Mathematical Theory and Modeling
ISSN 2224-5804 (Paper) ISSN 2225-0522 (Online)
Vol.5, No.8, 2015
www.iiste.org

Shortest Transportation Route Network in Nigeria Using Floyd-Warshall's Algorithm

Esuabana, Ita Micah[1] Ikpang, Ikpang Nkereuwem[2] Okon, Ekom-obong Jackson[1]

1. Department of Mathematics/Statistics, Cross River University of Technology, Calabar, Cross River State, Nigeria

2. Department of Statistics, University of Calabar, Calabar, Cross River State, Nigeria

Abstract:

This study presents the application of Floyd – Warshall algorithm, which is an all pairs shortest path algorithm in finding the shortest route network for major cities in Nigeria. Twenty one (21) city route networks in Nigeria showing distances (km) are considered with Calabar, Cross River State as the origin node and Kaduna, Kaduna State as the sink node. Distance and precedence matrices are computed for all iterations to obtain the weight between nodes in the network and the shortest route respectively. The optimal route for all pairs in the network and the total distance travelled from one node to another are obtained respectively from the precedence and distance matrices of the final iteration. Detailed results showed that the algorithm is efficient. The designed route network shows the shortest route for all pairs of nodes in the network and also exposes hidden shortest routes. These routes are recommended for inter-city transportation in Nigeria.

Keywords: Floyd-Warshall algorithm, optimal shortest path, shortest distances, route determination

1. Introduction

Efficient transportation route network is a challenge to some developing countries and Nigeria as well. Optimization of networks is seen to be the heart of operations research. Shortest route model is one of the network models whose application covers a wide range of areas, such as telecommunication, electric navigation, geographic information system, project planning, traffic, tourism, transportation planning, pipeline network designs and so on, (Jahan & Hasan, 2011). The greatest problem arising in network analysis is shortest path analysis. Shortest route also extends to other measurement such as distance, time, cost, and capacity of line. With the growing population and development of new roads in Nigeria, it is necessary to develop an optimal shortest route network which will determine the shortest routes from one city to another. Man, nations, regions and the world would be severely limited in development without transportation, which is a key factor for physical and economic growth, (Oyesiku, 2002).

The travel of the passengers and transport requirements for respective trips are different. For example a business trip passenger will require that time of travel is as short as possible while tourists prefer the cost of travel as minimum as possible. National City traffic advisory procedures can satisfy the passengers demand and provide two to three optimal decision rules according to the transport advisory for passengers. The procedures allow users to look up to city information, choose the optimal decision rule and edit city information, (Sangaiah *et al.*, 2014)

In todays civilized society, economic development and the vast increase in the number of cars has made traffic jams become more serious. This does not only cause inconveniencies for travellers, but also brings about underestimation of the nation's economic growth. City traffic congestion has become a bottleneck for constraining the development of medium and large cities at home and abroad (Wei, 2010). This does not only cause economic loss, but also contributes to the reason for the maximum growth of traffic accidents, energy waste and environmental hazards. According to Syslo *et al.* (1983), shortest path problems are the most fundamental and the most commonly encountered problems in the study of transportation and communication networks. They are classical problems in network analysis and these models can be used in solving many optimization problems, such as designing efficient service route networks, plant layout, and also line arrangement.

Finding the shortest path is an important task in many network and transportation related analysis. This problem arises as a main decision question or as a step in some situation. There are many variations depending on type of network, cost involved and source/destination pairs of nodes for which we need solution (Rardin, 2003). It is therefore important to determine the specificity of the shortest path problem we are concerned with. Sniedovich (2005), classifies in depth shortest path problems and also summarizes several variants of the shortest path as follows:

i. Cyclic or Acyclic problems: If there is at least one cycle in the network it is called a cyclic network otherwise acyclic.
ii. Non-negative or Negative distance problems: If the distances are non-negative, that is, $d_{ij} \geq 0$ for all i and j, or if there is at least one negative distance, that is, $d_{ij} < 0$
iii. Non-negative cyclic or Negative cyclic problems: If the cyclic problems have non-negative length of all cycles or if the length of at least one cycle is negative.

In Nigeria for instance, a major challenge is that of deciding the route to take from the origin to destination while minimizing cost, total travel distance, fuel and time consumption. Overcoming this challenge however requires a sophisticated knowledge of the optimal public transport network. Several algorithms can be used to determine the shortest distance and shortest route between two nodes in a network, but this study employs the classical and recursive Floyd–Warshall algorithm. This study will also expose those hidden and in some cases neglected shortest routes unknown to transporters and the nation as well.

2. Methodology

Intercity distances (km) for twenty one (21) cities with respect to the origin node used in this study were collected and the Floyd-Warshall algorithm was employed to find the optimal routes for all pairs in the network.

The Floyd-Warshall algorithm is a dynamic programming algorithm to solve the all-pairs (all-to-all) shortest path problem on a directed network. The arcs of the network may have negative costs but must not have any negative-costs cycles. The Floyd-Warshall algorithm compares all possible paths through the graph between each pair of vertices. It is able to do this in $\Theta(|v|^3)$ comparison in a graph. The algorithm is based on inductive arguments developed by an application of a dynamic programming technique. It will compute the shortest path between all possible pairs of vertices in a (possibly weighted) graph or digraph simultaneously in time. In this problem we want the minimum routes between all the pairs of peaks. Floyd's algorithm calculates the costs of the shortest path between each pair of vertices in $\Theta(|v|^3)$ time. It consists of three nested loops. The invariant of the outer loop is the key to the algorithm. When the iteration starts, P holds the optimal path length from v_i to v_j for each i and j, considering only paths that go direct or via vertices v_n for $n < k$. This is true initially when $k = 1$ and holds for only direct paths.

