Available online at www.sciencedirect.com

ScienceDirect

Transportation Research Procedia 00 (2018) 000–000

Transportation Research Procedia

www.elsevier.com/locate/procedia

World Conference on Transport Research - WCTR 2019 Mumbai 26-31 May 2019

Domain Independent Planning for Air Cargo Movement

Middi Venkata Sai Rishita[a], Middi Appala Raju[b], Tanvir Ahmed Harris[c]

[a]*RNS Institute of Technology, Affiliated to VTU, Belgaum, Bangalore 560076, India*
[b]*Christ(Deemed to be university), Bangalore 560029, India*
[c]*IIT Bombay, Mumbai 400076, India*

Abstract

The technology advancement and usage of information technology in specific, brought drastic changes in the manufacturing locations. Conventional methods are done away with the advent of information technology revolution. As comparative cost of air cargo is more than other modes of logistics, it is highly essential manage the air cargo systems to increase the frequency of delivery. Economic development and high disposable incomes created a heavy demand for import of goods, health equipment etc. This calls for frequency of Air Cargo services. A testimonial to the high Air Cargo business growth is based on the given statistics of India's Merchandise escalated to 284.33 thousand tonnes in April 2018. This paper is based on a Planning search for effective Air Cargo operating system. In this work we have considered two hypothetical air cargo problems with the same action schema but different initial state and goal. Using different Heuristic and non-Heuristic algorithms of Artificial Intelligence, we have come up with optimal solutions of each of the problems. The performance and optimality of each of the planning searches is compared by enumerating the time elapsed and the path length.

Keywords: planning, algorithm, functions, heuristics, freight, logistics, nodes, search

1. Main text

Air Cargo provides wide range of services to facilitate trade, commerce and business. The exponential growth of Air Cargo over the decades, especially with the economic integration of nations, it adds to the economic growth of the nation significantly. According to the working group report, it is expected that air cargo would grow over 10 times with the increase in the economic activities across nations. The phenomenal growth in the Air Cargo demands addressing of infrastructural issues in the form of planning, expansion, required skill development, training the employees to extend world class services to the customers. According to the Economic Survey 2013-14 by the Government of India, the cargo handled at the different airports is increased significantly over the years from 2008. This is given as one of the primary indicators of the "Growth of Civil Aviation" in India.

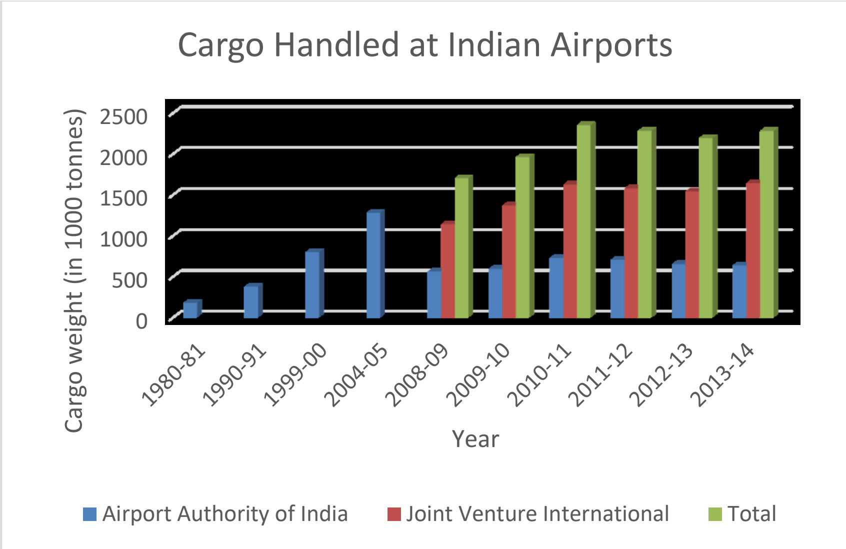

Fig 1.Cargo Handled at Indian Airports

As per the India Economy review, international cargo movement has escalated by 7.2% to 1.86 lakh tonnes in July 2018 as compared to the movement present over a year back. Domestic cargo movement at airports increased by 16.4% to 1.17 lakh tonnes. Considering the total cargo movement, it has increased by 13.1% to 4.46 lakh tonnes.

