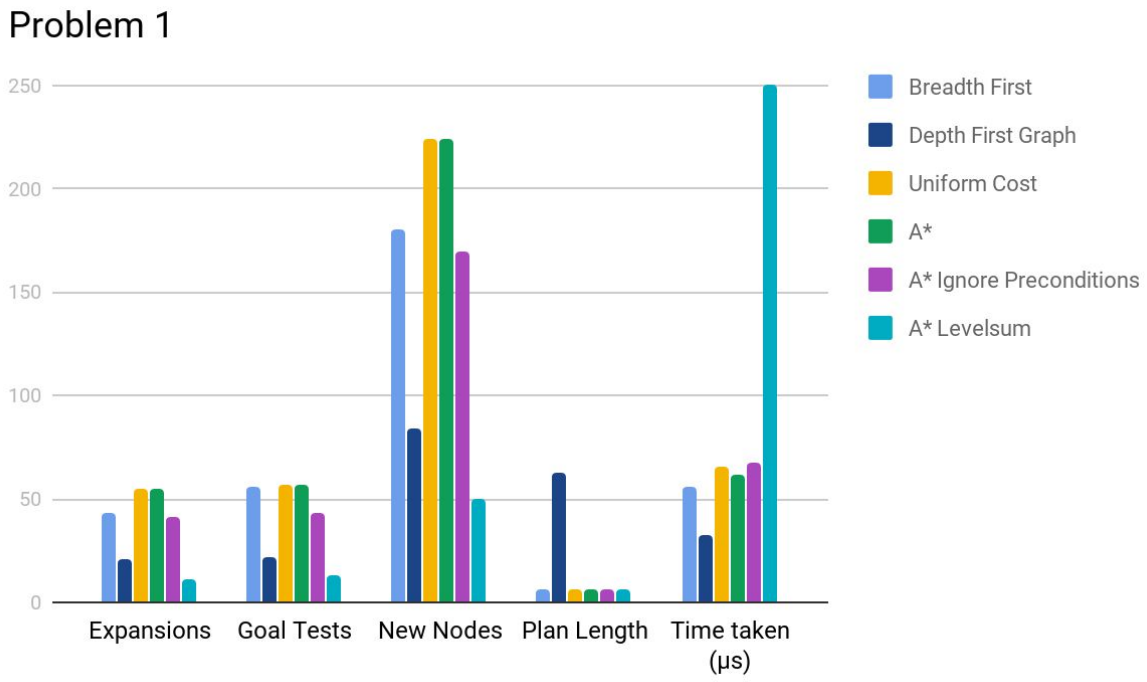
AIND-PLANNIG Written Analysis

Pulasthi Bandara

Analysis



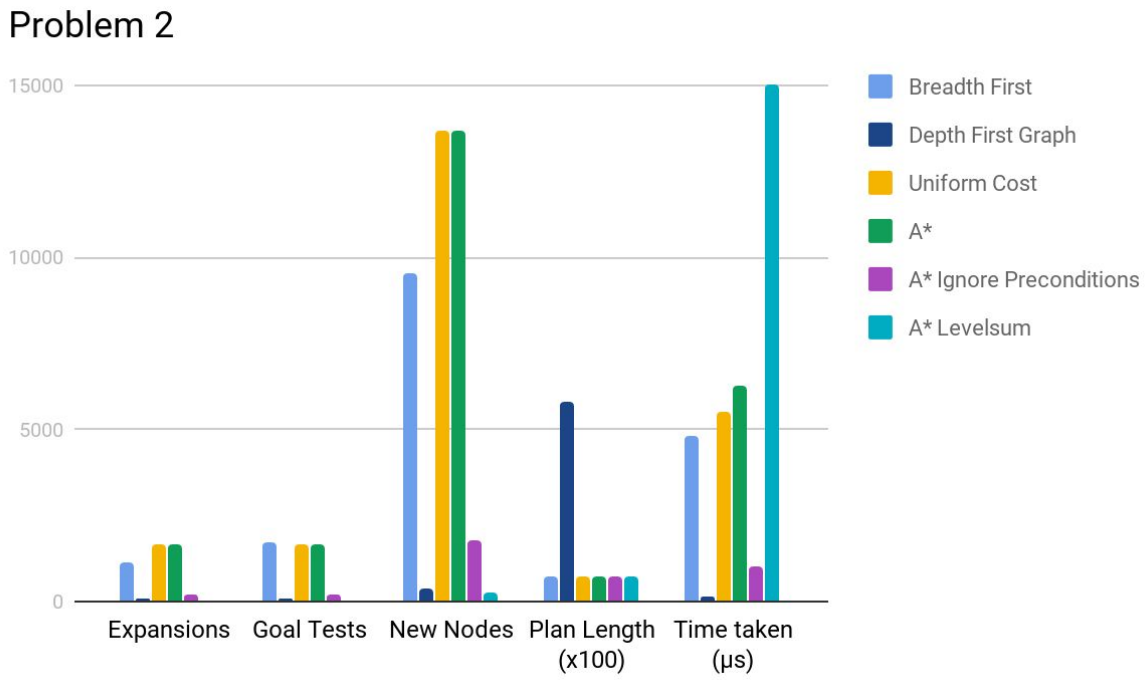
(Levelsum search took 2116.9063770212233 µs)

