IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 2, February 2024

DOI: 10.17148/IJARCCE.2024.13229

SHORTEST PATH SYSTEM FOR NIGERIAN AIR DISPATCH NETWORK USING MODIFIED DIJKSTRA'S ALGORITHM

Mba, Uchenna Ewa [1], Mbeledogu, N. Njideka[2]

Department of Computer Science, Nnamdi Azikiwe University Awka, Anambra State, Nigeria[1,2]

declanmba@gmail.com[1], nn.mbeledogu@unizik.edu.ng[2]

Abstract: In recent times, the problems of efficiency and marginal productivity in the air transportation industry have become one of a global concern, particularly, in Nigeria. Safer and faster movement of passengers and their luggages in a record time with drastic reduction in the running cost are bottlenecks facing the aviation industry. The Traditional Dijkstra's Algorithm could determine shortest path in air route distance but its exiting mechanism leads it to infinite loop. Based on this, the research sought to deploy Modified Dijkstra's Algorithm in the planning, calculation and implementation of air route network to determine the shortest path distance. The approach employed the Comparison Addition Model (CAM) in determining the optimal distance from the source to the destination within the network. Permutation and combination analysis was used to determine all possible routes. Multiple parameters (time factor optimization, congestion reduction, memory utilization) capable of determining the shortest and optimal distance a flight can possibly travel in a routing network were handled. Python Programming Language was proficient for its coding. The performance evaluation of Modified Dijkstra's Algorithm and Traditional Dijkstra's algorithm based on F_1 Score, Recall and Precision was carried. The Modified Dijkstra's Algorithm was found to perform better than the Traditional Dijkstra's Algorithm.

Keywords: Modified Dijkstra Algorithm (MDA), Comparison Addition Model (CAM), Shortest Path Air Route Network, Air Transportation

I. INTRODUCTION

Air transportation is one of the most important components of the world's transportation system. It is a prime yardstick for gauging development. Not only does it provide a major means for long-distance travel in the world, its economic impacts on global and national economies are substantial. Nigeria operates a certified Category 1 aviation (FAAN, n.d) and its air safety status has improved and now presents the country in the premier league of nations that are highly rated in air transport (Akpoghomeh, 1999). This propels it to generate some financial benefits for the nation, thereby, making all the levels of government and private sectors keenly driven to invest in it for its high performance in the transport sector.

Air routing is a major component of air transportation system. The designated links called flight paths connects all nodes (airports) within a given geographical region of coverage of an airline operation. In real life situations, using a large size of aircrafts coupled with complex geographical diverse regions and difficult terrains, designated air routes have been a major concern to airline operators as its determinant is usually stochastic and time-dependent. Due to this, there is a need to necessitate the use of efficient air transportation routing which would safely and speedily convey people and goods from one place to another without delay. Hence, the concept of shortest path determination is considered.

Shortest path routing refers to the process of finding paths through a network that has a minimum of weighted distance or other cost metric. Its topology communication network is defined using a directed weighted graph. Any path between two nodes can go through various intermediary nodes and link arc.

The goal of shortest path routing is to find a path between two nodes that has the lowest total cost. The total cost of a path is the sum of arc costs in that path. Shortest path problem algorithms are classified into four categories (Singhal, 2020). They are single source shortest path algorithm, single destination shortest path algorithm, allpairs shortest path algorithm and single pair shortest algorithm.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 2, February 2024

DOI: 10.17148/IJARCCE.2024.13229

Single Source Shortest Path (SSSP) Problem Algorithm

The shortest path problem finds the path with the lowest total weight between two vertices for a weighted directed graph. Here, the SSSP finds the shortest path from the source (a single vertex) to every other vertex in the graph. If there is a cycle with a negative total weight, the cycle allows the smallest weight to reduce the total weight of the path. The algorithm must either return to the shortest paths between all vertices or report that there is a negative cycle in the graph.

Single Destination Shortest Path (SDSP) Problem Algorithm

It is the shortest path problem where the shortest path from all the vertices to a single destination vertex is computed. By reversing the direction of each edge in the graph, this problem reduces the single-source shortest path problem.

All Pairs Shortest Path (APSP) Problem Algorithm

It finds the shortest path between any combinations of two pairs selected (every pair of vertices). This runs on every vertex in the graph and the runtime is determined by multiplying the runtime of the APAS algorithm n.

Single Pair Shortest Path (SPSP) Problem Algorithm

The shortest path between a given pair of vertices is computed. The A* Search Algorithm is used for solving this. It uses the best-first search that is, a heuristic function h (n), and cost to reach the node n from the start state.