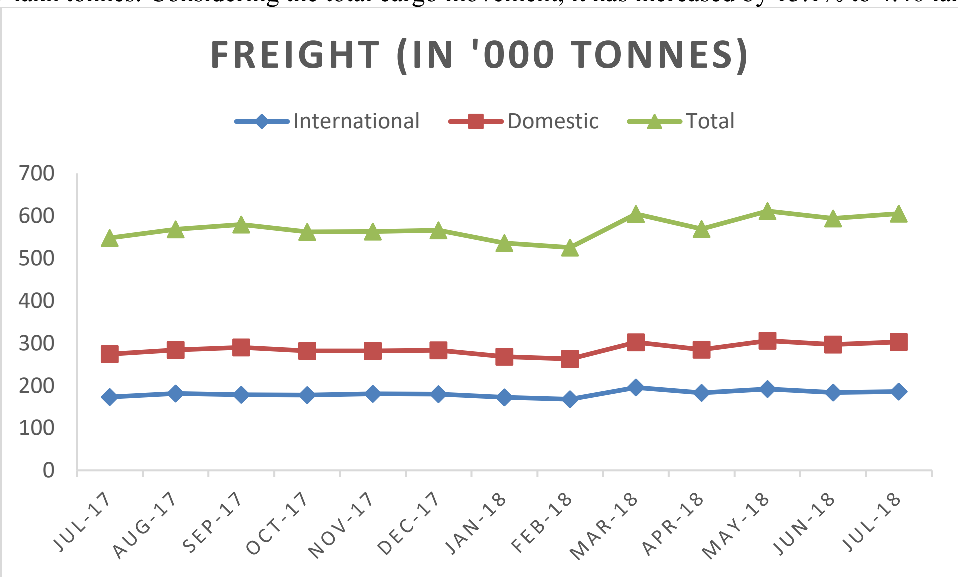

Fig 2. Air Cargo movement from July 2017 to July 2018

International Cargo movement gained 16.5% to 1.79 lakh tonnes in December 2017 which is over a year ago and domestic cargo movement at airports fell 9.6% to 1.03 lakh tonnes. Total Cargo movement rose 13.8% to 2.83 lakh tonnes.

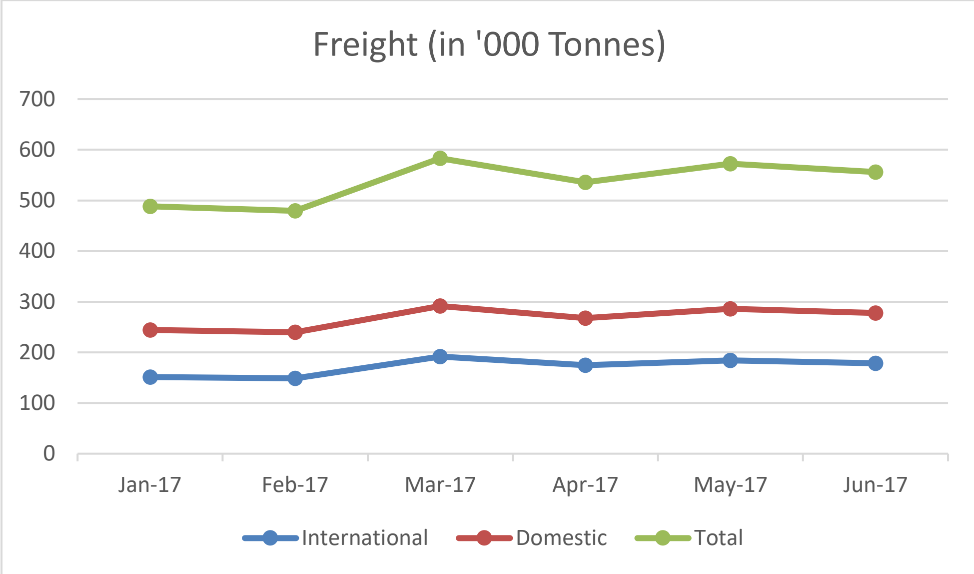

Fig 3. Air cargo movement from jan 2017 to Jun 2017

The ultimate aim of this project is to perform a domain Independent planning search on Air Cargo problem. All problems in the air Cargo domain have the same action schema defined but different initial states and goals. The air Cargo Action Schema:

