick - the plans should be created in the least amount of time as possible.
- Efficient - the plans should be created using the least amount of memory as possible.

Outline of Problem

All three problems have the same action schema, presented below, but their initial states and goals differ. An optimal plan for each problem is presented to the right of each problem's initial states and goals.

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:

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))

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

Problem 2:

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))

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

Problem 3:

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))

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

Uninformed Non-Heuristic Search Result Metrics

Note: Strategies that exceed 10 minutes to execute will have a dashes in their rows; this time limit was set by Udacity. 'Time Elapsed' is measured in seconds. The best values are bolded and the best strategy for each problem is italicized.

Problem 1

Search Strategy	Optimal	Path Length	Node Expansions	Goal Tests	New Nodes	Time Elapsed
Breadth First Search	**Yes**	**6**	43	56	180	0.08
Breadth First Tree Search	**Yes**	**6**	1458	1459	5960	1.52
Depth First Graph Search	No	20	21	22	84	0.03
Depth Limited Search	No	50	101	271	414	0.14
Uniform Cost Search	**Yes**	**6**	55	57	224	0.06
Recursive Best First Search	**Yes**	**6**	4229	4230	17023	5.30
Greedy Best First Graph Search	**Yes**	**6**	**7**	**9**	**28**	**0.01**

Problem 2

Search Strategy	Optimal	Path Length	Node Expansions	Goal Tests	New Nodes	Time Elapsed
Breadth First Search	**Yes**	**9**	3343	4609	30509	18.82
Breadth First Tree Search	-	-	-	-	-	-
Depth First Graph Search	No	619	624	625	5602	4.25
Depth Limited Search	-	-	-	-	-	-
Uniform Cost Search	**Yes**	**9**	4761	4763	43206	15.47
Recursive Best First Search	-	-	-	-	-	-
Greedy Best First Graph Search	**Yes**	**9**	**550**	**552**	**4950**	**1.72**

Problem 3

Search Strategy	Optimal	Path Length	Node Expansions	Goal Tests	New Nodes	Time Elapsed
Breadth First Search	**Yes**	**12**	14491	17947	128184	125.68
Breadth First Tree Search	-	-	-	-	-	-
Depth First Graph Search	No	2014	**2099**	**2100**	**17558**	27.55
Depth Limited Search	-	-	-	-	-	-
Uniform Cost Search	**Yes**	**12**	17783	17785	155920	68.20
Recursive Best First Search	-	-	-	-	-	-
Greedy Best First Graph Search	No	22	4031	4033	35794	**17.06**

Summary of Results

As we can see from these results, the time and memory required to complete each search strategy increases exponentially with the path length. This increase is so significant that three strategies (Breadth First Tree Search, Depth Limited Search, and Recursive Best First Search) fail to complete within the allotted ten minutes for problems 2 and 3.

As the complexity of the task grows, there are fewer strategies that are able to find an optimal solution. For the first problem, five of the seven strategies find the optimal solution, but this number reduces to two for the third and most complex problem.

Based on the performance of each strategy for all three problems, I will argue that **Uniform Cost Search** is the best strategy. Although it was outperformed by Greedy Best First Graph Search in the first two problems, Greedy failed to find an optimal path for problem 3. Given that I stated finding the optimal path as the most important metric, it is for this reason that I think Uniform is better overall. It is worth noting that Breadth First Search also found an optimal path in each problem, but it always took longer than Uniform to find a path, thus it is an inferior strategy.

Informed Heuristic Search Result Metrics

Problem 1

Search Strategy	Optimal	Path Length	Node Expansions	Goal Tests	New Nodes	Time Elapsed
A* Search with h1 heuristic	**Yes**	**6**	55	57	224	**0.05**
A* Search with Ignore Preconditions heuristic	**Yes**	**6**	41	43	170	**0.05**
A* Search with Level Sum heuristic	**Yes**	**6**	**11**	**13**	**50**	1.22

Problem 2

Search Strategy	Optimal	Path Length	Node Expansions	Goal Tests	New Nodes	Time Elapsed
A* Search with h1 heuristic	**Yes**	**9**	4761	4763	43206	17.47
A* Search with Ignore Preconditions heuristic	**Yes**	**9**	1450	1452	13303	**5.49**
A* Search with Level Sum heuristic	**Yes**	**9**	**86**	**88**	**841**	237.09

Problem 3

Search Strategy	Optimal	Path Length	Node Expansions	Goal Tests	New Nodes	Time Elapsed
A* Search with h1 heuristic	**Yes**	**12**	17783	17785	155920	77.99
A* Search with Ignore Preconditions heuristic	**Yes**	**12**	5003	5005	44586	**24.45**
A* Search with Level Sum heuristic	**-**	**-**	**-**	**-**	**-**	-

Summary of Results

A* Search with Ignore Preconditions heuristic is the best strategy for this section. It was able to find an optimal path for each problem and always completed its search in, at least, the fastest time. For the first two problems, A* Search with Level Sum heuristic was by far the most memory efficient strategy, but this came at the expense of time. Given its speed, it was not able to find a solution for the final problem in under ten minutes, so its search was terminated.

Comparing the Best Strategies

Problem	Search Strategy	Optimal	Path Length	Node Expansions	Goal Tests	New Nodes	Time Elapsed
1	**Uniform Cost Search**	**Yes**	**6**	55	57	224	0.06
1	***A* Search with Ignore Preconditions heuristic***	**Yes**	**6**	**41**	**43**	**170**	**0.05**
2	**Uniform Cost Search**	**Yes**	**9**	4761	4763	43206	15.47
2	***A* Search with Ignore Preconditions heuristic***	**Yes**	**9**	**1450**	**1452**	**13303**	**5.49**
3	**Uniform Cost Search**	**Yes**	**12**	17783	17785	155920	68.20
3	***A* Search with Ignore Preconditions heuristic***	**Yes**	**12**	**5003**	**5005**	**44586**	**24.45**

It is easy to see the benefits of using an informed heuristic search strategy. In every measure, A* Search with Ignore Preconditions heuristic outperforms Uniform Cost Search, so despite the extra work that is needed to create an informed heuristic search strategy, the improved performance is a just reward.