Air Cargo Planning Search Heuristic Analysis

The aim of this document is analyse the result obtained from running uniform non heuristic search algorithms, and heuristic search algorithm A* using domain-independent heuristics to solve three deterministic logistics planning problems with different complexity for an Air Cargo transport system described below

<u>Air Cargo Action Schema:</u>

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))

<u>Problem 1 initial state and goal:</u>

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))

<u>Problem 2 initial state and goal:</u>

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))

Problem 3 initial state and goal:

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))

1. Results

This section list the result obtained from solving the above tree problems, details comparison include In the analysis section.

1.1. Using uniform non-heuristic search algorithms

Problem 1

	Expansions	Goal Tests	New Nodes	Plan Length	Time Elapsed	Optimal
Breadth First search	43	56	180	6	0.067014	Yes
Depth First Graph Search	12	12	48	12	0.006426	No
Uniform Cost Search	101	271	414	6	0.069964	Yes

Problem 2

	Expansions	Goal Tests	New Nodes	Plan Length	Time Elapsed	Optimal
Breadth First search	3401	4672	31049	9	3.335059	Yes
Depth First Graph Search	350	351	3142	346	0.296551	No
Uniform Cost Search	4761	4763	43206	9	3.436852	Yes

Problem 3

	Expansions	Goal Tests	New Nodes	Plan Length	Time Elapsed	Optimal
Breadth First search	14491	17947	128184	12	3.335059	Yes
Depth First Graph Search	1948	1949	16253	1878	1.854708	No
Uniform Cost Search	17783	17785	155920	12	12.746928	Yes

1.2. Using heuristics with A* algorithm

Problem 1

	Expansions	Goal Tests	New Nodes	Plan Length	Time Elapsed	Optimal
H 1	55	57	224	6	3.335060	Yes
Ignore Preconditions	41	43	170	6	0.063518	Yes
pg Level Sum	11	13	50	6	0.827530	Yes

Problem 2

	Expansions	Goal Tests	New Nodes	Plan Length	Time Elapsed	Optimal
H 1	4761	4763	43206	9	3.621328	Yes
Ignore Preconditions	1450	1452	13303	9	2.087355	Yes
pg Level Sum	86	88	841	9	24.852925	Yes

Problem 3

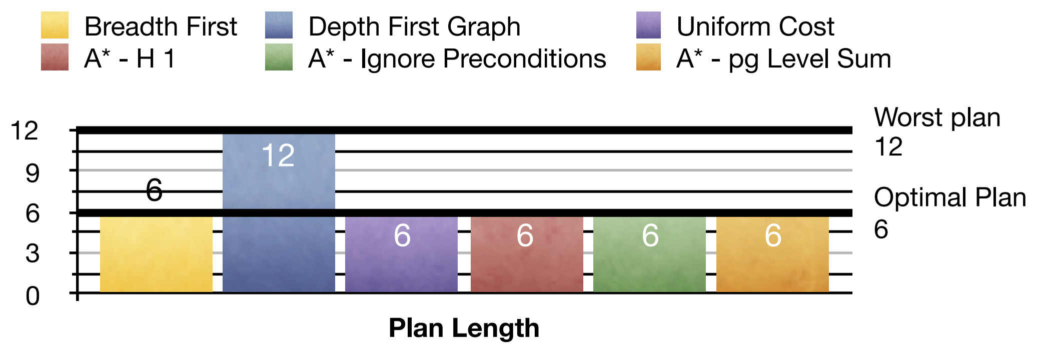
	Expansions	Goal Tests	New Nodes	Plan Length	Time Elapsed	Optimal
H 1	17783	17785	43206	12	3.621328	Yes
Ignore Preconditions	1450	1452	13303	12	2.087355	Yes
pg Level Sum	86	88	841	12	24.852925	Yes

2. Analysis

2.1 Problem 1

We can see in the below figure two of the Uniform Non-Heuristic search algorithms "Breadth First" and "Uniform cost" manage to find the optimal solutions, where "Depth First search" return the worst plan, because [1]Breadth First" and "Uniform cost" optimal search algorithms where "Depth First search" isn't,