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: At(c,a) ^ 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))

Different non-heuristic search algorithms like Breadth First Search, Uniform Cost search, Depth First Search algorithms are performed to find an optimal plan. Similarly, Heuristic based A* search is also performed to find which among the algorithms gives the most optimal path for the given air cargo problem. The path of the length and the time elapsed are the key factors in determining the most optimal path.

Nomenclature

A	Load(c,p,a): It indicates the action Load Cargo ‘c’ into plane ‘p’ at the Airport ‘a’
B	PRECOND : Pre-conditions
C	Goal(At(c, JFK): It means that the goal is to deliver cargo ‘c’ to the airport ‘JFK’
D	BFS: Breadth First Search
E	UCS: Uniform Cost Search
F	C1, C2, C3: Cargo 1, Cargo 2, Cargo 3
G	JFK, SF0 : Names of airports

1.1. Breadth First Search Algorithm

Breadth First Search Algorithm is an uninformed search strategy. The traversing begins from the root node which is expanded first. Subsequent nodes are expanded later. The nodes are traversed depth wise i.e all the nodes at a given depth in the search tree. The unexpanded node with the least depth is in front of the FIFO queue. Therefore, it is expanded first. The goal test is not applied to a node when it is identified in the queue for expansion, instead, the goal test is applied when it is generated.

However, the shallowest goal node is not necessarily the most optimistic one. There has to be a non-decresing relationship between path cost and depth of the node. Since every node that has to be expanded is placed in the buffer, the space complexity is too high for this algorithm.

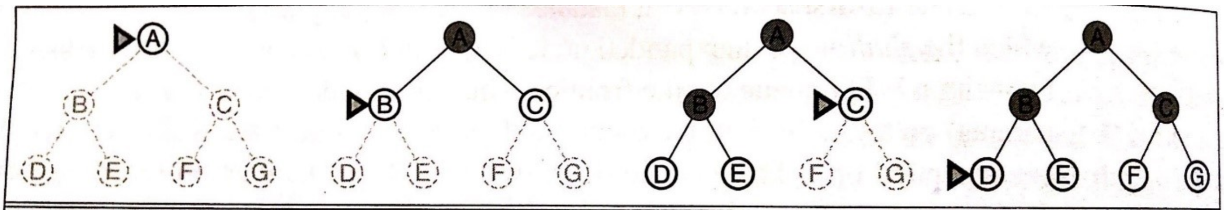

Fig 4. Breadth First search on a simple binary tree

1.2. Uniform Cost search

The Uniform Cost Search expands the node n with the lowest path cost. The lowest path cost function is g(n). There are two important differences with respect to the BFS algorithm. When the node is selected for expansion itself, the goal test is applied. The first node generated maybe the suboptimal path. A test is added in case a better path is found to a node currently on the frontier of a FIFO queue. In this algorithm, the emphasis is on the total cost of the path rather than the number of nodes.