At any iteration, the next value of k is considered. A better path may exist possibly from v_i to v_j via this new v_k while noting that it will visit v_k at most once. This means that it is sufficient to consider paths from v_i through v_k possibly via $\{v_1,...,v_{k-1}\}$ and paths from v_k through v_j possibly via $\{v_1,...,v_{k-1}\}$ hence invariance is maintained. Finally, P holds optimal path lengths for unrestricted paths. In simple terms, the Floyd-Warshall algorithm obtains a matrix of shortest paths distances within $\Theta(n^3)$ computations which represents an n- nodes network as a square matrix with n- rows and n- columns. The (i, j) entries of the matrix represent the distance from node i to node j, which is finite if i is linked directly to j and infinite otherwise.

Given three nodes i, j and k with connecting distances, it will be shorter to reach k from i passing through j *iff* $d_{ij} + d_{jk} < d_{ik}$. In this case, it is optimal to replace the direct route from $i \rightarrow k$ with the indirect route $i \rightarrow j \rightarrow k$. The Floyd-Warshall algorithm takes as input an adjacency matrix representation and maintains two types of matrices; the distance matrix D^0 and initial precedence matrix U^0 as input. The algorithm proceeds for n- iterations where n is the number of nodes in the distance matrix. The n^{th} iteration gives the optimal/final distance matrix $D^{k=n}$ and the final precedence matrix $U^{k=n}$. The optimal distance matrix D^n represents the shortest distance between any two nodes in the network and the corresponding shortest paths can be traced out from the final precedence matrix U^n.

The Floyd-Warshall algorithm therefore is as follows:

Step 1: Form the initial distance matrix D^0 and the initial precedence matrix U^0. For all arcs/edges (i, j) in the network, initialize $d_{ij}^{(0)} = w_{ij,}$ $u_{ij} = i$ while for i, j pairs without arcs/edges (i, j), assign $d_{ij}^{(0)} = 0$ if $i = j$ and $+\infty$ otherwise.

Step 2: Set the iteration counter $k = 1$

Step 3: Find the value of the distance matrix D^k for all $i, j \neq k$ using the relation; $d_{ij}^{(k)} = \min\{d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\}$

Step 4: Assign values to the precedence matrix U^k by replacing u_{ij} (where i, j is the double boxed cell in U^{k-1}) with k

Step 5: Terminate if

i. $k = n$ (where n is the number of nodes in the network)
ii. $d_{ii}{}^{k} < 0$ for any node i

Where for the case in (i), $d_{ij}^{(k)}$ is equal to the required shortest distances while for the case in (ii), evidence of any negative cycle through i is detected.

Step 6: If $k < n$ and all $d_{ii}^{\ k} > 0$, set $k = k + 1$ and go to step 3.

Step 7: After n steps, we can determine the shortest route(s) between nodes i and nodes j from the matrices D^n and U^n using the following rules;

i. From D^n, $d_{ij}^{(k)}$ gives the shortest distance between nodes i and j.

ii. From U^n, determine the intermediate node $k = u_{ij}$ that yields the route $i \rightarrow j \rightarrow k$. If $u_{ik} = k$ and $u_{kj} = j$, stop; all the intermediate nodes have been found. Otherwise, repeat the procedure between nodes i and k, and between nodes k and j.

3. Implementation

The major city route network with their distances (km) is shown in Fig. 1. The cities listed below represent the nodes in the network in the order in which they are listed.

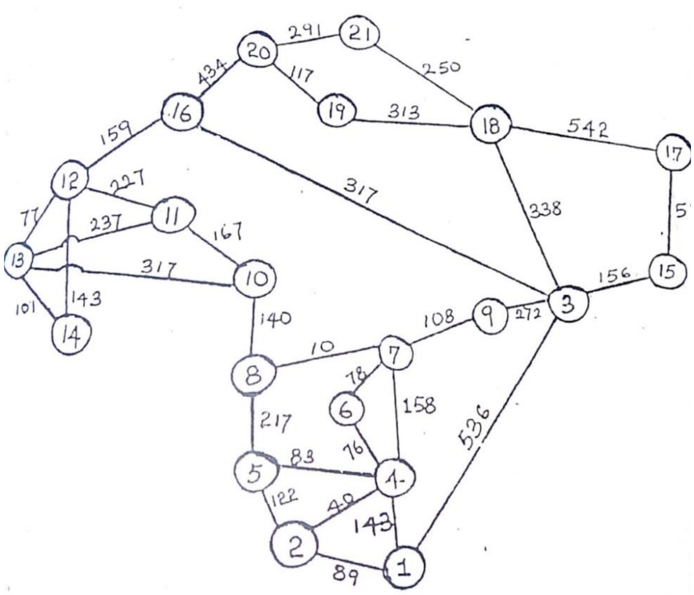

Figure 1. Twenty-one major city route network in Nigeria

1. Calabar	8. Asaba	15. Jalingo
2. Uyo	9. Enugu	16. Ilorin
3. Makurdi	10. Benin	17. Yola
4. Aba	11. Akure	18. Jos
5. Port Harcourt	12. Ibadan	19. Abuja
6. Owerri	13. Abeokuta	20. Minna
7. Onitsha	14. Lagos	21. Kaduna

To achieve determination of shortest route between all pairs of nodes in the network, the analysis shall be carried out for all the nodes in the network. The resulting matrices are symmetrical since traffic is allowed to flow in both directions. That is, $d_{ij}^{k} = d_{ji}^{k}$.

3.1 Iteration 0

The initial distance matrix D^0 as shown in Fig. 2 and the initial precedence matrix U^0 as shown in Fig. 3 represents the initial presentation of the network. $d_{ij} = +\infty$ implies that no traffic is allowed from node i to node j and $d_{ij} = 0$ implies that node i is equal to node j.