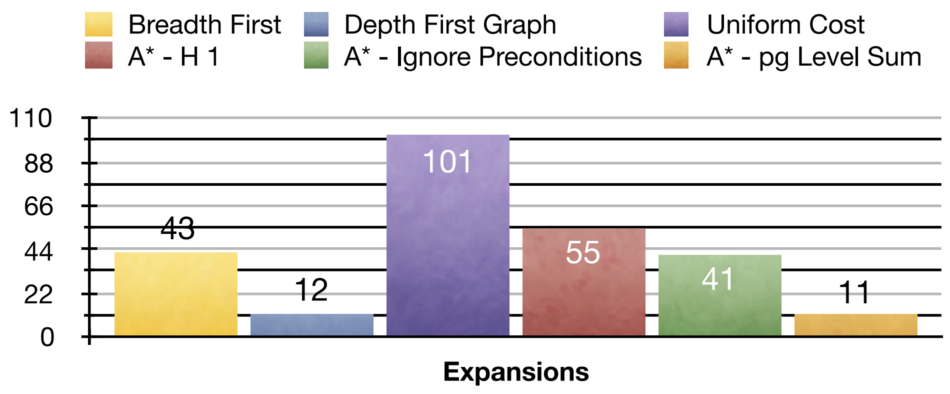
In the other hand all the Heuristic based algorithm manage to find optimal solution, [2] Heuristic search algorithm bound to find an optimal solution given the heuristic function is admissible and solution exist.



And here is an example

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

Out of the search algorithms which manage to find optimal solution A* with Level sum heuristic seems to have the lowest number of expansions, followed by the A* with Ignore Preconditions heuristic and Breadth First search.