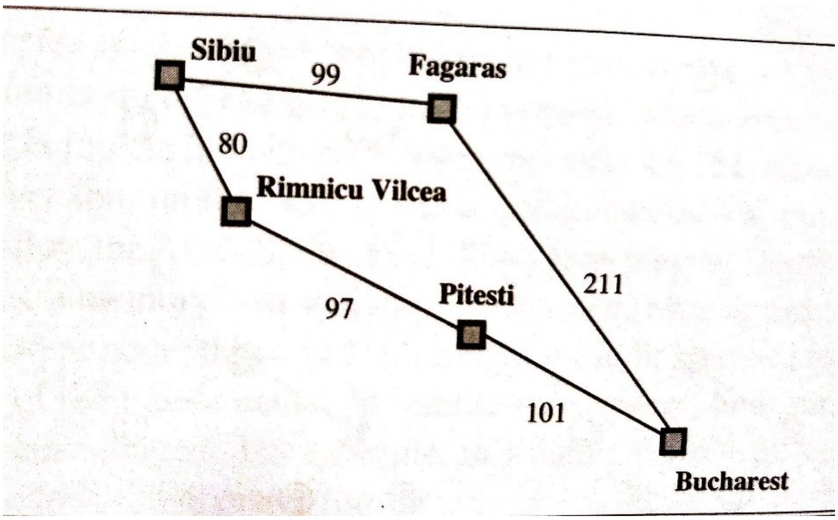

Fig 5. Part of a state space to illustrate uniform-cost search

1.3. A search: Minimizing the total estimated solution cost*

A* search evaluates nodes by combining g(n), the cost to reach the node, and h(n), the cost to get from the node to the goal:

f(n)=g(n)+h(n)

Since g(n) gives the path cost from the start node to node n, and h(n) is the estimated cost of the cheapest path from n to the goal, we have

f(n)= estimated cost of the cheapest solution through n

2. Optimal Solution for the Problem 1

The initial state and goal of problem1 is as follows: -

Init(At(C1, SF0) ^ At(C2, JFK)
^ At (P1, SF0) ^ At(P2, JFK)
^ Cargo(C1) ^ Cargo(C2)
^ Plane(P1) ^ Plane (P2)
^ Airport(JFK) ^ Airport(SF0))
Goal(At(C1, JFK) ^ At(C2, SF0))

The first algorithm applied on problem1 is the Breadth First Search Algorithm. The optimal solution for problem obtained from applying the first algorithm is:
Load(C1, P1,SF0)
Load(C2, P2, JFK)
Fly(P2.JFK,SF0)
Unload(C2, P2, SF0)
Fly(P1,SF0, JFK)
Unload(C1, P1, JFK)
The time elapsed was 0.054 seconds and plan length was found to be 6 nodes.
When Uniform Cost search algorithm was applied the optimal problem that was arrived at had 6 nodes as follows:
Load(C1, P1, SF0)
Load(C2,P2,JFK)
Fly(P1, SF0,JFK)
Fly(P2, JFK,SF0)
Unload(C1,P1,JFK)
Unload(C2,P2,SF0)
Breadth First Search finds the shortest path in terms of the least number of steps but it does not take into account the cost of each path. Hence, BFS will not find the shortest path in terms of the total cost. The number of nodes expanded by UCS is more because after finding the goal state, the UCS algorithm traverses to find the least cost path. The time elapsed and the nodes expanded maintain an inverse proportionality. BFS finds the goal 1.2 times faster than Uniform Cost Search.
The Ignore pre-conditions heuristic(HIP) drops all the pre-conditions from actions. The A* search with HIP, when applied to the problem1 gives the solution as: -
Load(C1,P1,SF0)
Fly(P1,SF0,JFK)
Unload(C1,P1,JFK)
Load(C2,P2,JFK)
Fly(P2,JFK,SF0)
Unload(C2,P2,SF0)

The time elapsed is 0.068. The next in line is the A* Search planning graph Level Sum Heuristic. The optimal plan after applying this heuristic is: -
Load(C1, P1, SFO)
Fly(P1, SFO, JFK)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
Unload(C1, P1, JFK)
Unload(C2, P2, SFO)
The Level Sum Heuristic uses a Planning Graph and estimates the sum of all actions that must be carried out from the current state to satisfy each individual goal condition. The H ignore pre-conditions finds the goals faster as it ignores pre-conditions required for an action to be executed to make the problem easier in order to estimate the minimum number of actions. These actions are carried out from the current state to satisfy all goal conditions. A* Search Ignore Pre-condition heuristic finds goal 30 times faster than Planning Graph Level Sum. The former expands 4 times more nodes than the latter.