Mathematical Models of Shortest Path Problems

a) Floyd-Warshall Algorithm

This algorithm finds the shortest paths in a directed weighted graph with positive or negative edge weights but with no negative cycles. A single execution of the algorithm will find the lengths (summed weights) of shortest paths between all pairs of vertices.

b) Bellman-Ford Algorithm

Is an algorithm that computes shortest paths from a single source vertex to all of the other vertices in a weighted digraph. It is slower than Dijkstra's algorithm for the same problem, but more versatile, as it is capable of handling graphs in which some of the edge weights are none-positive.

c) Dijkstra's Algorithm

The Dijkstra's shortest path algorithm is the most commonly used to solve the single source shortest path problem today. This algorithm avoids the use of negative cycles by assuming that all the edges are positive. This allows the algorithm to use a heap where the key of a vertex is the length. Beginning at the source vertex, each adjacent vertex is assigned the cost value of edge joining them. Next, each vertex will process the least accumulated cost and assign the accumulated cost plus the edge cost to each of its adjacent vertices. This step is repeated until each vertex has been processed. If a vertex is revisited, the algorithm will assign the new cost if it is lower than the currently assigned cost. At the end of the process, the algorithm has solved not only the single pair shortest path problem, but can find the distance between the source vertex and any other vertex (Gupta, 2017).

II. RELATED WORKS

Lavlinskaya*et al.* (2020) designed and implemented a shortest path algorithm for graph in instances of semantic optimization. The researchers pointed out that in search for shortest path in a poorly structured area like word puzzles or complex environment; the degree of thematic traffic is usually overwhelming. Using Floyd-Warshall algorithm of finding shortest paths as a mathematical model for weighted graph $G = (V, E)$, having tops V and arcs E ($|V|=n$, $|E|=m$). The weights $wij=(vi,vj)$, $vi,vj \in V$. II. This algorithm compared all possible paths through the graph between each pair of vertices. The work has a significant effect on the current research but the algorithm can be slow when used on a large graph. Also, the algorithm does not always find the shortest path between two nodes because it is best suited for graphs with negative edges and weights, thus,finds a path that is longer than the shortest path.

Arslan and Manguo˘gl (2019) designed a hybrid single-source shortest path algorithm for graphs. The researchers used Dijkstra's and Floyd-Warshall algorithms to find the shortest path in weighted graphs. They maintained that the defined weights of a graph are based on its reachable shortest paths between nodes which are polynomial, positive and expressive without the phenomenon of tottering. The work was efficient to show an approach to solve the problem of the Single-Source Shortest Path problem in dynamic hybrid graphs with optimized time and space complexities, but had large memory consumption rate.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 2, February 2024
DOI: 10.17148/IJARCCE.2024.13229

Ofem *et al*. (2017) implemented an algorithm for shortest pathway and time determination in a wireless packet switch network system. The research showed that using Dijkstra Algorithm and prioritizing open shortest path first (OSPF) will provide useful problem solving method in shortest path and time determination. The researcher maintained the approach of coding the algorithm and ability to manage and control the network is efficient in time/delay-sensitive applications with differing quality of service requirements. Finding the shortest paths to a node by halting the algorithm when the shortest path to the destination node is defined makes the algorithm optimal for network routing but like the traditional Dijkstra algorithm, it still remains in efficient in digraphs.

Kairanbay and Hajar (2013) conducted a research on evaluations of shortest path algorithms. The researchers evaluated the Dijkstra's Algorithm, Floyd-Warshall Algorithm, Bellman-Ford Algorithm, and Genetic Algorithm (GA) in solving the shortest path problem for word length selection. Dijkstra's algorithm has a less computational time as compared to other reviewed algorithms. With Dijkstra's, Floyd-Warshall and Bellman-Ford algorithms having time complexities of *n2 + m, n3* and *nm* respectively. This research showed that these algorithms were acceptable in terms of their overall performance in solving the shortest path problem, but can produce only one solution which is not always optimal.

Samui (2011) pointed out in distribution system for planning and considering reliable feeder routing in transport model that the main goal of shortest path determination is to reduce travel time and increase profit. It was also observed that the most fundamental and commonly encountered problems in the study of routing, transport and communication networks are determining shortest path issues and nodes placement. Dijkstra's algorithm was perfect in finding the shortest path; it was effective for a non-negative edges and vertices for every transport model so computed, but very ineffective for combinatorial negative edges and vertices.

Jindal *et al*. (2010) in their work on analysis of shortest path algorithms showed that efficient algorithm for finding shortest paths must have less time and space complexity as compared to other existing algorithms. They used Bellman Ford algorithm to solve the single source shortest path problems in which edge weight was negative. This algorithm returned relaxed edges when iteration for *n-1* times specifying a negative weighted edge.

The researchers maintained that for optimality to be obtained at *n-1* iteration, the edges should have been relaxed and with the shortest path discovered. Bellman-Ford's algorithm lacks in scalability and takes more polynomial time, Floyd-Warshall algorithm best handles graphs with negative edges and weights but Dijkstra's algorithm tends to work better than these.

Dijkstra's label algorithm can effectively solve the shortest path problem of simple weighted undigraph but conducts a blind scan, thus, increasing the processing time. Also, the exiting mechanism which is check T_r ("not pass vertex set" in step r, r≥0), if $T_r = \emptyset$ then the algorithm ends can only be effective in solving the shortest path problem of simple weighted undigraph and very inadequate in digraph (Wang, 2012). Since air route trajectory is a seemingly directed path on graphs with the routes clearly defined, there is a need for improvement in order to work in directed graph, hence, a modified Dijkstra's algorithm is needed.

