Heuristic Analysis of Planning Search

Optimal Sequence of Actions

Problem 1

Load(C2, P2, JFK)

Load(C1, P1, SFO)

Fly(P2, JFK, SFO)

UnLoad(C2, P2, SFO)

Fly(P1, SFO, JFK)

UnLoad(C1, P1, JFK)

Problem 2

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)

Problem 3

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)

Metrics for Non Heuristic Planning Searches

Method	Problem	Nodes Expanded	Goal Tests	New Nodes	Time	Optimal?
BFS	1	43	56	180	0.052	Yes
DFGS	1	12	13	48	0.013	No
Uniform Cost	1	55	57	224	0.057	Yes
BFS	2	3343	4609	30509	14.15	Yes
DFGS	2	582	583	5211	3.256	No
Uniform Cost	2	4852	4854	44030	42.48	Yes
BFS	3	14663	18098	129631	103.2	Yes
DFGS	3	627	628	5176	3.474	No
Uniform Cost	3	18234	18236	159707	412.8	Yes

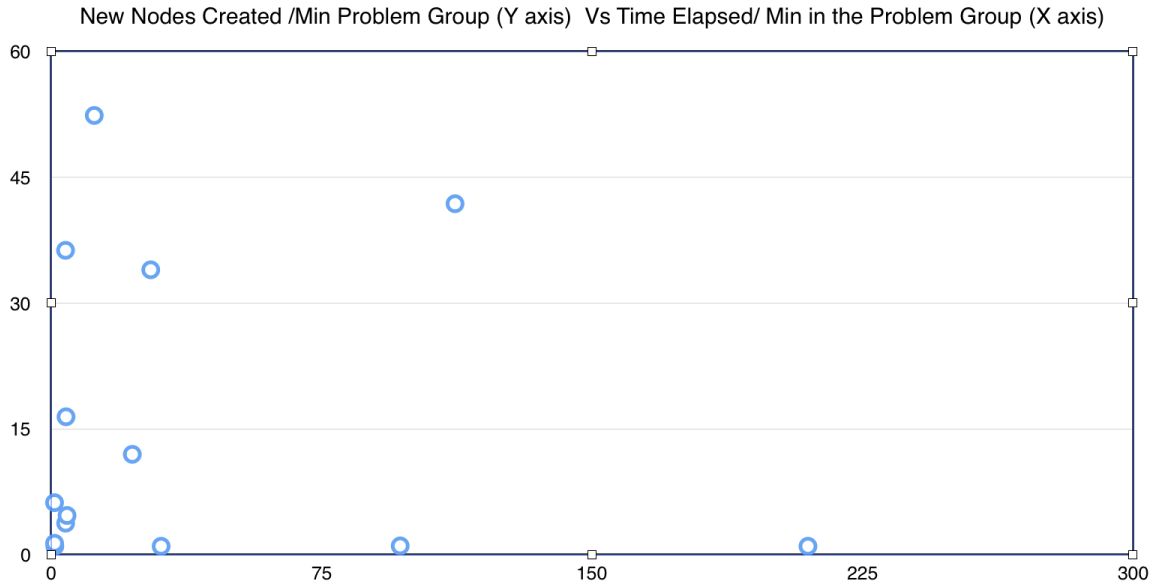
- Depth first graph search never produces an optimal solution.It produces solutions with unnecessary repetitive actions. This is due to the fact that these problems have actions which will negate previous actions and dfs simply keeps trying to execute one action after another until if finds a goal, only backtracking when no action is possible. Generally also, DFS is not optimal.
- Breadth first search and uniform cost search are comparable and always produces optimal solutions.
- Uniform Cost search creates about 20% more new nodes than BFS. This leads to exploring more nodes and eventually more time. Hence BFS is better than uniform cost search in all problems.
- One thing I was surprised by was the time elapsed difference between DFS and BFS, which is 10-20x, which probably explains the complexity involved in creation of a new node.
- Another thing to notice (for verification) is nodes expanded is always 2 greater than goals tests for DFS, because of how the algorithm works.
- BFS is best among these three.

Metrics for A* using Domain Independent Searches

Heuristic	Problem	Nodes Expanded	Goal Tests	New Nodes	Time	Optimal ?

Ignore. Precond	1	41	43	170	0.061	Yes
Level Sum	1	11	13	50	1.258	Yes
Ignore. Precond	2	1506	1508	13820	14.62	Yes
Level Sum	2	86	88	841	108.5	Yes
Ignore. Precond	3	5118	5120	45650	84.09	Yes
Level Sum	3	414	416	3818	784.8	Yes

- Considering number of new nodes created as a proxy for space complexity and time elapsed for time complexity, you are essentially trading off one for another in these algorithms.



- Assuming that in this planing problem, there are considerable effect / resource allocated for executing an action, I would simply not choose non-optimal generations algos, namely DFS evenfor simple problems such as Problem1.
- As we can see heuristic methods are better than non-heuristic methods as complexity of the problem increases.
- Problems 1 and Problems 2 considerably simpler than Problem 3. In problems 1 and 2, the goals are more or less independent on each other, ie one flight can load, fly and unload and finish a part of the goal. Hence you see Level Sum heuristic perform well (ie nodes expanded are very less)
- It should be noted that as expected, level sum heuristic expands lesser nodes than Ignore Preconditions heuristic. (Also lesser nodes than DFGS) Level sum in these problems becomes a better estimate because for each action there are 2-3 constraints (no actions without constraints) and there is a good probability that goals / sub-goals are pretty independent.

- In Conclusion, my optimal solutions
  - **Problem 1 : BFS or A star Ignore Preconditions**
  - **Problem 2 : Level Sum or A star Ignore Preconditions**
  - **Problem 3 : A star Ignore Preconditions**
  - **Overall: A star Ignore Preconditions is a very efficient in these set of problems**