The A* Search Ignore Pre-condition Heuristic achieves 50% of its At(C1,JFK) ^ At(C2,SF0) goal fastest by focusing on finishing one task at a time. It uses the shortest possible path length of 6 and reaches the goal in 0.068 seconds.

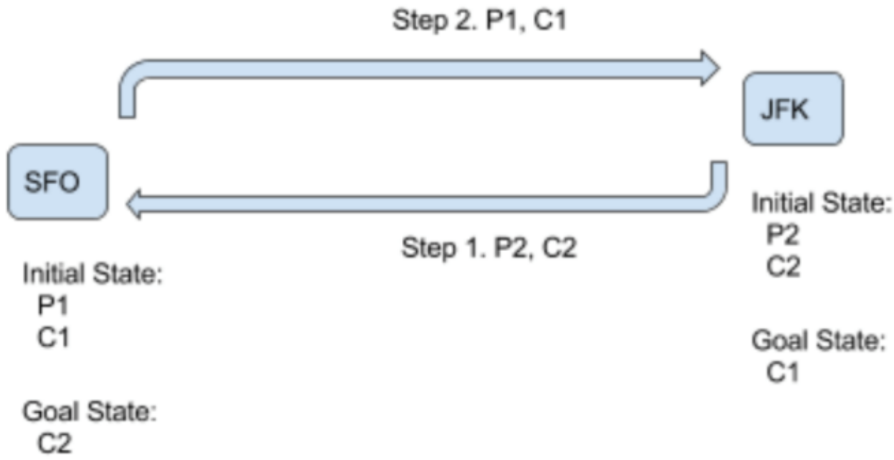

Fig 6:Visualization of Optimal Solution for Problem1

3. Optimal Solution for Problem 2

The initial state and goal of problem is slightly different for problem 2 even though the action schema remains the same. It is given as follows: -

Init(At(C1, SF0) ^ At(C2, JFK) ^ At(C3, ATL)
^ At (P1, SF0) ^ At(P2, JFK) ^ At(P3, ATL)
^ Cargo(C1) ^ Cargo(C2) ^ Cargo(C3)
^ Plane(P1) ^ Plane (P2) ^ Plane(P3)
^ Airport(JFK) ^ Airport(SF0)) ^ Airport(ATL))
Goal(At(C1, JFK) ^ At(C2, SF0)) ^ At(C3, SF0))

Optimal Sequence of Actions after running the Breadth First Search Algorithm are given below. The plan length consists of 9 nodes with 24.9 seconds elapsed.

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)
When Uniform Cost search algorithm was applied the optimal problem that was arrived at had 9 nodes as follows:
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(C1, P1, JFK)
Unload(C2, P2, SFO)

Both Uniform Cost Search algorithm and Breadth first search algorithm are optimal with plan length of 9. The BFS algorithm finds goal more than two times faster than the UCS algorithm. However Uniform Cost Search algorithm expands 1.5 times more nodes than Breadth first search algorithm.
The Ignore pre-conditions heuristic(HIP) drops all the pre-conditions from actions. The A* search with HIP, when applied to the problem2 gives the solution as: -
Load(C3, P3, ATL)
Fly(P3, ATL, SFO)
Unload(C3, P3, SFO)
Load(C1, P1, SFO)
Fly(P1, SFO, JFK)
Unload(C1, P1, JFK)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
Unload(C2, P2, SFO)
The time elapsed is 21.434 seconds. In the case of A* Planning Graph Level Sum the time elapsed is 256.832 seconds.

The optimal plan of actions using the Level Sum Heuristic is as follows: -
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(C1, P1, JFK)
Unload(C2, P2, SFO)

A*S algorithm will always find the lowest cost path to the goal dependent on whether the heuristic estimate function h for a state is less than the true cost of the path to the goal through that state. It is therefore not guaranteed to be optimal until it is used with an appropriate heuristic function. The time elapsed with A* Search Ignore Pre condition heuristic finds the goal 12 times faster than the level sum heuristic. The former expands 17 times more nodes than planning graph Level Sum.