Iteration 0 (Distance Matrix): D^0

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	0	89	536	143	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞
2	89	0	+∞	40	122	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞
3	536	+∞	0	+∞	+∞	+∞	+∞	+∞	272	+∞	+∞	+∞	+∞	+∞	156	317	+∞	338	+∞	+∞	+∞
4	143	40	+∞	0	83	76	158	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞
5	+∞	122	+∞	83	0	+∞	+∞	217	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞
6	+∞	+∞	+∞	76	+∞	0	78	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞
7	+∞	+∞	+∞	158	+∞	78	0	10	108	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞
8	+∞	+∞	+∞	+∞	217	+∞	10	0	+∞	140	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞
9	+∞	+∞	272	+∞	+∞	+∞	108	+∞	0	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞
10	+∞	+∞	+∞	+∞	+∞	+∞	+∞	140	+∞	0	167	+∞	317	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞
11	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	167	0	227	237	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞
12	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	227	0	77	143	+∞	159	+∞	+∞	+∞	+∞	+∞
13	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	317	237	77	0	101	+∞	+∞	+∞	+∞	+∞	+∞	+∞
14	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	143	101	0	+∞	+∞	+∞	+∞	+∞	+∞	+∞
15	+∞	+∞	156	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	0	+∞	520	+∞	+∞	+∞	+∞
16	+∞	+∞	317	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	159	+∞	+∞	+∞	0	+∞	+∞	+∞	434	+∞
17	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	520	+∞	0	542	+∞	+∞	+∞
18	+∞	+∞	338	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	542	0	313	+∞	250
19	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	313	0	117	+∞
20	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	434	+∞	+∞	117	0	291
21	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	250	+∞	291	0

Figure 2. Distance matrix for iteration 0

Iteration 0 (Precedence Matrix): U^0

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	0	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
2	1	0	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
3	1	2	0	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
4	1	2	3	0	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
5	1	2	3	4	0	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
6	1	2	3	4	5	0	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
7	1	2	3	4	5	6	0	8	9	10	11	12	13	14	15	16	17	18	19	20	21
8	1	2	3	4	5	6	7	0	9	10	11	12	13	14	15	16	17	18	19	20	21
9	1	2	3	4	5	6	7	8	0	10	11	12	13	14	15	16	17	18	19	20	21
10	1	2	3	4	5	6	7	8	9	0	11	12	13	14	15	16	17	18	19	20	21
11	1	2	3	4	5	6	7	8	9	10	0	12	13	14	15	16	17	18	19	20	21
12	1	2	3	4	5	6	7	8	9	10	11	0	13	14	15	16	17	18	19	20	21
13	1	2	3	4	5	6	7	8	9	10	11	12	0	14	15	16	17	18	19	20	21
14	1	2	3	4	5	6	7	8	9	10	11	12	13	0	15	16	17	18	19	20	21
15	1	2	3	4	5	6	7	8	9	10	11	12	13	14	0	16	17	18	19	20	21
16	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	0	17	18	19	20	21
17	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	0	18	19	20	21
18	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	0	19	20	21
19	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	0	20	21
20	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	0	21
21	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	0

Figure 3. Precedence matrix for iteration 0

3.2 Iteration 1

Setting $k = 1$; the pivot row and pivot column are highlighted by the shaded first row and first column in the initial distance matrix D^0 since all the entries in the pivot column is $+\infty$. The double-boxed element d_{23} and d_{34} are the only ones that can be improved using the triple operation. Thus to obtain D^1 and U^1 from D^0 and U^0;

1. Replace $d_{23}^{\ 1}$ with $\min\left(d_{23}^{\ 0}, d_{21}^{\ 0} + d_{13}^{\ 0}\right)$ = min ($+\infty$, 89 + 536) = 625 and set $u_{23}^{\ 1} = 1$
2. Replace $d_{34}^{\ 1}$ with $\min\left(d_{34}^{\ 0}, d_{32}^{\ 0} + d_{24}^{\ 0}\right)$ = min (665, 625 + 40) = 665 and set $u_{34}^{\ 1} = 1$

These changes are shown in bold faces in matrices D^1 and U^1 as shown in Fig. 4 and Fig. 5 respectively.

Iteration 1 (Distance Matrix): D^1

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	0	89	536	143	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
2	89	0	625	40	122	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
3	536	625	0	665	$+\infty$	$+\infty$	$+\infty$	$+\infty$	272	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	156	317	$+\infty$	338	$+\infty$	$+\infty$	$+\infty$
4	143	40	665	0	83	76	158	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
5	$+\infty$	122	$+\infty$	83	0	$+\infty$	$+\infty$	217	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
6	$+\infty$	$+\infty$	$+\infty$	76	$+\infty$	0	78	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
7	$+\infty$	$+\infty$	$+\infty$	158	$+\infty$	78	0	10	108	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
8	$+\infty$	$+\infty$	$+\infty$	$+\infty$	217	$+\infty$	10	0	$+\infty$	140	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
9	$+\infty$	$+\infty$	272	$+\infty$	$+\infty$	$+\infty$	108	$+\infty$	0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
10	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	140	$+\infty$	0	167	$+\infty$	317	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
11	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	167	0	227	237	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
12	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	227	0	77	143	$+\infty$	159	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
13	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	317	237	77	0	101	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
14	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	143	101	0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
15	$+\infty$	$+\infty$	156	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	0	$+\infty$	520	$+\infty$	$+\infty$	$+\infty$	$+\infty$
16	$+\infty$	$+\infty$	317	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	159	$+\infty$	$+\infty$	$+\infty$	0	$+\infty$	$+\infty$	$+\infty$	434	$+\infty$
17	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	520	$+\infty$	0	542	$+\infty$	$+\infty$	$+\infty$
18	$+\infty$	$+\infty$	338	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	542	0	313	$+\infty$	250
19	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	313	0	117	$+\infty$
20	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	434	$+\infty$	$+\infty$	117	0	291
21	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	250	$+\infty$	291	0