Modified Dijkstra's Shortest Path (MDSP) Algorithm is works on deterministic or fixed routing concept. It involves advanced determination of the optimal routes between given nodes. This routing technique uses multiple parameters instead of single parameters.It can find shortest paths in either digraph or undigraph.

When applied to air route network, it becomes apparent that even when landing in a destination is impossible, flight path can be re-directed to the closest airport using the shortest MDSP algorithm without necessarily re-computing the parameters while minimizing losses in air route network trajectory which leads to excessive fuel consumption. MDSP enhances network flow control.

It overcomes known complexity of air delay as well as assists the aviation industry in setting bench mark in the air route network system. Also, Itdecreases the number of iterations with fewer nodes in the Dijkstra standard algorithm, thus, resolves the problem of the big memory area of the traditional Dijkstra.

III. RESEARCH METHODOLOGY

Object oriented and design methodology was employed for both the analysis and design. Figure 1 shows the data flow diagram of the traditional Dijkstra's algorithm.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 2, February 2024

DOI: 10.17148/IJARCCE.2024.13229

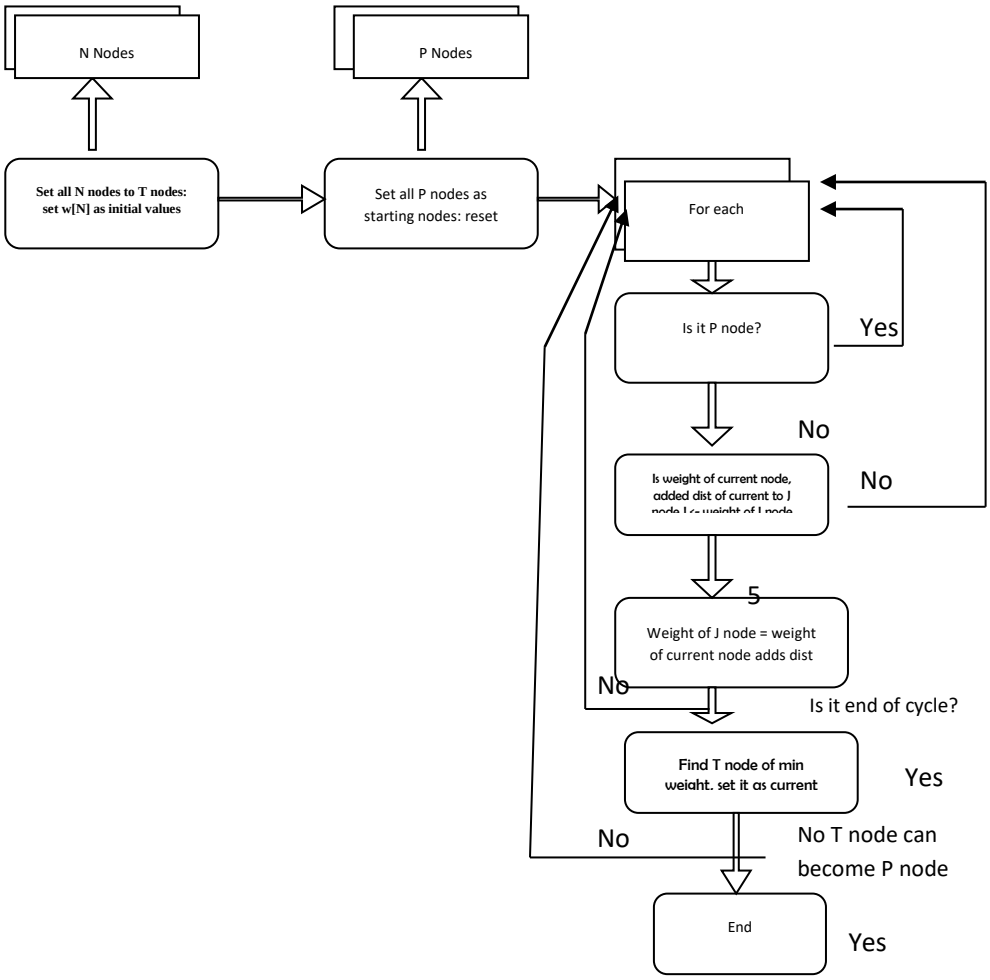

Figure 1: Data Flow Diagram of Dijkstra's Algorithm

IV. MATERIALS AND METHOD

A. Network Modeling

The network model for an air route and flight dispatch system was abstracted as a graph (Figure 2). The algorithm works efficiently for both non-negative directed and undirected graph network systems.

