Search methods analysis by Yevgen Nerush

This analysis contains brief overview of the different search methods including non-heuristic and heuristic ones, their performance metrics along with benefits and limitations. Each search method is researched on the Air Cargo Problem of small (1), medium (2) and large sizes (3).

There are five performance metrics in each search method evaluation:

- Expansions: the number of times the frontier is expanded by calling `PlanningProblem`'s `actions` function
- Goal Tests: the number of nodes verified for the goal match by calling `PlanningProblem`'s `goal_test` function
- New Nodes: the number of nodes added to the graph during the search by calling `PlanningProblem`'s `result` function
- Plan length: size of a list of the actions consecutive execution of which leads to an optimal solution (a solution with all sub-goals being satisfied)
- Execution time: the number of seconds a search algorithm takes to search for an optimal solution

Air Cargo Problem 1

From the optimal solution and the performance metrics listed in the Figure 1 and the Table 1 respectively, we can conclude that all but "Depth first graph search" and "Depth limited search" search methods find out the optimal solution of 6 actions. Taking into account that "Breadth first search", "Breadth first tree search" and "Recursive best first search with h1" come up with the optimal solution of 6 actions, they are not optimal in terms of number of expansions, goal tests and new nodes (EGN for convenience). "Breadth first search", "Depth first graph search", "Uniform cost search", "A* search with h1 heuristic" and "A* search with h_ignore_preconditions heuristic" search methods have slightly better performance characteristics, but they are still not the best ones in terms of performance metrics. The fastest heuristic search is "A* search with h_ignore_preconditions heuristic", but the most optimal heuristic search in terms of EGN is "A* search with levelsum heuristic". It is approximately 5 times cheaper in terms of EGN than "A* search with h_ignore_preconditions heuristic", but its performance is 20 times slower. Both "Greedy best first graph search with h1" and "A* search with levelsum heuristic" come up with the optimal solution with the fewest number of EGN with the only one difference between them: "Greedy best first graph search with h1 " is almost 30 times faster than "A* search with levelsum heuristic."

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

`Figure 1. The optimal solution for the Problem 1.`

	Expansions	Goal Tests	New Nodes	Plan length	Execution time in seconds
1. Breadth first search	43	56	180	6	0.035

2. Breadth first tree search	1458	1459	5960	6	6.019
3. Depth first graph search	21	22	84	20	0.098
4. Depth limited search	101	271	414	50	0.480
5. Uniform cost search	55	57	224	6	0.241
6. Recursive best first search with h1	4229	4230	17023	6	17.971
7. Greedy best first graph search with h1	7	9	28	6	0.027
8. A* search with h1 heuristic	55	57	224	6	0.251
9. A* search with h_ignore_preconditions heuristic	41	43	170	6	0.042
10. A* search with levelsum heuristic	7	9	28	6	0.900

Table 1. Performance metrics of the different search methods of Problem 1.

Air Cargo Problem 2

For the problem 2 we have different and more interesting performance characteristics of the search methods. Three search methods, such as "Breadth first tree search", "Depth limited search" and "Recursive best first search with h1", cannot search for the solution without violation of time constraint of 10 minutes. The fastest search method at this time is "Depth first graph search", it comes up with the solution in approximately 2 seconds, but optimality of the solution is the most non-optimal comparing with the other search methods. It simply returns the first found solution which satisfies all the sub-goals. Both "Breadth first search" and "Uniform cost search" non-heuristic searches come up with the optimal sequence of actions, but their EGN metrics are much worse than of "Greedy best first graph search with h1" search. "Greedy best first graph search with h1", on its turn, has one of the most non-optimal solutions for the problem 2. From the observed three heuristic searches, the most optimal in terms of EGN is the latest one, "A* search with levelsum heuristic". It solves the problem with only 77 expansions, 79 goal tests and only 760 new nodes. On the other hand, it has one of the slowest execution times and it is 10 times slower than "A* search with h_ignore_preconditions heuristic".

```
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)
```

Figure 2. The optimal solution for the Problem 2.