Figure 4. Distance matrix for iteration 1

Iteration 1 (Precedence Matrix): U^1

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	0	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
2	1	0	1	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
3	1	1	0	1	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
4	1	2	1	0	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
5	1	2	3	4	0	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
6	1	2	3	4	5	0	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
7	1	2	3	4	5	6	0	8	9	10	11	12	13	14	15	16	17	18	19	20	21
8	1	2	3	4	5	6	7	0	9	10	11	12	13	14	15	16	17	18	19	20	21
9	1	2	3	4	5	6	7	8	0	10	11	12	13	14	15	16	17	18	19	20	21
10	1	2	3	4	5	6	7	8	9	0	11	12	13	14	15	16	17	18	19	20	21
11	1	2	3	4	5	6	7	8	9	10	0	12	13	14	15	16	17	18	19	20	21
12	1	2	3	4	5	6	7	8	9	10	11	0	13	14	15	16	17	18	19	20	21
13	1	2	3	4	5	6	7	8	9	10	11	12	0	14	15	16	17	18	19	20	21
14	1	2	3	4	5	6	7	8	9	10	11	12	13	0	15	16	17	18	19	20	21
15	1	2	3	4	5	6	7	8	9	10	11	12	13	14	0	16	17	18	19	20	21
16	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	0	17	18	19	20	21
17	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	0	18	19	20	21
18	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	0	19	20	21
19	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	0	20	21
20	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	0	21
21	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	0

Figure 5. Precedence matrix for iteration 1

3.3 *Iteration 2*

Setting $k = 2$; as highlighted by the shaded row and column in the distance matrix D^1. The triple operation is applied to the double-boxed element in D^1 and U^1 .The changes are shown in D^2 and U^2 as seen in Fig. 6 and Fig. 7 respectively. Thus we obtain D^2 and U^2 from D^1 and U^1 as follows; Replace $d_{14}^{\ 2}$ with $\min\left(d_{14}^{\ 1}, d_{12}^{\ 1} + d_{24}^{\ 1}\right)$ = min (143, 89 + 40) = 129 and set $u_{14}^{\ 2} = 2$. Since traffic is allowed to flow in both directions, $d_{41}^{\ 2} = 129$

1. Replace $d_{15}^{\ 2}$ with $\min\left(d_{15}^{\ 1}, d_{12}^{\ 1} + d_{25}^{\ 1}\right) = \min(+\infty, 89 + 122) = 211$and set $u_{15}^{\ 2} = 2$. Similarly, (3, 4) = 665, (3, 5) = 747. These changes are shown in bold faces in matrices D^2 and U^2 (see Fig. 6 and Fig. 7 respectively). Then for the precedence matrix, set $u^2 = 2$ at all stages of improvement.

Iteration 2 (Distance Matrix): D^2

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	0	89	536	129	211	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞
2	89	0	625	40	122	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞
3	536	625	0	665	747	+∞	+∞	+∞	272	+∞	+∞	+∞	+∞	+∞	156	317	+∞	338	+∞	+∞	+∞
4	129	40	665	0	83	76	158	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞
5	211	122	747	83	0	+∞	+∞	217	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞
6	+∞	+∞	+∞	76	+∞	0	78	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞
7	+∞	+∞	+∞	158	+∞	78	0	10	108	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞
8	+∞	+∞	+∞	+∞	217	+∞	10	0	+∞	140	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞
9	+∞	+∞	272	+∞	+∞	+∞	108	+∞	0	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞
10	+∞	+∞	+∞	+∞	+∞	+∞	+∞	140	+∞	0	167	+∞	317	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞
11	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	167	0	227	237	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞
12	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	227	0	77	143	+∞	159	+∞	+∞	+∞	+∞	+∞
13	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	317	237	77	0	101	+∞	+∞	+∞	+∞	+∞	+∞	+∞
14	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	143	101	0	+∞	+∞	+∞	+∞	+∞	+∞	+∞
15	+∞	+∞	156	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	0	+∞	520	+∞	+∞	+∞	+∞
16	+∞	+∞	317	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	159	+∞	+∞	+∞	0	+∞	+∞	+∞	434	+∞
17	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	520	+∞	0	542	+∞	+∞	+∞
18	+∞	+∞	338	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	542	0	313	+∞	250
19	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	313	0	117	+∞
20	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	434	+∞	+∞	117	0	291
21	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	250	+∞	291	0

Figure 6. Distance matrix for iteration 2

Iteration 2 (Precedence Matrix): U^2

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	0	2	3	2	2	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
2	1	0	1	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
3	1	1	0	1	2	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
4	2	2	1	0	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
5	2	2	2	4	0	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
6	1	2	3	4	5	0	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
7	1	2	3	4	5	6	0	8	9	10	11	12	13	14	15	16	17	18	19	20	21
8	1	2	3	4	5	6	7	0	9	10	11	12	13	14	15	16	17	18	19	20	21
9	1	2	3	4	5	6	7	8	0	10	11	12	13	14	15	16	17	18	19	20	21
10	1	2	3	4	5	6	7	8	9	0	11	12	13	14	15	16	17	18	19	20	21
11	1	2	3	4	5	6	7	8	9	10	0	12	13	14	15	16	17	18	19	20	21
12	1	2	3	4	5	6	7	8	9	10	11	0	13	14	15	16	17	18	19	20	21
13	1	2	3	4	5	6	7	8	9	10	11	12	0	14	15	16	17	18	19	20	21
14	1	2	3	4	5	6	7	8	9	10	11	12	13	0	15	16	17	18	19	20	21
15	1	2	3	4	5	6	7	8	9	10	11	12	13	14	0	16	17	18	19	20	21
16	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	0	17	18	19	20	21
17	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	0	18	19	20	21
18	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	0	19	20	21
19	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	0	20	21
20	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	0	21
21	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	0

Figure 7. Precedence matrix for iteration 2

The iteration process continues and necessary improvements are made until the final (21^{st}) iteration. This iteration will produce the final matrices where none of the d_{ij} entries can be improved by the triple operation.