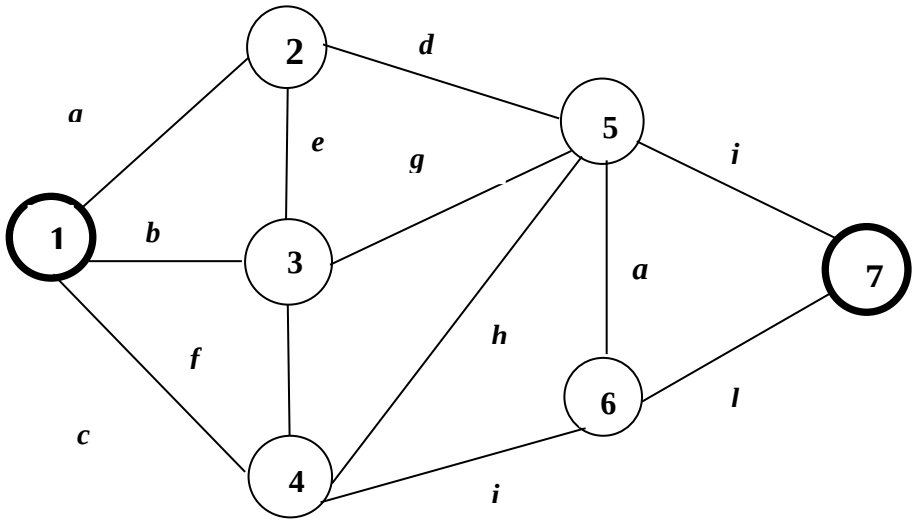

Figure 2: Air Route Network Model Showing Connected Nodes and Distances

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 2, February 2024

DOI: 10.17148/IJARCCE.2024.13229

B. High Level Model of the Proposed System

Seven Nigerian airports were used as case study – Enugu (source), Owerri, Asaba, Calabar, Port-Harcourt (PHC), Abuja and Lagos (Destination). This model diagram (Figure 3) shows the flight movement from source node to destination node.

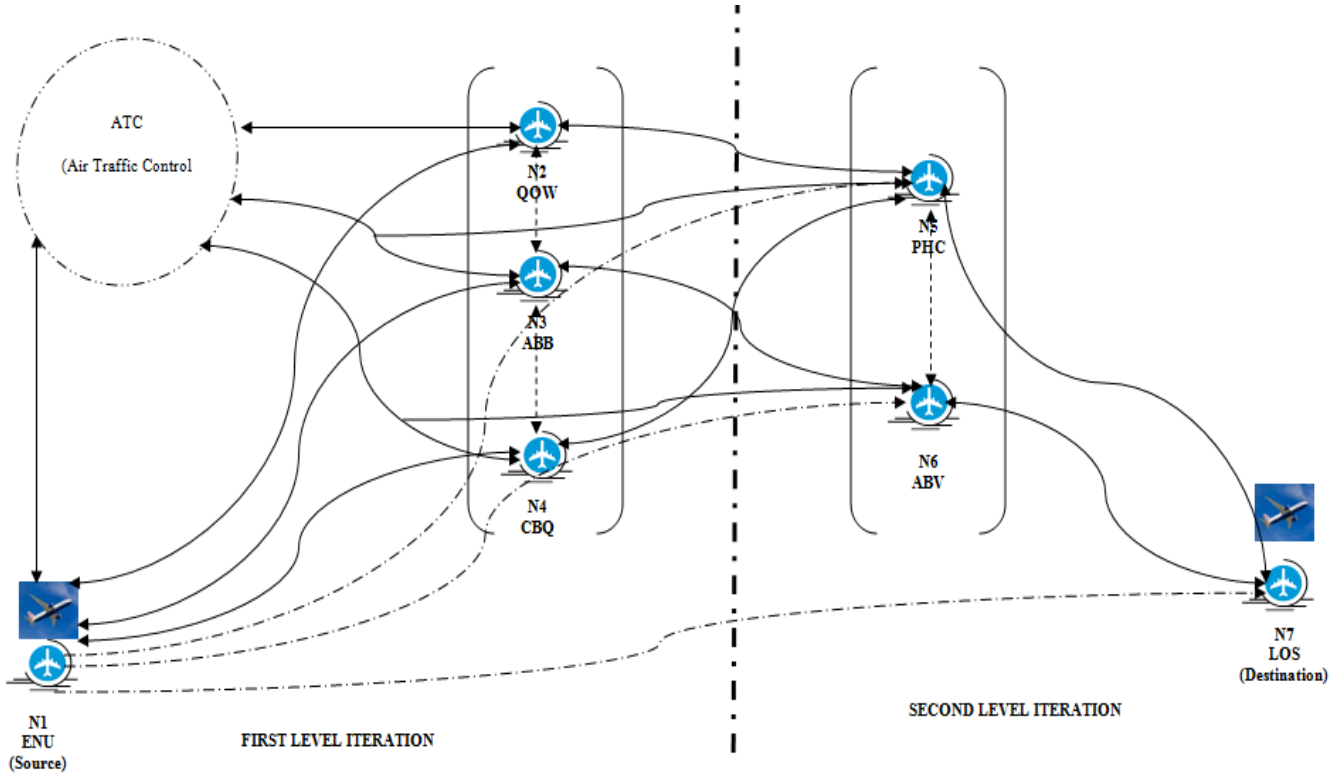

Figure 3: High Level Model of the System

The straight lines from ATC to all the nodes indicate a direct communication between the air traffic controllers to the airports. The straight line from node N1 to N2, N3 and N4 indicates that these nodes are reachable at first iteration while the dotted lines shows that those nodes cannot be reached from the source node. This process continues till the destination node is reached.