All algorithms considered, managed to find the optimal solution (6 steps) except for depth-first search(20 steps) which was 4 times more expensive. However when considering the time taken for execution, depth-first search executed the fastest just taking 32µs. Most of the other algorithms took as much as twice the the to finish with exception being the A* levelsome search. It took more than 2000µs to return a result. which is more than 40 times longer when compared to the average execution time of other services. When considering the number of expansions and new nodes, A*, A* ignore preconditions and breadth first search traversed twice as much nodes when compared with depth-first or level some algorithms. One notable observation is that for problem 1 uniform cost search and A*h1 search returned identical results for all metrics.

Optimal Path (6 steps)

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

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



(Levelsum search took 107899.55452084541 µs)

In problem 2 the results are consistent with the problem 1 result only more evident than the earlier observations. Even in problem 2 depth-first search failed to achieve the optimal search (Optimal being 7 steps and the depth-first result being 58 steps) which was 8 time more expensive (Problem 1 was only 4 times expensive for depth-first search). Similar to problem 1 depth-first search executed the fastest (.1s). One notable difference to observe is A* ignore preconditions executed faster (1s) than rest of the algorithms. A* Levelsum took 100s which is 40 times the average of other algorithms. When considering the nodes processed, breadth-first and A* processed around 40-times more nodes than the rest, where A* level-some processed the least nodes. Similar to Problem 1 Uniform Cost and A* h1 search has identical results.

Optimal Path (7 steps)

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

Problem 3

Expansions	Goal Tests	New Nodes	Steps	Time (mins)
14663	18098	129631	12	3
17783	17785	155920	12	12.6

I was only able to get results from breadth-first search and uniform cost search. Breadth first search took 3 minutes to return a value and uniform cost took 4 times more time (12 minutes). Depth first search ran more than 20mins, in which point I had to stop. However this does make a critical observation. Which is that, while depth first search runs really fast for less complex problems as the complexity of the problem increase it can potentially take even more time than breadth first search.

Optimal Path (12 Steps)

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

Non-heuristic search Analysis

Considering the three non-heuristic searches tested (breadth-first, depth-first and uniform cost), uniform cost search performed the worst processing most number of nodes and taking the longest time to process. Comparing the differences of breadth-first and depth-first, depth-first search executed the faster while processing the least amout of nodes. However the solution it achieved was not optimal. Bredth-first search managed to achive an optimal solution but consumed more memory while processing nodes and took longer to return a result. However on the last problem, which was more complex that the other two. Depth first search was unable to return a result within 10mins. Bredth-first search returned a result within 3mins while processing a considerable number of more nodes than problem 2 (10times more) However, available memory did not became a constraint.

According to Video lessons from AIND, depth-first search can be used in cases where the state space is large and available memory is a constraint. The results show while breadth-first search used considerably more memory while depth first kept it to a minimum.

Heuristic Search Analysis

The most notable difference between levelsum search and ignore-preconditions is that levelsum search is considerably slower than ignore-preconditions. In problem 2 levelsum took 100 times longer (100s vs 1s) to complete. However when considering the number node traversals and goal checks ignore-preconditions consumed more resources (10x more).

Concusion

Out of all the algorithms considered above ignore-preconditions and levelsum algorithms performed the best. However for very complex problems it is possible that levelsum may take a long time to complete. On the other hand ignore-preconditions may consume more memory when required to search deeper. Therefore the actual best algorithm is dependent on the resource availability. If not constrained by available memory ignore-preconditions is more desirable since it delivers optimal results fastest. However, when constrained by available memory or if execution time is not a constraint, a levelsum search is more desirable. This is backed by the discussion on “Norvig & Russell’s” discussion on levelsum where it states that levelsum algorithms works very well on practice for largely decomposable problems.

Raw Results

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

Expansions   Goal Tests   New Nodes
    43          56         180

Plan length: 6  Time elapsed in seconds: 0.055929404916241765
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
Load(C2, P2, SFO)
Fly(P1, SFO, JFK)
Load(C1, P1, JFK)


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: 0.032889571972191334
Fly(P1, SFO, JFK)
Fly(P2, JFK, SFO)
Load(C2, P1, JFK)
Fly(P1, JFK, SFO)
```

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

Solving Air Cargo Problem 2 using breadth_first_search...