	Expansions	Goal Tests	New Nodes	Plan length	Execution time in seconds
1. Breadth first search	3346	4612	30534	9	81.659
2. Breadth first tree search	?	?	?	?	> 600
3. Depth first graph search	107	108	959	105	2.255
4. Depth limited search	?	?	?	?	> 600
5. Uniform cost search	4853	4855	44041	9	140.891
6. Recursive best first search with h1	?	?	?	?	> 600
7. Greedy best first graph search with h1	998	1000	8982	21	26.120
8. A* search with h1 heuristic	4853	4855	44041	9	162.157
9. A* search with h_ignore_preconditions heuristic	1506	1508	13820	9	13.943
10. A* search with levelsum heuristic	77	79	760	9	129.232

`Table 2. Performance metrics of the different search methods of Problem 2.`

Air Cargo Problem 3

The most interesting statistical information comes with problem 3. There are only three search methods which come up with the optimal solution of 12 actions in reasonable amount of time. Not surprisingly, "Depth first graph search" has the most non-optimal solution of 392 actions because of its recursive nature and optimality detection approach. Non-heuristic "Greedy best first graph search with h1" search algorithm is better than mentioned above, bot worse than "A* search with h_ignore_preconditions heuristic", which seems to be the most optimal in terms of EGN and time among the others. "A* search with levelsum heuristic" is again (like in the problem 2) 10 times slower than "A* search with h_ignore_preconditions heuristic" method, but it has much better characteristics for expansions, goal tests and new nodes.

```
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)
```

Figure 3. The optimal solution for the Problem 3.

	Expansions	Goal Tests	New Nodes	Plan length	Execution time in seconds
1. Breadth first search	14663	18098	129631	12	457.617
2. Breadth first tree search	?	?	?	?	> 600
3. Depth first graph search	408	409	3364	392	11.076
4. Depth limited search	?	?	?	?	> 600
5. Uniform cost search	?	?	?	?	> 600
6. Recursive best first search with h1	?	?	?	?	> 600
7. Greedy best first graph search with h1	5614	5616	49429	22	259.745
8. A* search with h1 heuristic	?	?	?	?	> 600
9. A* search with h_ignore_preconditions heuristic	5118	5120	45650	12	93.625
10. A* search with levelsum heuristic	403	405	3708	12	950.926

Table 3. Performance metrics of the different search methods of Problem 3.

Conclusion

The most fastest search method for solving problem 1 is non-heuristic search called "Greedy best first graph search with h1". It finds the most optimal solution of 6 steps in 0.027 seconds and has the lowest number of expansions, goal tests and new nodes, which are 7, 9 and 28 respectively. For the problem 2 the the choice depends on context of the problem. For the problems, where exploration costs are higher than cost of time, "A* search with levelsum heuristic" is the most optimal search method. On the other hand, e.g. for the real time systems, where exploration costs are cheap but time cost is the most expensive one, the "A* search with h_ignore_preconditions heuristic" search is the most optimal one. The same logic is applicable for the problem 3: if time is not an issue and it is cheaper to think twice, the "A* search with levelsum heuristic" search is the most optimal one, because it finds the plan of 12 actions in 403 expansions, 405 goal tests and 3708 new nodes. It is worth to mention, that "A* search with levelsum heuristic" is slow because of runtime complexity of its heuristic function. The algorithm of

choice is "A* search with h_ignore_preconditions heuristic", because it provides the optimal solution in reasonable time and has relatively not high exploration costs, such as expansions, goal tests and new nodes.

Search results

1. Breadth first search

Air Cargo Problem 1

- Expansions: 43
- Goal Tests: 56
- New Nodes: 180
- Plan length: 6
- Execution time: 0.03547631500987336 seconds

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

Air Cargo Problem 2

- Expansions: 3346
- Goal Tests: 4612
- New Nodes: 30534
- Plan length: 9
- Execution time: 81.65928123100002 seconds

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

Air Cargo Problem 3

- Expansions: 14663
- Goal Tests: 18098
- New Nodes: 129631
- Plan length: 12

- Execution time: 457.6173511959996 seconds

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

2. Breadth first tree search

Air Cargo Problem 1

- Expansions: 1458
- Goal Tests: 1459
- New Nodes: 5960
- Plan length: 6
- Execution time: 6.019270433000202 seconds

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

Air Cargo Problem 2

Does not terminate in 10 minutes.

Air Cargo Problem 3

Does not terminate in 10 minutes.