C. Permutation and Combination Model

The various ways in which objects from a set may be selected, generally without replacement to form subsets. This selection of subsets is called a permutation when the order of selection is a factor but a combination when order is not a factor.

(Assuming $\{a_1,a_2,...,a_n\}$ is not $\{n ,n-1,...,2, 1\}$)

NextPermutation($a_1,a_2...,a_n$)

Begin

```
{
    j = n - 1;
   while (a[j] > a [j+1])
{
    j--;
 }
    k = n;
   while (a[j] > a[k])
       {
           k--;
       }
Swap(a[j],a[k]);
```

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 2, February 2024
DOI: 10.17148/IJARCCE.2024.13229

```
r = n;
s = j + 1;
while (r > s)
      {
            Swap(a[r],a[s]);
            r--;
            s++;
       }
  }
```

(Assuming $\{a_1, a_2,...,a_r\}$ is not $\{n-r+1,...,n\}$, and $a_i<a_j$ when $i< j$)

```
NextCombination(a1a2...ar)
Begin
    {
       i = r;
       while (a[i] > n - r + i)
         {
              i = i - 1;
          }
            a[i] = a[i] + 1;
            for (j = i + 1; j <= r; j++)
            {
                a[j] = a[i] + j - i;
             }
       }
```

Where a is the sequence or set of air route nodes for which permutations or combinations are being generated, n is the length of the sequence or set, j is the index variable in the 'NextPermutation' algorithm to find the next permutation, I is an index variable used in both algorithms to iterate over elements of the sequence/ set, k is used in the 'NextPermutation' algorithm to find the next permutation, r is the length of the combination being generated and s is used in the 'NextPermutation' algorithm to find the next permutation.

D. Network Model Routing/Implementation

The above named parameters from the air route network system were used to develop the model using the following keynotes:

(i) TL = temporary label of a node
(ii) PL = permanent label of a node
(iii) □ = permanent label of a node
(iv) O = temporary label of a node
(v) * = permanent label of a node
(vi) n = a node in the network *i.e* n_1 = node1, n_2 = node2, n_3 = node3, n_4 = node4, n_5 = node5 and n_6 = node6, n_7 = node7.
(vii) d_{ij} =distance cost between node *i* and *j* in the network
(viii) *a-l* =edge weight (integer varies)

The model algorithm employed the Comparison-Addition Model (CAM) to determine the optimal paths for all the possible shortest paths generated by the permutation and combination model.

The generalized condition for the comparison test is as follows:

IF $TLn_i \leq TLn_j$
$TLn_i \leq TLn_j, \ldots, TLn_i \leq TLn_K$
Then $PL = TLn_i$ otherwise $PL \neq TLn_i$