```
Expansions   Goal Tests   New Nodes
   1125        1704        9548

Plan length: 7  Time elapsed in seconds: 4.828249841928482
Load(C2, P2, JFK)
Load(C3, P2, ATL)
Fly(P2, ATL, SFO)
Load(C1, P2, SFO)
Load(C1, P2, JFK)
Load(C2, P2, SFO)
Load(C3, P2, SFO)
```

Solving Air Cargo Problem 2 using depth_first_graph_search...

```
Expansions   Goal Tests   New Nodes
    62          63         397

Plan length: 58  Time elapsed in seconds: 0.1587238151114434
Fly(P2, ATL, SFO)
Fly(P1, SFO, ATL)
Fly(P2, SFO, JFK)
Fly(P1, ATL, JFK)
Fly(P2, JFK, ATL)
Load(C3, P2, ATL)
Fly(P2, ATL, SFO)
Fly(P1, JFK, ATL)
Fly(P2, SFO, JFK)
Fly(P1, ATL, SFO)
Load(C3, P2, JFK)
Fly(P1, SFO, JFK)
Fly(P2, JFK, ATL)
Fly(P1, JFK, ATL)
Fly(P2, ATL, SFO)
Load(C1, P2, SFO)
Fly(P1, ATL, SFO)
Fly(P2, SFO, ATL)
Fly(P1, SFO, JFK)
Fly(P2, ATL, JFK)
Load(C1, P2, JFK)
Fly(P2, JFK, ATL)
Load(C3, P1, JFK)
Fly(P2, ATL, SFO)
Fly(P1, JFK, ATL)
Fly(P2, SFO, JFK)
Fly(P1, ATL, SFO)
Load(C3, P1, SFO)
Fly(P1, SFO, ATL)
Fly(P2, JFK, ATL)
Fly(P1, ATL, JFK)
Fly(P2, ATL, SFO)
```

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

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

```
Expansions   Goal Tests   New Nodes
    55          57         224

Plan length: 6  Time elapsed in seconds: 0.06140073388814926
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P1, SFO, JFK)
Fly(P2, JFK, SFO)
Load(C1, P1, JFK)
Load(C2, P2, SFO)
```

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

```
Expansions   Goal Tests   New Nodes
   1643        1645       13692

Plan length: 7  Time elapsed in seconds: 6.246298179961741
Load(C3, P2, ATL)
Fly(P2, ATL, SFO)
Load(C1, P2, SFO)
Load(C2, P2, JFK)
Load(C3, P2, SFO)
Load(C1, P2, JFK)
Load(C2, P2, SFO)
```

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: 0.06736779795028269
Load(C1, P1, SFO)
Fly(P1, SFO, JFK)
Load(C1, P1, JFK)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
Load(C2, P2, SFO)
```

Solving Air Cargo Problem 2 using astar_search with h_ignore_preconditions...

```
Expansions   Goal Tests   New Nodes
   201         203         1774

Plan length: 7  Time elapsed in seconds: 1.0436042819637805
Load(C3, P2, ATL)
Fly(P2, ATL, SFO)
Load(C1, P2, SFO)
Load(C1, P2, JFK)
Load(C3, P2, SFO)
Load(C2, P2, JFK)
Load(C2, P2, SFO)
```

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: 2.1169063770212233
Load(C1, P1, SFO)
Fly(P1, SFO, JFK)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
Load(C1, P1, JFK)
Load(C2, P2, SFO)
```

Solving Air Cargo Problem 2 using astar_search with h_pg_levelsum...

```
Expansions   Goal Tests   New Nodes
    21          23         232

Plan length: 7  Time elapsed in seconds: 107.89955452084541
Load(C3, P2, ATL)
Fly(P2, ATL, SFO)
Load(C1, P2, SFO)
Load(C2, P2, JFK)
Load(C3, P2, SFO)
Load(C1, P2, JFK)
Load(C2, P2, SFO)
```

Solving Air Cargo Problem 1 using uniform_cost_search...

```
Expansions   Goal Tests   New Nodes
    55          57         224

Plan length: 6  Time elapsed in seconds: 0.06532905716449022
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P1, SFO, JFK)
Fly(P2, JFK, SFO)
Load(C1, P1, JFK)
Load(C2, P2, SFO)
```

Solving Air Cargo Problem 2 using uniform_cost_search...

```
Expansions   Goal Tests   New Nodes
   1643        1645       13692

Plan length: 7  Time elapsed in seconds: 5.509934653993696
Load(C3, P2, ATL)
Fly(P2, ATL, SFO)
Load(C1, P2, SFO)
Load(C2, P2, JFK)
Load(C3, P2, SFO)
Load(C1, P2, JFK)
Load(C2, P2, SFO)
```

Solving Air Cargo Problem 3 using breadth_first_search...

```
Expansions   Goal Tests   New Nodes
  14663       18098       129631

Plan length: 12  Time elapsed in seconds: 179.93737308098935
Load(C2, P2, JFK)
Load(C1, P1, SFO)
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
  17783       17785       155920

Plan length: 12  Time elapsed in seconds: 757.397472619079
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)
```