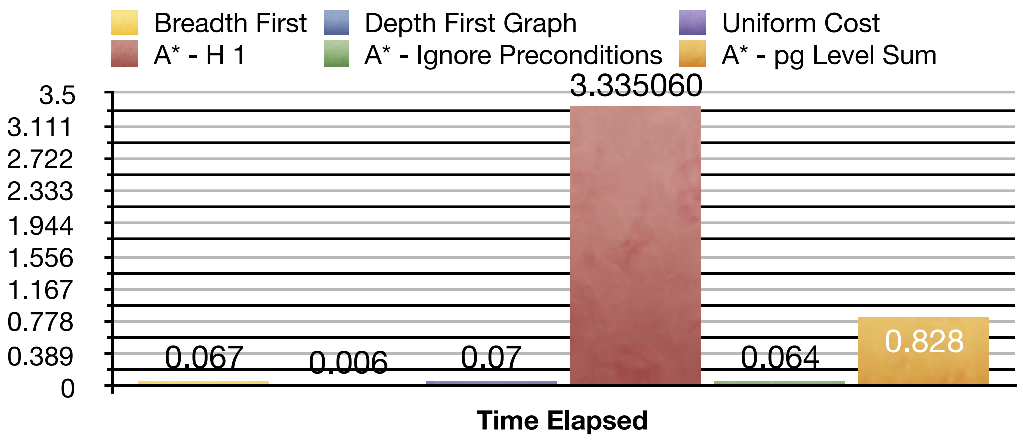


[1] Lesson 11-22 Quiz: Search Comparison

[2] Lesson 11-32 A* Search 5

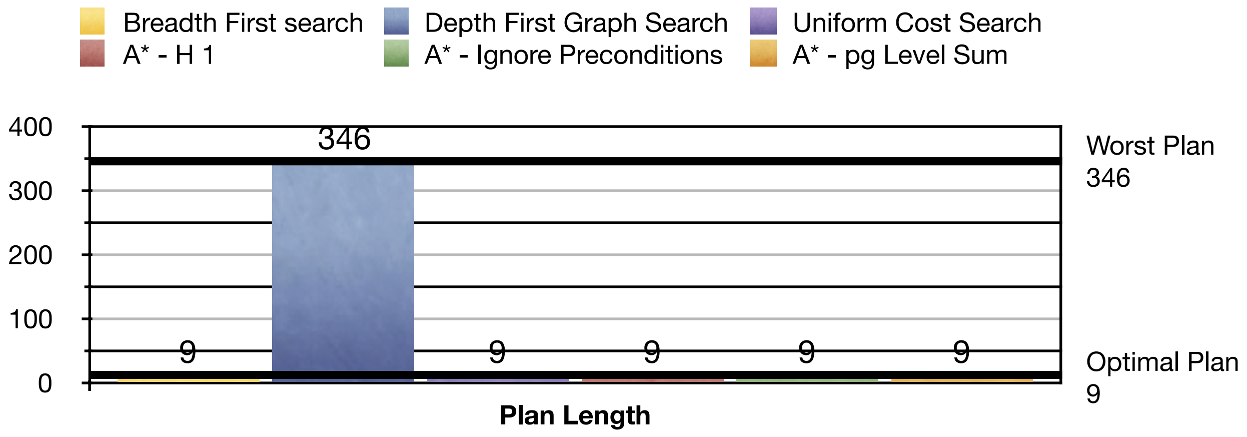
However we can see from the below figure, the time toke Heuristic search based algorithms could be long then non-heuristic algorithms, because it depends on the design of the heuristic and the quality of the implementation.

Where A* with H_1 heuristic is an example of bad design because it toke the algorithm 3.35 seconds to fine solution, even though it has simple and fast implementation. Where A* with Level Sum heuristic is an example of poor implementation because it toke 0.828 second to expand only 11 nodes and find a solution.



2.2 Problem 2

similar to problem 1 , below below figure shows all search algorithms except for “Depth First search” manage to find optimal solution.