Even in the case of Problem 2 A* Search Ignore Pre-condition heuristic is clearly the most optimal plan since it achieves 33.33% of its At(C1, JFK) ^ At(C2, SFO) ^ At(C3, JFK) goal fastest and 66.67% of its goal by the 6^{th} action by focusing on finishing one task at a time. It uses the shortest possible plan length of 9 and reaches the goal in the fastest time of 21.434 seconds as compared to all other optimal sequences. It takes the shorter route from ATL to SFO first to deliver a first portion of the goal faster.

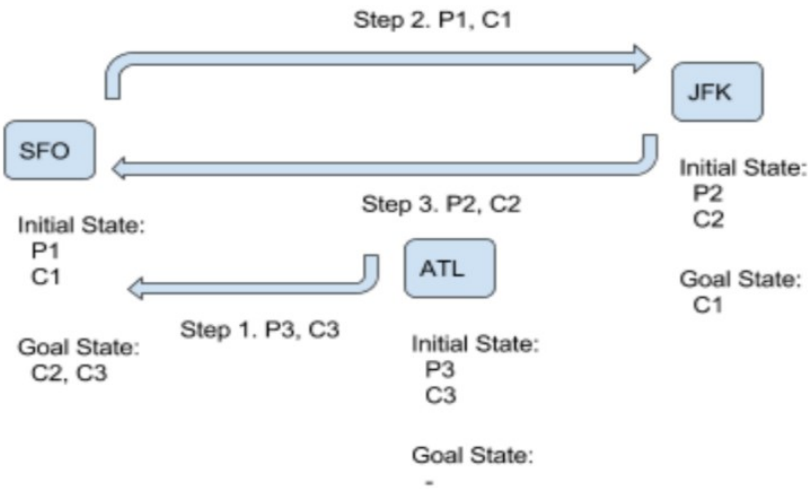

Fig 7. Visualization of optimal solution for Problem 2

4. Optimal solution for problem 3

The initial state and goal of problem 3 is defined as

Init(At(C1, SF0) ^ At(C2, JFK) ^ At(C3, ATL) ^ At(C4,ORD)
^ At (P1, SF0) ^ At(P2, JFK)
^ Cargo(C1) ^ Cargo(C2) ^ Cargo(C3)^ Cargo(C4)
^ Plane(P1) ^ Plane (P2)
^ Airport(JFK) ^ Airport(SF0)) ^ Airport(ATL)) ^ Airport(ORD))
Goal(At(C1, JFK) ^ At(C2, SF0)) ^ At(C3, SF0)^ At(C4, SF0))

Optimal Sequence of Actions after running the Breadth First Search Algorithm are given below. The plan length consists of 12 nodes with around 196 seconds elapsed.

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)
The solution after Uniform Cost search has been applied has 12 nodes with the time elapsed being equal to 595.826 seconds
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(P2, ORD, SFO)
Fly(P1, ATL, JFK)
Unload(C4, P2, SFO)
Unload(C3, P1, JFK)
Unload(C1, P1, JFK)
Unload(C2, P2, SFO)
Both the non-heuristic algorithms are equally optimal with the number of nodes being equal to 12. However, the BFS algorithm arrives at the goal state more than two times faster than the Uniform Cost Search Algorithm. This is evident from the fact that the UCS algorithms expands 1.3 times more nodes than BFS.
The Ignore pre-conditions heuristic(HIP) drops all the pre-conditions from actions. The A* search with HIP, when applied to the problem3 gives the solution as: -
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(C1, P1, JFK)
Unload(C2, P2, SFO)
Planning Graph limit sum heuristic estimates the sum of all actions including non-minimal ones that adds a lot of redundant actions into consideration. This makes the process unnecessarily slow. No result was obtained even after 10 Minutes. A*S Ignore pre-conditions is clearly the most optimal plan since it achieves portions of its goal soonest. It achieves 25% of its **At(C1, JFK) ^ At(C2, SF0) ^ At(C3, SF0)^ At(C4, SF0))** goal fastest, 50% of its goal by the 10th action, 75% by the 11th action, and 100% by the 12th action. It uses the shortest possible path length of 12 and reaches the goal in the fastest time of 138.025. It takes shorter route from JFK to SFO via ORD first instead of the longer SFO to JFK via ATL to deliver the first portion of the goal faster.