3.4 Iteration 21

Set k = 21, as will be shown in the shaded row and column of the distance matrix D^{20} obtained. No changes are made in the last iteration since it is the last stage of the network analysis. This iteration also provides the shortest distance from the distance matrix and shortest routes from the precedence matrix in the network, hence the solution. The final distance matrix D^{21} and the final precedence matrix U^{21} for this iteration are shown in Fig. 8 and Fig. 9 respectively.

Final Iteration (Distance Matrix): D^{21}

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	0	89	536	129	211	205	283	293	391	433	600	827	750	851	692	853	1212	874	1187	1287	1124
2	89	0	574	40	122	116	194	204	302	344	511	738	661	762	730	891	1250	912	1225	1325	1162
3	536	574	0	534	607	458	380	390	272	530	697	476	553	619	156	317	676	338	651	751	588
4	129	40	534	0	83	76	154	164	262	304	471	698	621	722	690	851	1210	872	1185	1285	1122
5	211	122	607	83	0	159	227	217	335	357	524	751	674	775	763	910	1283	945	1258	1344	1195
6	205	116	458	76	159	0	78	88	186	228	395	622	545	646	614	775	1134	796	1109	1209	1046
7	283	194	380	154	227	78	0	10	108	150	317	544	467	568	536	697	1056	718	1031	1131	968
8	293	204	390	164	217	88	10	0	118	140	307	534	457	558	546	693	1066	728	1041	1127	978
9	391	302	272	262	335	186	108	118	0	258	425	652	575	676	428	589	948	610	923	1023	860
10	433	344	530	304	357	228	150	140	258	0	167	394	317	418	686	553	1206	868	1104	987	1118
11	600	511	697	471	524	395	317	307	425	167	0	227	237	338	853	386	1373	1034	937	820	1111
12	827	738	476	698	751	622	544	534	652	394	227	0	77	143	632	159	1152	814	710	593	884
13	750	661	553	621	674	545	467	457	575	317	237	77	0	101	709	236	1229	891	787	670	961
14	851	762	619	722	775	646	568	558	676	418	338	143	101	0	775	302	1295	957	853	736	1027
15	692	730	156	690	763	614	536	546	428	686	853	632	709	775	0	473	520	494	807	907	744
16	853	891	317	851	924	775	697	707	589	847	1014	159	236	302	473	0	993	655	551	434	725
17	1212	1250	676	1210	1283	1134	1056	1066	948	1206	1373	1152	1229	1295	520	993	0	542	855	972	792
18	874	912	338	872	945	796	718	728	610	868	1034	814	891	957	494	655	542	0	313	430	250
19	1187	1225	651	1185	1258	1109	1031	1041	923	1104	937	710	787	853	807	551	855	313	0	117	408
20	1287	1325	751	1285	1344	1209	1131	1127	1023	987	820	593	670	736	907	434	972	430	117	0	291
21	1124	1162	588	1122	1195	1046	968	978	860	1118	1111	884	961	1027	744	725	792	250	408	291	0

Figure 8. Distance matrix for the final iteration

Final Iteration (Precedence Matrix): U^{21}

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	0	2	3	2	2	4	6	7	7	8	10	11	10	13	3	3	15	3	18	16	18
2	1	0	9	4	5	4	6	7	7	8	10	11	10	13	9	9	15	9	18	16	18
3	1	9	0	9	9	9	9	9	9	9	10	16	16	16	15	16	15	18	18	16	18
4	2	2	9	0	5	6	6	7	7	8	10	11	10	13	9	9	15	9	18	16	18
5	2	2	9	4	0	4	8	8	8	8	10	11	10	13	9	12	15	9	18	16	18
6	4	4	9	4	4	0	7	7	7	8	10	11	10	13	9	9	15	9	18	16	18
7	6	6	9	6	8	6	0	8	9	8	10	11	10	13	9	9	15	9	18	16	18
8	7	7	9	7	5	7	7	0	7	10	10	11	10	13	9	12	15	9	18	16	18
9	7	7	3	7	8	7	7	7	0	8	10	11	10	13	3	3	15	3	18	16	18
10	8	8	9	8	8	8	8	8	8	0	11	11	13	13	9	12	15	9	20	16	18
11	10	10	10	10	10	10	10	10	10	10	0	12	13	13	10	12	15	10	20	16	20
12	13	13	16	13	13	13	13	13	13	13	13	0	13	14	16	16	16	16	20	16	20
13	10	10	16	10	10	10	10	10	10	10	11	12	0	14	16	12	16	16	20	16	20
14	12	12	16	12	12	12	12	12	12	12	12	12	13	0	16	12	16	16	20	16	20
15	3	9	3	9	9	9	9	9	3	9	10	16	16	16	0	3	17	3	18	16	18
16	3	9	3	9	12	9	9	12	3	12	12	12	12	12	3	0	15	3	20	20	20
17	15	15	15	15	15	15	15	15	15	15	15	16	16	16	15	15	0	18	18	19	18
18	3	9	3	9	9	9	9	9	3	9	10	16	16	16	3	3	17	0	19	19	21
19	18	18	18	18	18	18	18	18	18	20	20	20	20	20	18	20	18	18	0	20	20
20	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	19	19	19	0	21
21	18	18	18	18	18	18	18	18	18	18	20	20	20	20	18	20	18	18	20	20	0

Figure 9. Precedence matrix for the final iteration

3.5 Summary of Results

Summary of the results obtained from the final iteration of the Floyd- Warshall algorithm to determine the shortest route is shown in Table 1. The procedure did not only determine the shortest route from the origin node to the sink node, but also determines all shortest distances and routes from any node of origin to any sink node in the network.