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 2, February 2024

DOI: 10.17148/IJARCCE.2024.13229

E. Flowchart of the System

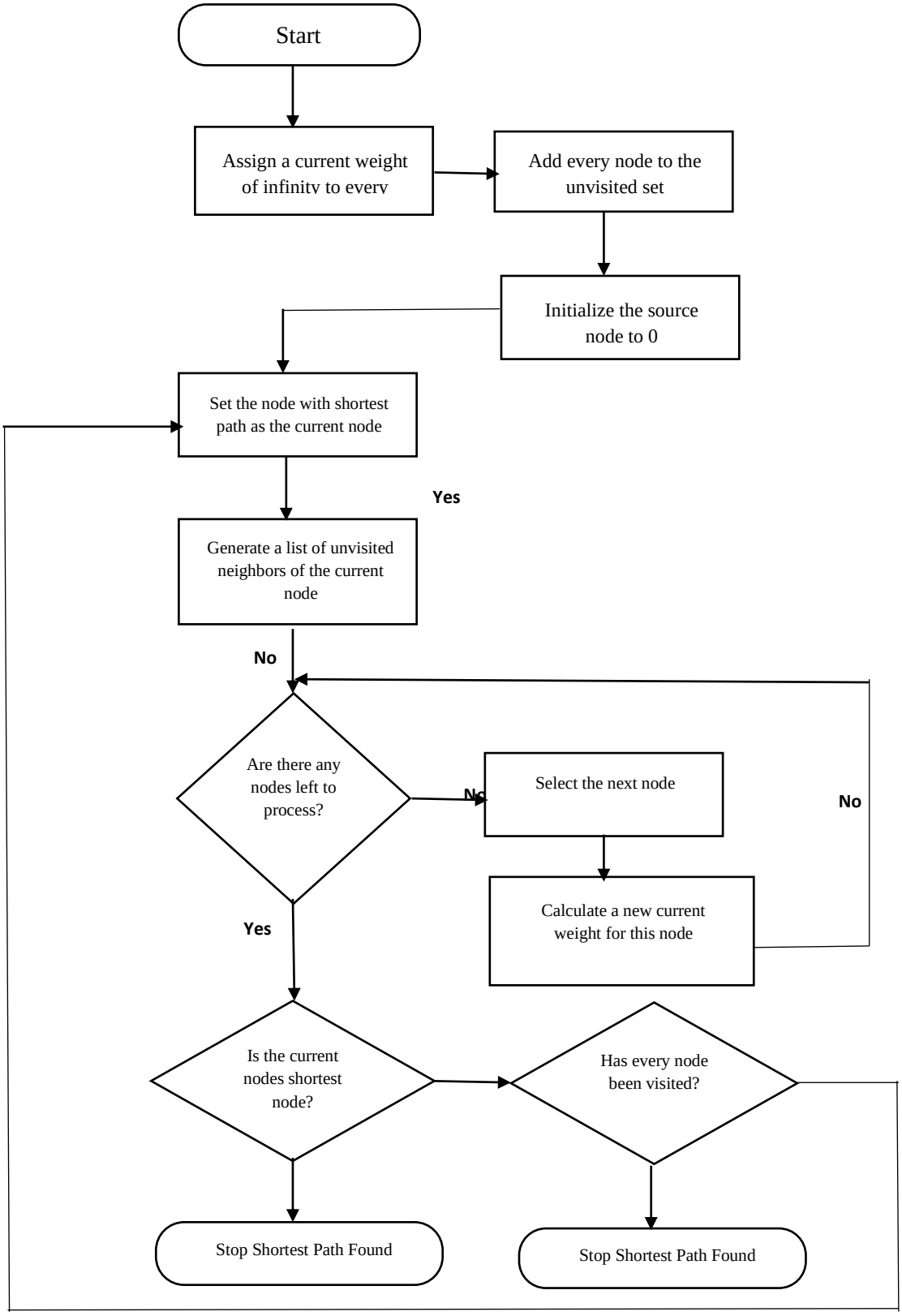

Figure 4: Modified Dijkstra's Algorithm Flowchart

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 2, February 2024

DOI: 10.17148/IJARCCE.2024.13229

V. RESULTS AND DISCUSSION

Performance Evaluation

Seven nodes were presented following the same trends of flight route of the airline within the study area. A trace route command (Set start) was issued from the source to destination which showed the path traversals in the modeled network.

The precision, recall, and F1-score were used to evaluate the performance of Modified Dijkstra's Algorithm over the traditional Dijkstra's algorithm using their respectivegraphs' weights for the 7 airport locations as shown on Tables 1 and 2.Here, the algorithm's decisions to select a particular path is considered as a prediction for either a true positive (TP), false positive (FP), true negative (TN) or false negative (FN).

a) Define Positive and Negative Instances:

Positive instance (class 1): The selected path by the algorithm.
Negative instance (class 0): Other paths not selected by the algorithm.

b) Calculate True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN):

TP: Identify the edges correctly included in the predicted shortest path.
FP: Identify the edges incorrectly included in the predicted shortest path.
TN: Identify the edges correctly excluded from the predicted shortest path
FN: Identify the edges incorrectly from the predicted shortest path

c) Precision:

Precision measures the accuracy of the positive predictions. High precision indicates that when the algorithm selects a path, it is likely to be the correct one

$$Precision = \frac{TruePositives}{(TruePositives + FalsePositives)} \quad \ldots \text{(equ. 1)}$$

d) Recall (Sensitivity or True Positive Rate):

Recall measures the ability of the algorithm to capture all the positive instances. High recall indicates that the algorithm can find most of the optimal paths.

$$Recall = \frac{TruePositives}{(TruePositives + FalseNegatives)} \quad \ldots \text{(equ. 2)}$$

e) F1-Score:

F1-score is the harmonic mean of precision and recall, providing a balance between the two.F1-score ranges from 0 to 1, where 1 is the best possible score.

$$F1\_Score = 2\,x(\frac{\text{precision x recall}}{precision + recall}) \quad \ldots \text{(equ. 3)}$$

Table 1: Precision, Recall and F1_Score for Traditional Dijkstra Algorithm

S/N	LOCATIONS	TP	FP	FN	PRECISION	RECALL	F1_SCORE
1	Enugu (1)	50	35	15	0.59	0.67	0.63
2	Owerri (2)	40	30	10	0.57	0.67	0.61
3	Asaba (3)	90	40	30	0.69	0.75	0.72
4	Calabar (4)	150	80	65	0.65	0.68	0.66
5	PHC (5)	200	90	75	0.69	0.72	0.71
6	Abuja (6)	210	100	95	0.68	0.69	0.69
7	Lagos (7)	95	65	60	0.59	0.63	0.61