3. Depth first graph search

Air Cargo Problem 1

- Expansions: 21

- Goal Tests: 22
- New Nodes: 84
- Plan length: 20
- Execution time: 0.09816352300003928 seconds

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

Air Cargo Problem 2

- Expansions: 107
- Goal Tests: 108
- New Nodes: 959
- Plan length: 105
- Execution time: 2.255084823999823 seconds

```
Fly(P3, ATL, JFK)
Fly(P2, JFK, ATL)
Fly(P3, JFK, SFO)
Fly(P2, ATL, SFO)
Fly(P1, SFO, ATL)
Fly(P3, SFO, ATL)
Fly(P1, ATL, JFK)
Fly(P3, ATL, JFK)
...
Unload(C2, P3, SFO)
```

Air Cargo Problem 3

- Expansions: 408

- Goal Tests: 409
- New Nodes: 3364
- Plan length: 392
- Execution time: 11.076630202999695 seconds

```
Fly(P1, SFO, ORD)
Fly(P2, JFK, ORD)
Fly(P1, ORD, ATL)
Fly(P2, ORD, ATL)
Fly(P1, ATL, JFK)
Fly(P2, ATL, SFO)
...
Unload(C3, P1, JFK)
```

4. Depth limited search

Air Cargo Problem 1

- Expansions: 101
- Goal Tests: 271
- New Nodes: 414
- Plan length: 50
- Execution time: 0.48083225499794935 seconds

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

Air Cargo Problem 2

Does not terminate in 10 minutes.

Air Cargo Problem 3

Does not terminate in 10 minutes.

5. Uniform cost search

Air Cargo Problem 1

- Expansions: 55
- Goal Tests: 57
- New Nodes: 224
- Plan length: 6
- Execution time: 0.2418411139951786 seconds

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

Air Cargo Problem 2

- Expansions: 4853
- Goal Tests: 4855
- New Nodes: 44041
- Plan length: 9
- Execution time: 140.8912663539959 seconds

```
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)
```

Air Cargo Problem 3

Does not terminate in 10 minutes.

6. Recursive best first search with h1 heuristic function

Air Cargo Problem 1

- Expansions: 4229
- Goal Tests: 4230
- New Nodes: 17023
- Plan length: 6
- Execution time: 17.971775871999853 seconds

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

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

Air Cargo Problem 2

Does not terminate in 10 minutes.

Air Cargo Problem 3

Does not terminate in 10 minutes.

7. Greedy best first graph search with h1

Air Cargo Problem 1

- Expansions: 7
- Goal Tests: 9
- New Nodes: 28
- Plan length: 6
- Execution time: 0.02786448599999858 seconds

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

Air Cargo Problem 2

- Expansions: 998
- Goal Tests: 1000
- New Nodes: 8982
- Plan length: 21
- Execution time: 26.12001982899983 seconds

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

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

Air Cargo Problem 3

- Expansions: 5614
- Goal Tests: 5616
- New Nodes: 49429
- Plan length: 22
- Execution time: 259.74592149100044 seconds

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

8. A* search with h1 heuristic function

Air Cargo Problem 1

- Expansions: 55
- Goal Tests: 57
- New Nodes: 224
- Plan length: 6
- Execution time: 0.25108054200245533 seconds

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

Air Cargo Problem 2

- Expansions: 4853
- Goal Tests: 4855
- New Nodes: 44041
- Plan length: 9
- Execution time: 162.15786768100224 seconds

```
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)
```

Air Cargo Problem 3

Does not terminate in 10 minutes.

9. A* search with h_ignore_preconditions

Air Cargo Problem 1

- Expansions: 41
- Goal Tests: 43
- New Nodes: 170
- Plan length: 6
- Execution time: 0.042065354995429516 seconds

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

Air Cargo Problem 2

- Expansions: 1506
- Goal Tests: 1508
- New Nodes: 13820
- Plan length: 9
- Execution time: 13.943244863010477 seconds

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

Air Cargo Problem 3

- Expansions: 5118
- Goal Tests: 5120
- New Nodes: 45650
- Plan length: 12
- Execution time: 93.62522890602122 seconds

```
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)
```

10. A* search with levelsum heuristic function

Air Cargo Problem 1

- Expansions: 7
- Goal Tests: 9
- New Nodes: 28
- Plan length: 6
- Execution time: 0.9002755659894319 seconds

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

Air Cargo Problem 2

- Expansions: 77
- Goal Tests: 79
- New Nodes: 760
- Plan length: 9
- Execution time: 129.23260724698775 seconds

```
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)
```

Air Cargo Problem 3

- Expansions: 403
- Goal Tests: 405
- New Nodes: 3708
- Plan length: 12
- Execution time: 950.9264681539935 seconds

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

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