The solution of all-to-all shortest path problem in road network of the cities considered in this study can be found in the final iteration matrix formulation for distance and precedence matrix respectively. For example the shortest distance from Kaduna (node 21) to Enugu (node 9) is 860km and its corresponding route recovered from final precedence matrix is 21-18-3-9 i.e. Kaduna- Jos – Makurdi –Enugu.

Table 1. Table showing all-pairs shortest route in Nigeria

From	To	Approximate distance(km)	Shortest routes
1	2	89	1-2
1	3	536	1-3
1	4	129	1-2-4
1	5	211	1-2-5
1	6	205	1-2-4-6
1	7	283	1-2-4-6-7
1	8	293	1-2-4-6-7-8
1	9	391	1-2-4-6-7-9
1	10	433	1-2-4-6-7-8-10
1	11	600	1-2-4-6-7-8-10-11
1	12	827	1-2-4-6-7-8-10-11-12
1	13	750	1-2-4-6-7-8-10-13
1	14	851	1-2-4-6-7-8-10-13-14
1	15	692	1-3-15
1	16	853	1-3-16
1	17	1212	1-3-15-17
1	18	874	1-3-18
1	19	1187	1-3-18-19
1	20	1287	1-3-16-20
1	21	1124	1-3-18-21
2	1	89	2-1
2	3	574	2-4 -6-7-9-3
2	4	40	2-4
2	5	122	2-5
2	6	116	2-4-6
2	7	194	2-4-6-7
2	8	204	2-4-6-7-8
2	9	302	2-4-6-7-9
2	10	344	2-4-6-7-8-10
2	11	511	2-4-6-7-8-10-11
2	12	738	2-4-6-7-8-10-11-12
2	13	661	2-4-6-7-8-10-13
2	14	762	2-4-6-7-8-10-13-14
2	15	730	2-4-6-7-9-3-15
2	16	891	2-4-6-7-9-3-16
2	17	1250	2-4-6-7-9-3-15-17
2	18	912	2-4-6-7-9-3-18
2	19	1225	2-4-6-7-9-3-18-19
2	20	1325	2-4-6-7-9-3-16-20
2	21	1162	2-4-6-7-9-3-18-21
3	1	536	3-1
3	2	574	3-9-7-6-4-2
3	4	534	3-9-7-6-4
3	5	607	3-9-7-6-5
3	6	458	3-9-7-6
3	7	380	3-9-7
3	8	390	3-9-7-8
3	9	272	3-9

3	10	530	3-9-7-8-10
3	11	697	3-9-7-8-10-11
3	12	476	3-16-12
3	13	553	3-16-12-13
3	14	619	3-16-12-14
3	15	156	3-15
3	16	317	3-16
3	17	676	3-15-17
3	18	338	3-18
3	19	651	3-18-19
3	20	751	3-16-20
3	21	588	3-18-21
4	1	129	4-2-1
4	2	40	4-2
4	3	534	4-6-7-9-3
4	5	83	4-5
4	6	76	4-6
4	7	154	4-6-7
4	8	164	4-6-7-8
4	9	262	4-6-7-9
4	10	304	4-6-7-8-10
4	11	471	4-6-7-8-10-11
4	12	698	4-6-7-8-10-11-12
4	13	621	4-6-7-8-10-13
4	14	722	4-6-7-8-10-13-14
4	15	690	4-6-7-9-3-15
4	16	851	4-6-7-9-3-16
4	17	1210	4-6-7-9-3-15-17
4	18	872	4-6-7-9-3-18
4	19	1185	4-6-7-9-3-18-19
4	20	1285	4-6-7-9-3-18-20
4	21	1122	4-6-7-9-3-18-21
5	1	211	5-2-1
5	2	122	5-2
5	3	607	5-8-7-9-3
5	4	83	5-4
5	6	159	5-4-6
5	7	227	5-8-7
5	8	217	5-8
5	9	335	5-8-7-9
5	10	357	5-8-10
5	11	524	5-8-10-11
5	12	751	5-8-10-11-12
5	13	674	5-8-10-13
5	14	775	5-8-10-13-14
5	15	763	5-8-7-9-3-15
5	16	910	5-8-10-11-12-16
5	17	1283	5-8-7-9-3-15-17
5	18	945	5-8-7-9-3-18
5	19	1258	5-8-7-9-3-18-19
5	20	1344	5-8-10-11-12-16-20
5	21	1195	5-8-7-9-3-18-21
6	1	205	6-4-2-1
6	2	116	6-4-2
6	3	456	6-7-9-3
6	4	76	6-4
6	5	159	6-4-5
6	7	78	6-7

6	8	88	6-7-8
6	9	186	6-7-9
6	10	228	6-7-8-10
6	11	395	6-7-8-10-11
6	12	622	6-7-8-10-11-12
6	13	545	6-7-8-10-13
6	14	646	6-7-8-10-13-14
6	15	614	6-7-9-3-15
6	16	775	6-7-9-3-16
6	17	1134	6-7-9-3-15-17
6	18	796	6-7-9-3-18
6	19	1109	6-7-9-3-18-19
6	20	1209	6-7-9-3-16-20
6	21	1046	6-7-9-3-18-21
7	1	283	7-6-4-2-1
7	2	194	7-6-4-2
7	3	380	7-9-3
7	4	154	7-6-4
7	5	227	7-8-5
7	6	78	7-6
7	8	10	7-8
7	9	108	7-9
7	10	150	7-8-10
7	11	317	7-8-10-11
7	12	544	7-8-10-11-12
7	13	467	7-8-10-13
7	14	568	7-8-10-13-14
7	15	536	7-9-3-15
7	16	697	7-9-3-16
7	17	1056	7-9-3-15-17
7	18	718	7-9-3-18
7	19	1031	7-9-3-18-19
7	20	1131	7-9-3-16-20
7	21	968	7-9-3-18-21
8	1	293	8-7-6-4-2-1
8	2	204	8-7-6-4-2
8	3	390	8-7-9-3
8	4	164	8-7-6-4
8	5	217	8-5
8	6	88	8-7-6
8	7	10	8-7
8	9	118	8-7-9
8	10	140	8-10
8	11	307	8-10-11
8	12	534	8-10-11-12
8	13	457	8-10-13
8	14	558	8-10-13-14
8	15	546	8-7-9-3-15
8	16	693	8-10-11-12-16
8	17	1066	8-7-9-3-15-17
8	18	728	8-7-9-3-18
8	19	1041	8-7-9-3-18-19
8	20	1127	8-10-11-12-16-20
8	21	978	8-7-9-3-18-21
9	1	391	9-7-6-4-2-1
9	2	302	9-7-6-4-2
9	3	272	9-3
9	4	262	9-7-6-4