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 2, February 2024

DOI: 10.17148/IJARCCE.2024.13229

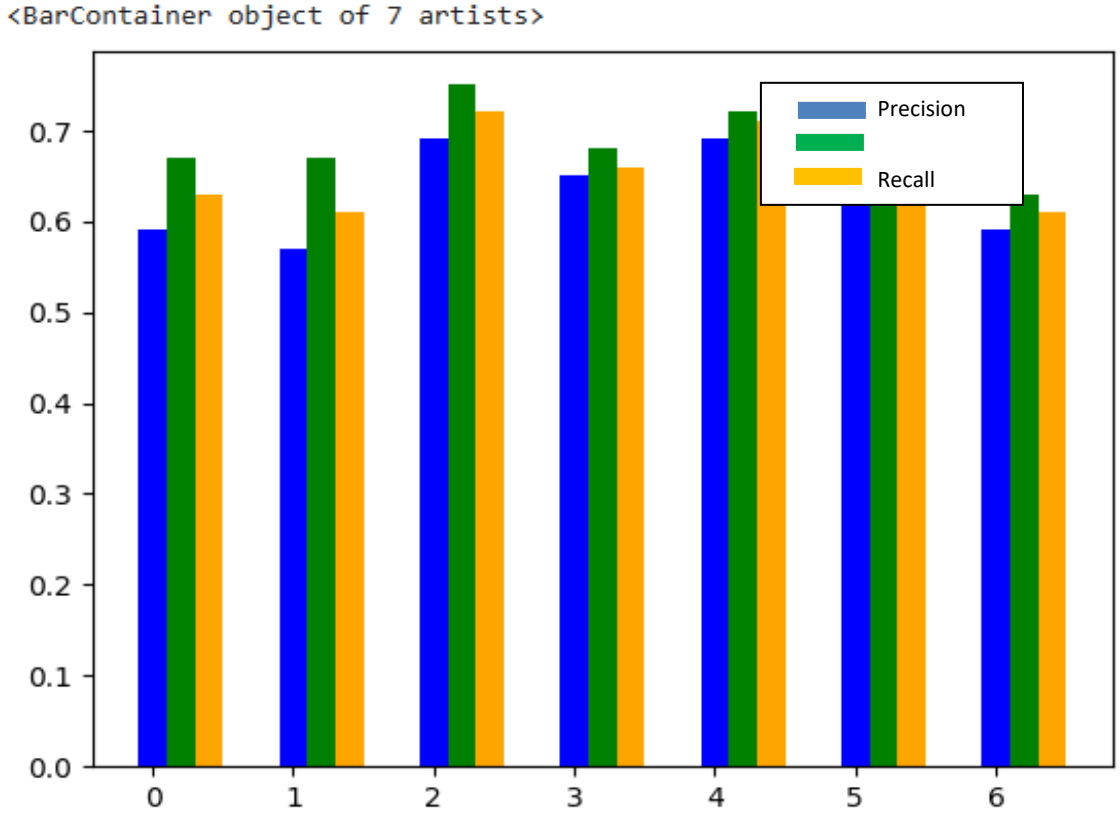

Table 2: Precision, Recall and F1_Score for Modified Dijkstra Algorithm

S/N	LOCATIONS	TP	FP	TN	FN	PRECISION	RECALL	F1_SCORE
1	Enugu (1)	90	30	85	15	0.75	0.86	0.80
2	Owerri (2)	60	20	50	10	0.75	0.86	0.80
3	Asaba (3)	130	40	100	30	0.76	0.81	0.78
4	Calabar (4)	220	85	180	70	0.72	0.76	0.74
5	PHC (5)	290	90	220	75	0.76	0.79	0.78
6	Abuja (6)	250	100	230	95	0.71	0.72	0.73
7	Lagos (7)	120	60	80	50	0.70	0.71	0.70