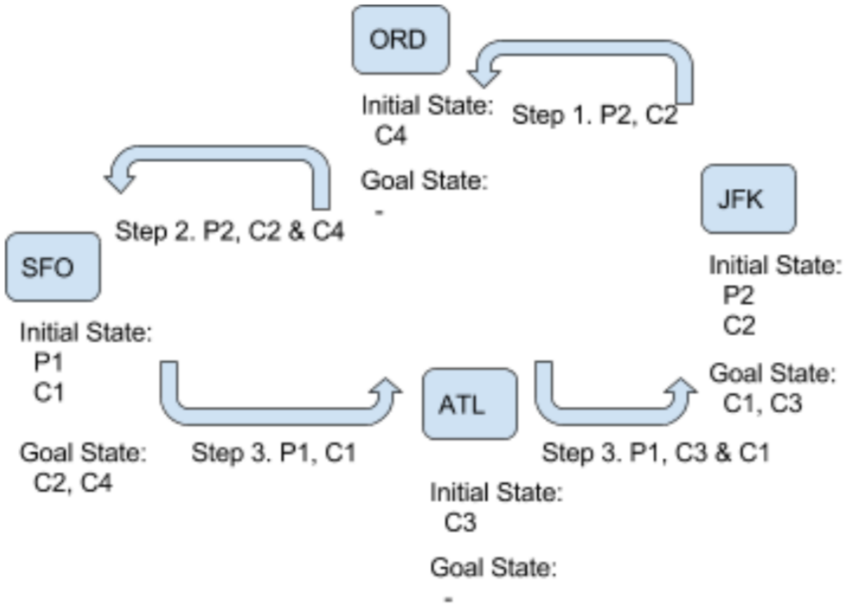

Fig 8. Visualization of most optimal solution of problem 3

5. Results

A visualization of the Specific Metrics of each of the algorithms applied on Problem1, Problem2 and Problem3 gives the comprehensive study of the distinctions between the heuristic and non-heuristic approaches.