9	5	335	9-7-8-5
9	6	186	9-7-6
9	7	108	9-7
9	8	118	9-7-8
9	10	258	9-7-8-10
9	11	425	9-7-8-10-11
9	12	652	9-7-8-10-11-12
9	13	575	9-7-8-10-13
9	14	676	9-7-8-10-13-14
9	15	428	9-3-15
9	16	589	9-3-16
9	17	948	9-3-15-17
9	18	610	9-3-18
9	19	923	9-3-18-19
9	20	1023	9-3-16-20
9	21	860	9-3-18-21
10	1	433	10-8-7-6-4-2-1
10	2	344	10-8-7-6-4-2
10	3	530	10-8-7-9-3
10	4	304	10-8-7-6-4
10	5	357	10-8-5
10	6	228	10-8-7-6
10	7	150	10-8-7
10	8	140	10-8
10	9	258	10-8-7-9
10	11	167	10-11
10	12	394	10-11-12
10	13	317	10-13
10	14	418	10-13-14
10	15	686	10-8-7-9-3-15
10	16	553	10-11-12-16
10	17	1206	10-8-7-9-3-15-17
10	18	868	10-8-7-9-3-18
10	19	1104	10-11-12-16-20-19
10	20	987	10-11-12-16-20
10	21	1118	10-8-7-9-3-18-21
11	1	600	11-10-8-7-6-4-2-1
11	2	511	11-10-8-7-6-4-2
11	3	697	11-10-8-7-9-3
11	4	471	11-10-8-7-6-4
11	5	524	11-10-8-5
11	6	395	11-10-8-7-6
11	7	317	11-10-8-7
11	8	307	11-10-8
11	9	425	11-10-8-7-9
11	10	167	11-10
11	12	227	11-12
11	13	237	11-13
11	14	338	11-13-14
11	15	853	11-10-8-7-9-3-15
11	16	386	11-12-16
11	17	1373	11-10-8-7-9-3-15-17
11	18	1035	11-10-8-7-9-3-18
11	19	937	11-12-16-20-19
11	20	820	11-12-16-20
11	21	1111	11-12-16-20-21
12	1	827	12-11-10-8-7-6-4-2-1
12	2	738	12-11-10-8-7-6-4-2

12	3	476	12-16-3
12	4	698	12-11-10-8-7-6-4
12	5	751	12-11-10-8-5
12	6	622	12-11-10-8-7-6
12	7	544	12-11-10-8-7
12	8	534	12-11-10-8
12	9	652	12-11-10-8-7-9
12	10	394	12-11-10
12	11	227	12-11
12	13	77	12-13
12	14	143	12-14
12	15	632	12-16-3-15
12	16	159	12-16
12	17	1152	12-16-3-15-17
12	18	814	12-16-3-18
12	19	710	12-16-20-19
12	20	593	12-16-20
12	21	884	12-16-20-21
13	1	750	13-10-8-7-6-4-2-1
13	2	661	13-10-8-7-6-4-2
13	3	553	13-12-16-3
13	4	621	13-10-8-7-6-4
13	5	674	13-10-8-5
13	6	545	13-10-8-7-6
13	7	467	13-10-8-7
13	8	457	13-10-8
13	9	575	13-10-8-7-9
13	10	317	13-10
13	11	237	13-11
13	12	77	13-12
13	14	101	13-14
13	15	709	13-12-16-3-15
13	16	236	13-12-16
13	17	1229	13-12-16-3-17
13	18	891	13-12-16-3-18
13	19	787	13-12-16-20-19
13	20	670	13-12-16-20
13	21	961	13-12-16-20-21
14	1	851	14-13-10-8-7-6-4-2-1
14	2	762	14-13-10-8-7-6-4-2
14	3	691	14-12-16-3
14	4	722	14-13-10-8-7-6-4
14	5	775	14-13-10-8-5
14	6	646	14-13-10-8-7-6
14	7	568	14-13-10-8-7
14	8	558	14-13-10-8
14	9	676	14-13-10-8-7-9
14	10	418	14-13-10
14	11	338	14-13-11
14	12	143	14-12
14	13	101	14-13
14	15	775	14-12-16-3-15
14	16	302	14-12-16
14	17	1295	14-12-16-3-15-17
14	18	957	14-12-16-3-18
14	19	853	14-12-16-20-19
14	20	736	14-12-16-20
14	21	1027	14-12-16-20-21