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 2, February 2024

DOI: 10.17148/IJARCCE.2024.13229

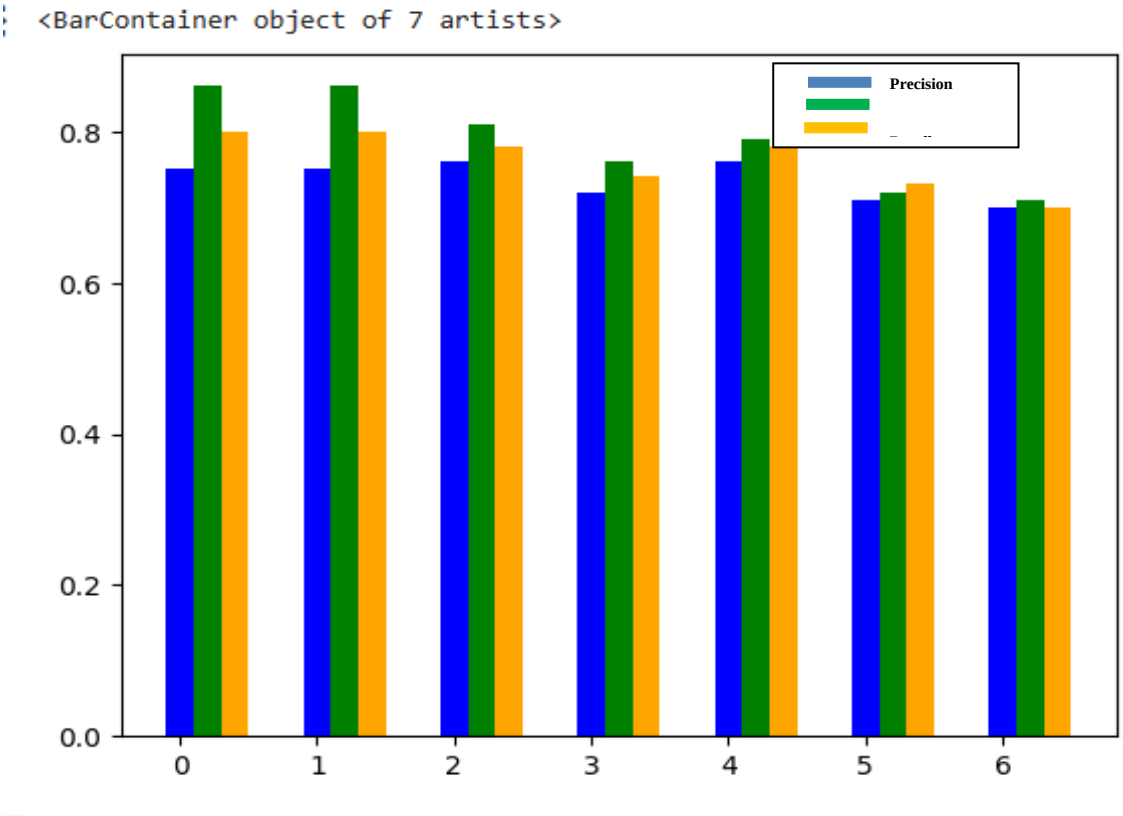

Figure 6: Bar chart of precision, recall, and f1-score for modified Dijkstra's Algorithm

Based on Tables 1 and 2, it can be seen that Modified Dijkstra's algorithm performed better than the traditional Dijkstra's algorithm.

VI. CONCLUSION

An efficient and scheduled air routing coupled with delivery system is the key to meeting the ever-increasing air transportation user's demand.

As one of the main goals of any airline company is to maximize profit as well as reduce cost (fuel and carbon emission), Modified Dijkstra's algorithm achieved this by overcoming known complexity in air delay in either a digragh or undigraph network model for shortest path distance.

REFERENCES

[1]. Abu-Ryash, H., &Tamimi, A. (2015). Comparison studies for different shortest path algorithms. International Journal of Computers and Applications, 14(8), 5979-5986.
[2]. Akpoghomeh, O. S (1999). The Development of Air Transportation in Nigeria, Journal of Transport Geography Volume 7, Issue 2, June http://www.sciencedirect.com/science?-ob. Retrieved 27-05-2021
[3]. Arslan, H. and Manguo˘gl M. (2019). A Hybrid Single-Source Shortest Path Algorithm for graphs. Turkish Journal of Electrical and Computer Sciences. Vol. 27 (2019). No.4. DOI 10.3906/elk-1901-23.
[4]. Federal Airports Authority of Nigeria, FAAN (n.d). Aviation Sector Hand book and Reports (2010–2018, 2019, Q3-Q4 of 2020, Q1 of 2021.Retrieved on Wednesday, October 13, 2021.
[5]. Garnierjm (n.d). Graphs and Dijkstra's Algorithm - http://rollerjm.free.fr/pro/graphs.html. Retrieved on 12/09/23
[6]. Gupta, A. (2017): Advanced Algorithm: Shortest Paths and Seidel's; lecture note. Retrieved from http://www.cs.cmu.edu/ranupamg/analysisIS/lectures/lecture04.pdf
[7]. Jindal Pawan & Amit Kumar (2010). Analysis of shortest path algorithms. Global Journal of Computer Science and Technology, 10(8), - Retrieved from https://computerresearch.org/index.php/computer/article/view/976

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 2, February 2024

DOI: 10.17148/IJARCCE.2024.13229

[8]. Kairanbay Magzhan and Hajar Mat Jani (2013). A Review and Evaluations of Shortest Path Algorithms. International Journal of Scientific & Technology Research • January 2013 - at: https://www.researchgate.net/publication/310594546. Retrieved on Wednesday, October 13, 2021

[9]. Lavlinskaya, O. Y., Kurchenkova, T. V., & Kuripta, O. V. (2020). Shortest path algorithm for graphs in instances of semantic optimization. In Journal of Physics: Conference Series (Vol. 1479, No. 1, p. 012036). IOP Publishing.

[10]. Ofem, O. A., Agana, M. A., & Ejogobe Owai, E. (2017). Optimal power flow-path determination for voltage control in electricity distribution using the modified Dijkstra's algorithm. International Journal of Engineering & Technology, 10(2), 108-115.

[11]. Samui, A. U. (2011). Distribution System Planning Considering Reliable Feeder Routing and Transport Model. International Journal of Innovative Research in Computer and Communication Engineering,3, 50-72.

[12]. Wang Shu-Xi (2012). The Improved Dijkstra's Shortest Path Algorithm and Its Application. 2012 International Workshop on Information and Electronics Engineering (IWIEE) - School of Information Technology, The University of International Business and Economics, Beijing, China, 100029. Retrieved on Wednesday, October 13, 2021.