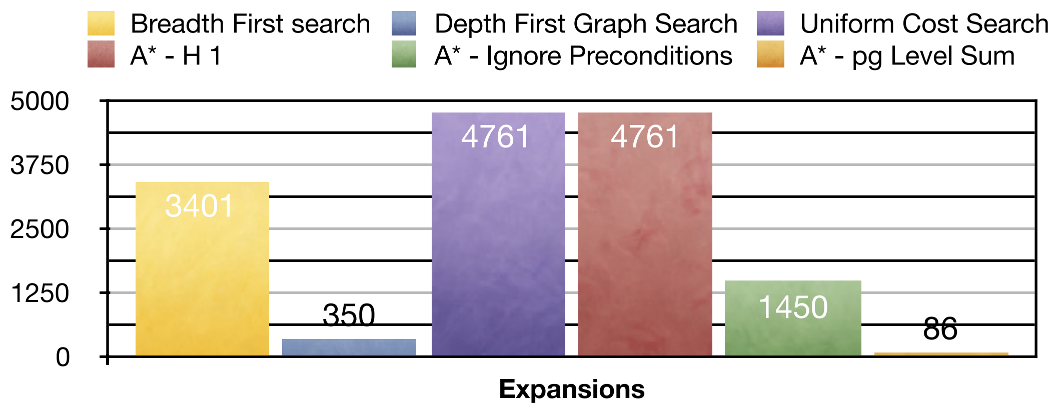


And where is an example

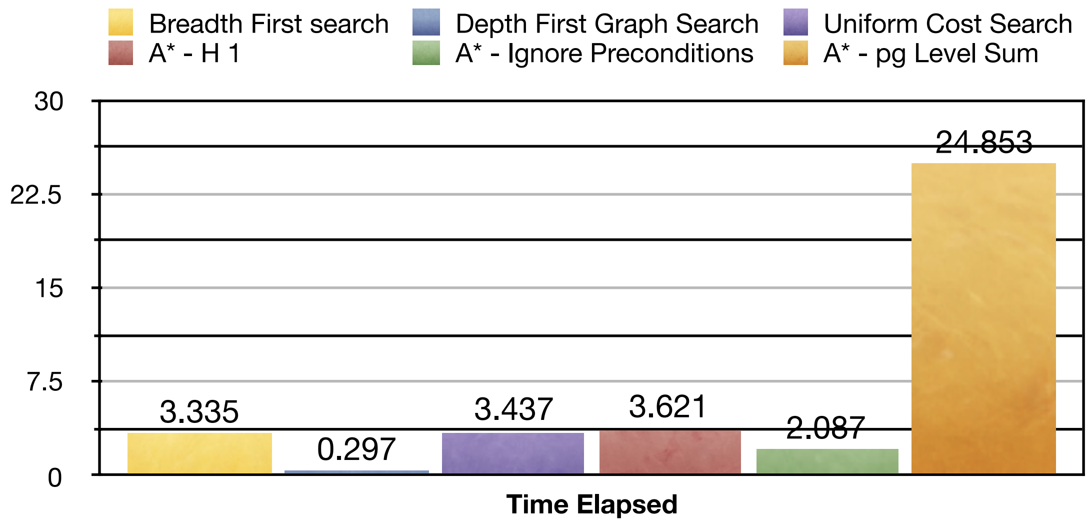
Load(C1, P1, SFO)
Fly(P1, SFO, JFK)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
Load(C3, P3, ATL)
Fly(P3, ATL, SFO)
Unload(C3, P3, SFO)
Unload(C2, P2, SFO)
Unload(C1, P1, JFK)

And for the number of expansion A* with Level Sum heuristic still the lowest, but A* with H1 Heuristic seems to have more or less the same result as “Uniform search” as showing in below figure, most likely because the the heuristic function return the same cost for all nodes which makes the heuristic function less effective.

[3] Considering the job of the heuristic function is to direct the search toward the goal, by adding more knowledge, but in case of H_1 heuristic using same constant doesn’t add any more information, which explain why A* search expanding number of nodes.



A* with Level Sum seems to take longer in problem 2 which has larger state space where it toke the algorithm 24.86 seconds to expand 86 nodes comparing to Breadth First search which toke 3.34 seconds to expand 3401 nodes. And thats due to using poorly implemented heuristic function.