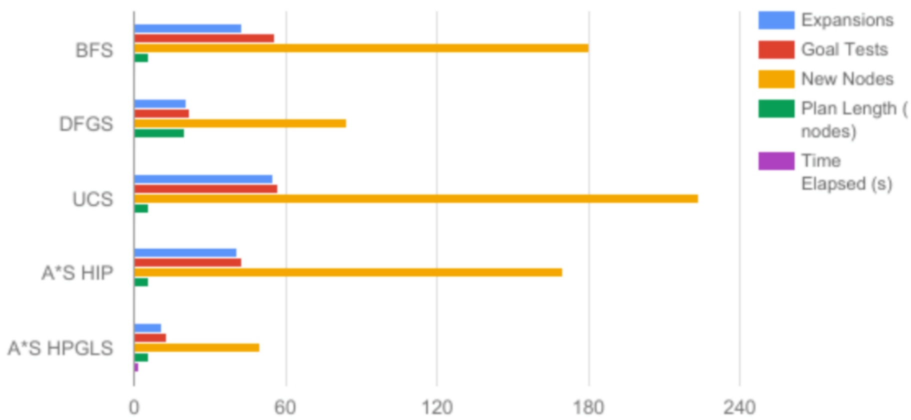

Fig 9. Specific metrics of problem1

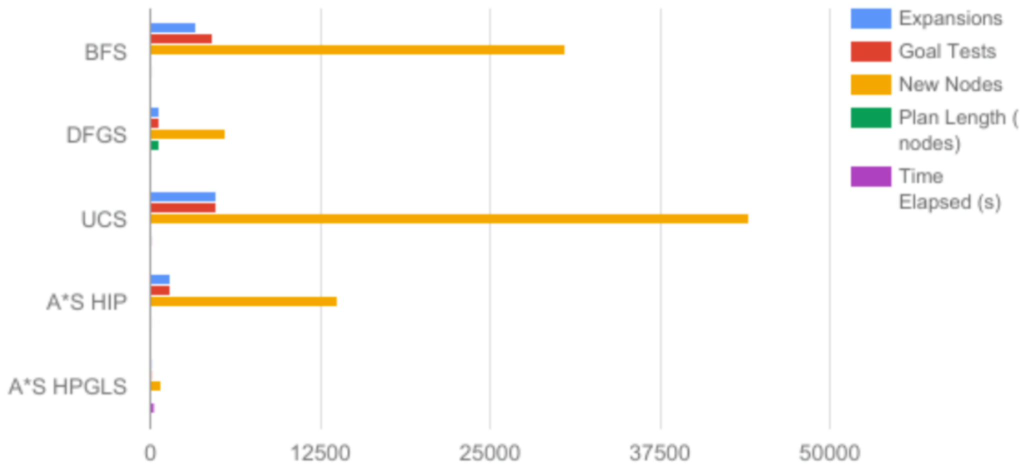

Fig 10. Specific metrics of Problem2

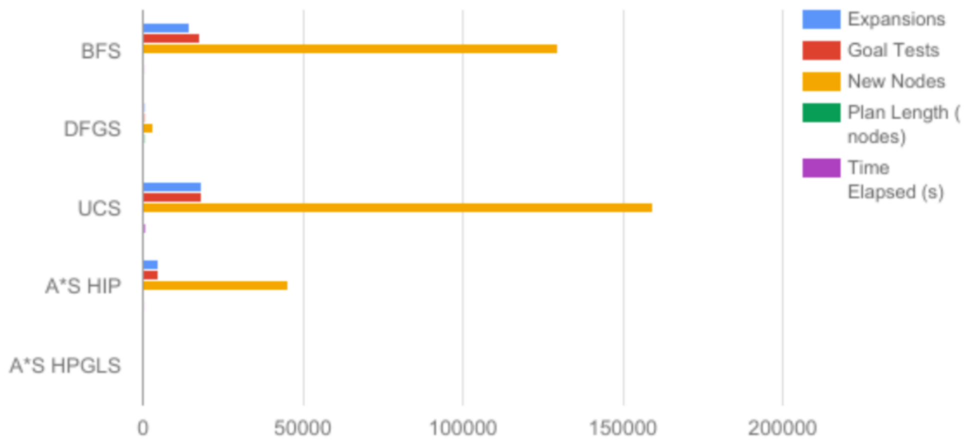

Fig 11.Specific Metrics of Probem3

6. Conclusions

In all the three problems, the A* Ignore Pre-condition Heuristic turned out to be the most optimal solution for air cargo planning. In the case of problem 1 the Depth First Search algorithm required a plan length which was nearly four times larger than A* Search Ignore Pre-Conditions.

The time frame for both BFS and A* search heuristic was same but the Uniform Cost Search Algorithm required two times more expansions than the A* search. For problem 3 the Breadth First Search took 1.4 times longer than A*search to find the solution. BFS and UCS required more than 3 times the expansions than the A* search.

References

K. Zhu, M.W Ludema, 2000. Air Cargo transport by multi-agent based planning, Proceedings of the 33rd Annual International Conference on System Sciences

New York, pp. 387.

Ching-Hui Tang, Hsin-Wei Chang, 2010, Optimization of Stocastic Cargo Container Loading plans for Air Express Delivery, 2010 Second International Conference on Computer and Network Technology.

Chunhua Tian, Hao Zhang, Feng Li, 2008, 2008 IEEE International Conference on service operations and logistics, and Informatics.

Chunhua Tian, Hao Zhang, Feng Li, Tie Liu on Air Cargo load planning system: A rule based optimization approach

Economic Survey, Government of India, 2013-14

Economic Survey, Statistical Appendix Government of India, 2013-14

India Economy Review, Capital Market February 2018