Comparison of non-heuristic search result metrics

Looking at the metrics of the results of the non-heuristic searches it is possible to see that optimal plans are given by the breadth first and uniform cost search. A star search with h equal to 1 gives the same results of the uniform cost search because the first uses the path cost used by the latter and adds 1, so nothing changes in term of the priority queue management in the two algorithms. The breadth first compared to the uniform cost search gives slightly better results in term of node expansions, goal tests and new nodes added to the frontier, with the same order of magnitude. In contrast the uniform cost search has comparable performance in term of elapsed time in the Air Cargo problem 1 and 2 and is almost half in the Air Cargo problem 3.

Breadth first graph and uniform cost search are complete and optimal and so produce these similar expected results.

Below the results just described.

For Air Cargo Problem 1.

```
Solving Air Cargo Problem 1 using breadth_first_search...

Expansions   Goal Tests   New Nodes
    43          56         180

Plan length: 6  Time elapsed in seconds: 0.032884143999922344
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
Unload(C2, P2, SFO)
Fly(P1, SFO, JFK)
Unload(C1, P1, JFK)
_____________________________________________________________

Solving Air Cargo Problem 1 using uniform_cost_search...

Expansions   Goal Tests   New Nodes
    55          57         224

Plan length: 6  Time elapsed in seconds: 0.04050698999344604
…
```

For Air Cargo Problem 2.

```
Solving Air Cargo Problem 2 using breadth_first_search...

Expansions   Goal Tests   New Nodes
   3343        4609       30509

Plan length: 9  Time elapsed in seconds: 14.520892202000027
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)
_____________________________________________________________________

Solving Air Cargo Problem 2 using uniform_cost_search...

Expansions   Goal Tests   New Nodes
   4780        4782       43381

Plan length: 9  Time elapsed in seconds: 12.599010500999952
…
```

For Air Cargo Problem 3.

```
Solving Air Cargo Problem 3 using breadth_first_search...

Expansions   Goal Tests   New Nodes
  14663       18098       129631

Plan length: 12  Time elapsed in seconds: 109.66662711799995
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)
_____________________________________________________________________

Solving Air Cargo Problem 3 using uniform_cost_search...

Expansions   Goal Tests   New Nodes
  18151       18153       159038

Plan length: 12  Time elapsed in seconds: 57.38898486700009
…
```

Looking at the results of greedy best first with h equal to 1 and depth first search, an optimal plan is given only in the Air Cargo problem 1 with the greedy best first search. In this case there are the best performances among all the other uninformed search in term of node expansions, goal tests, new nodes and elapsed time. In the Air Cargo problem 2 and 3, there are not optimal plans but while the greedy best first search results in plans that are near to the optimal ones, the depth first search results in plans that are two order of magnitude larger than the optimal ones. The performances are mostly comparable except in the Air Cargo problem 3 where greedy best first search results in node expansions, goal tests, new nodes and elapsed time that are an order of magnitude greater than those of the depth first search.

Depth first and greedy best first search are not optimal but the latter in this case is more efficient than the first. The greedy best first search with h equal to 1 produces poor results because it doesn't use a heuristic. It works only in the Air Cargo problem 1 where the problem is small enough and the value 1 approximates an effective heuristic.

Below the results.

For Air Cargo Problem 1.

```
Solving Air Cargo Problem 1 using depth_first_graph_search...

Expansions   Goal Tests   New Nodes
    21          22          84

Plan length: 20  Time elapsed in seconds: 0.016027904999987186
…
_____________________________________________________________________________

Solving Air Cargo Problem 1 using greedy_best_first_graph_search with h_1...

Expansions   Goal Tests   New Nodes
    7           9           28

Plan length: 6  Time elapsed in seconds: 0.008927734997996595
…
```

For Air Cargo Problem 2.

```
Solving Air Cargo Problem 2 using depth_first_graph_search...

Expansions   Goal Tests   New Nodes
   624         625         5602

Plan length: 619  Time elapsed in seconds: 3.973980359000052
…
_____________________________________________________________________________

Solving Air Cargo Problem 2 using greedy_best_first_graph_search with h_1...

Expansions   Goal Tests   New Nodes
   598         600         5382

Plan length: 21  Time elapsed in seconds: 1.7289827660060837
…
```