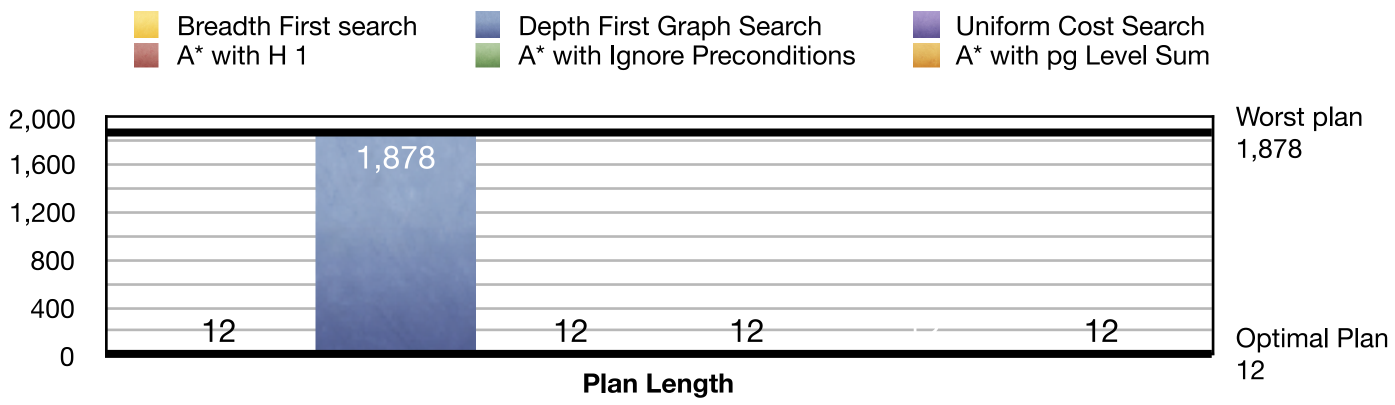
[3] Lesson 11-26 On Uniform Cost

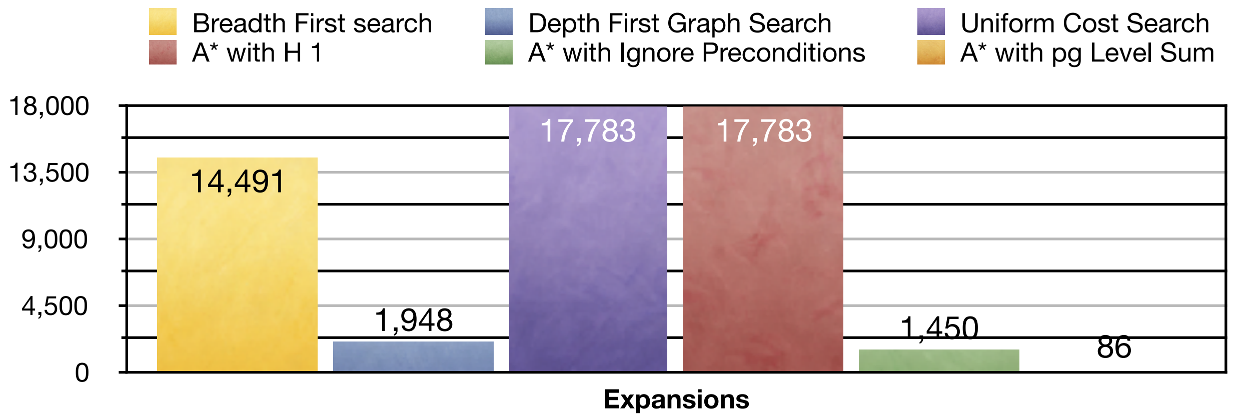
2.3 Problem 3

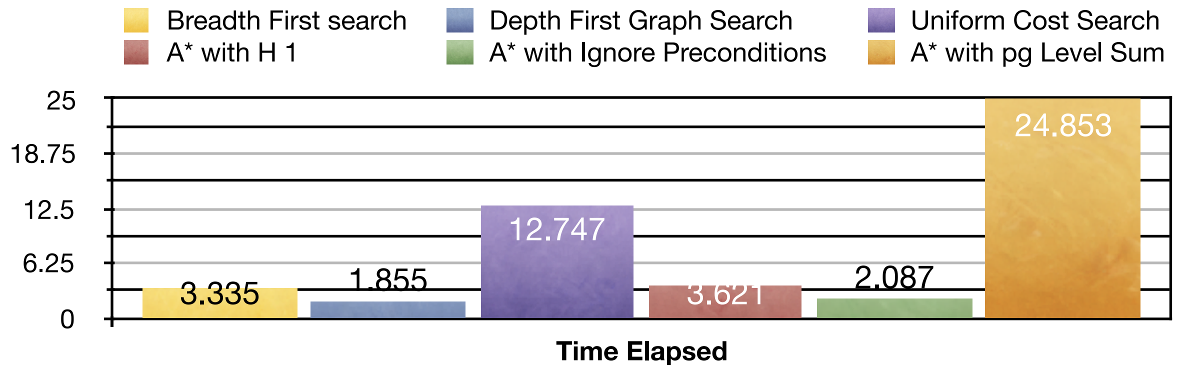
The result for problem 3 similar to problem 2

- All algorithms expect for “Depth First Search” manage to find optimal solution
- A* with had the lowest number of expansion node but the longest running time.
- Both “Uniform Cost” and “A* with H1 Heuristic” had same number of expanded node.

See below figures.







And here is example of an optimal plan

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

3. Conclusion[4]

- "Breadth First" and "Uniform Cost" optimal search algorithms where "Depth First" isn't. Because [1] "Breadth-first" search expands the shallowest nodes first, and "Uniform-cost search expands the node with lowest path cost, where "Depth-First" search expands the deepest unexpanded node first.
    - [2] Uniform cost function explore the state space in all direction, where Heuristic based algorithm use heuristic function to direct the search toward the goal.
    - A* with H1 heuristic seems to expand more or less the same number of nodes as Uniform cost, and with "Level sum" the number of expanded nodes is far lower. But it takes longer, thats because [1] The performance of heuristic search algorithms depends on the quality of the heuristic function

[1] Artificial Intelligence A Modern Approach
[2] Lesson 11-26 On Uniform Cost