For Air Cargo Problem 3.

```
Solving Air Cargo Problem 3 using depth_first_graph_search...

Expansions   Goal Tests   New Nodes
   408         409         3364

Plan length: 392  Time elapsed in seconds: 1.90146473599998
…
_____________________________________________________________________________
Solving Air Cargo Problem 3 using greedy_best_first_graph_search with h_1...

Expansions   Goal Tests   New Nodes
   5398        5400       47665

Plan length: 26  Time elapsed in seconds: 16.300526199999695
…
```

Looking at the results of the breadth first tree, depth limited and recursive best first with h equal to 1 the optimal plans are given by the results of the Air Cargo problem 1 with the breadth first tree and recursive best first search while the depth limited search gives a plan that is an order of magnitude greater than the optimal one. Regarding the Air Cargo problem 2 and 3 these three algorithms give no results during the observation (more than 10 minutes). These last mentioned three search algorithms don't work because:

- in the case of breadth first tree and depth limited search it is not kept track of the visited node and the goal could be never reached because nodes visited could be visited again many times.
- in the case of recursive best first search the effectiveness of the algorithm depends on the quality of the heuristic which in this case is always 1. It works only in the Air Cargo problem 1 where the problem is small enough and the value 1 approximates an effective heuristic.

Below the results.

For Air Cargo Problem 1.

```
Solving Air Cargo Problem 1 using breadth_first_tree_search...

Expansions   Goal Tests   New Nodes
   1458         1459         5960

Plan length: 6  Time elapsed in seconds: 0.974754063999967
…
_____________________________________________________________________________
Solving Air Cargo Problem 1 using depth_limited_search...

Expansions   Goal Tests   New Nodes
   101          271          414

Plan length: 50  Time elapsed in seconds: 0.10008098499997686
…
_____________________________________________________________________________
Solving Air Cargo Problem 1 using recursive_best_first_search with h_1...

Expansions   Goal Tests   New Nodes
   4229         4230        17023

Plan length: 6  Time elapsed in seconds: 3.266572089996771
…
```

Comparison of heuristic search result metrics

Looking at the metrics of the results of the heuristic searches the A star search with "ignore preconditions" and "level sum" heuristics result in optimal plans in all the three Air Cargo problems. The latter

heuristic search gives best results in terms of node expansions, goal tests and new nodes added to the frontier but takes more time than the first, of an order magnitude, because of the computation of the heuristic with the planning graph.
The heuristic searches outperform the non-heuristic searches in term of node expansions, goal tests and new nodes mostly when the problem become larger. The A star search with “level sum” heuristic take more time but is comparable with other searches, an order of magnitude greater in Air Cargo problem 3, because of the construction of the planning graph for the computation of the heuristic when a node is added to the frontier.

Below the results.

```
Solving Air Cargo Problem 1 using astar_search with h_ignore_preconditions...

Expansions   Goal Tests   New Nodes
    41          43         170

Plan length: 6  Time elapsed in seconds: 0.04372532400009277
…
_____________________________________________________________________________
Solving Air Cargo Problem 2 using astar_search with h_ignore_preconditions...

Expansions   Goal Tests   New Nodes
   1450        1452       13303

Plan length: 9  Time elapsed in seconds: 4.688471004000007
…
_____________________________________________________________________________
Solving Air Cargo Problem 3 using astar_search with h_ignore_preconditions...

Expansions   Goal Tests   New Nodes
   5038        5040       44926

Plan length: 12  Time elapsed in seconds: 17.604478988999972
…
_____________________________________________________________________________
Solving Air Cargo Problem 1 using astar_search with h_pg_levelsum...

Expansions   Goal Tests   New Nodes
    11          13          50

Plan length: 6  Time elapsed in seconds: 0.7378783200000498
…
_____________________________________________________________________________
Solving Air Cargo Problem 2 using astar_search with h_pg_levelsum...

Expansions   Goal Tests   New Nodes
    86          88         841

Plan length: 9  Time elapsed in seconds: 64.52407019100008
…
_____________________________________________________________________________
Solving Air Cargo Problem 3 using astar_search with h_pg_levelsum...

Expansions   Goal Tests   New Nodes
   314         316         2894

Plan length: 12  Time elapsed in seconds: 318.559158583
…
```