15	1	692	15-3-1
15	2	730	15-3-9-7-6-4-2
15	3	156	15-3
15	4	690	15-3-9-7-6-4
15	5	763	15-3-9-7-8-5
15	6	614	15-3-9-7-6
15	7	536	15-3-9-7
15	8	546	15-3-9-7-8
15	9	428	15-3-9
15	10	686	15-3-9-7-8-10
15	11	853	15-3-9-7-8-10-11
15	12	632	15-3-16-12
15	13	709	15-3-16-12-13
15	14	775	15-3-16-12-14
15	16	473	15-3-16
15	17	520	15-17
15	18	494	15-3-18
15	19	807	15-3-18-19
15	20	907	15-3-16-20
15	21	744	15-3-18-21
16	1	853	16-3-1
16	2	891	16-3-9-7-6-4-2
16	3	317	16-3
16	4	851	16-3-9-7-6-4
16	5	910	16-12-11-10-8-5
16	6	775	16-3-9-7-6
16	7	697	16-3-9-7
16	8	693	16-12-11-10-8
16	9	589	16-3-9
16	10	553	16-12-11-10
16	11	386	16-12-11
16	12	159	16-12
16	13	236	16-12-13
16	14	302	16-12-14
16	15	473	16-3-15
16	17	993	16-3-15-17
16	18	655	16-3-18
16	19	551	16-20-19
16	20	434	16-20
16	21	725	16-20-21
17	1	1212	17-15-3-1
17	2	1250	17-15-3-9-7-6-4-2
17	3	676	17-15-3
17	4	1210	17-15-3-9-7-6-4
17	5	1283	17-15-3-9-7-8-5
17	6	1134	17-15-3-9-7-6
17	7	1056	17-15-3-9-7
17	8	1066	17-15-3-9-7-8
17	9	948	17-15-3-9
17	10	1206	17-15-3-9-7-8-10
17	11	1373	17-15-3-9-7-8-10-11
17	12	1152	17-15-3-16-12
17	13	1229	17-15-3-16-12-13
17	14	1295	17-15-3-16-12-14
17	15	520	17-15
17	16	993	17-15-3-16
17	18	542	17-18
17	19	855	17-18-19

17	20	972	17-18-19-20
17	21	792	17-18-21
18	1	874	18-3-1
18	2	912	18-3-9-7-6-4-2
18	3	338	18-3
18	4	872	18-3-9-7-6-4
18	5	945	18-3-9-7-8-5
18	6	796	18-3-9-7-6
18	7	718	18-3-9-7
18	8	728	18-3-9-7-8
18	9	610	18-3-9
18	10	868	18-3-9-7-8-10
18	11	1035	18-3-9-7-8-10-11
18	12	814	18-3-16-12
18	13	891	18-3-16-12-13
18	14	957	18-3-16-12-14
18	15	494	18-3-15
18	16	655	18-3-16
18	17	542	18-17
18	19	313	18-19
18	20	430	18-19-20
18	21	250	18-21
19	1	1187	19-18-3-1
19	2	1225	19-18-3-9-7-6-4-2
19	3	651	19-18-3
19	4	1185	19-18-3-9-7-6-4
19	5	1258	19-18-3-9-7-8-5
19	6	1109	19-18-3-9-7-6
19	7	1031	19-18-3-9-7
19	8	1041	19-18-3-9-7-8
19	9	923	19-18-3-9
19	10	1104	19-20-16-12-11-10
19	11	937	19-20-16-12-11
19	12	710	19-20-16-12
19	13	787	19-20-16-12-13
19	14	853	19-20-16-12-14
19	15	807	19-18-3-15
19	16	551	19-20-16
19	17	855	19-18-17
19	18	313	19-18
19	20	177	19-20
19	21	408	19-20-21
20	1	1287	20-16-3-1
20	2	1325	20-16-3-9-7-6-4-2
20	3	751	20-16-3
20	4	1285	20-16-3-9-7-6-4
20	5	1344	20-16-12-11-10-8-5
20	6	1209	20-16-3-9-7-6
20	7	1131	20-16-3-9-7
20	8	1127	20-16-12-11-10-8
20	9	1023	20-16-3-9
20	10	987	20-16-12-11-10
20	11	820	20-16-12-11
20	12	593	20-16-12
20	13	670	20-16-12-13
20	14	736	20-16-12-14
20	15	907	20-16-3-15
20	16	434	20-16

20	17	972	20-19-18-17
20	18	430	20-19-18
20	19	117	20-19
20	21	291	20-21
21	1	1124	21-18-3-1
21	2	1162	21-18-3-9-7-6-4-2
21	3	588	21-18-3
21	4	1122	21-18-3-9-7-6-4
21	5	1195	21-18-3-9-7-8-5
21	6	1046	21-18-3-9-7-6
21	7	968	21-18-3-9-7
21	8	978	21-18-3-9-7-8
21	9	860	21-18-3-9
21	10	1118	21-18-3-9-7-8-10
21	11	1111	21-20-16-12-11
21	12	884	21-20-16-12
21	13	961	21-20-16-12-13
21	14	1027	21-20-16-12-14
21	15	744	21-18-3-15
21	16	725	21-20-16
21	17	792	21-18-17
21	18	250	21-18
21	19	408	21-20-19
21	20	291	21-20

4. Conclusion

The Floyd-Warshall algorithm has been efficient in finding the shortest route and distances in Nigeria. It can be concluded that network analysis is a tool capable of designing the best possible ways of enhancing our daily life problems.

References

Sniedovich, M. (2005). *Dijkstra's Algorithms revisited: The OR/MS Connextion*, Department of Mathematics and Statistics, The University of Melbourne, Parkville VIC 3052, Australia.

Syslo, M. M., Deo, N., & Kowalk, J. S. (1983). *Optimization on Networks.* In Greenblah , A., editor , Discrete Optimization Algorithms with Pascal Programs, Pages 221-392. Prentice Hall, Inc., Eaglewood Cliffs, New Jersey.

Oyesiku, O. O. (2002). “From Womb to Tomb”, 24th Inaugural Lecture at Olabisi Onabanjo University on 27 August 2002. Ago-Iwoye: Olabisi Onabanjo University Press.

Rardin, R. L. (2003). *Optimization in Operations Research*, Pearson Education, Delhi, India.

Wei, D. (2010). “An optimized Floyd algorithm for the shortest path problem”. *Journal of Networks*, **5** (12), 1496-1504

Jahan, S. & Hasan, M. S. (2011), “A comparative study on algorithms for shortest-route problems and some extensions”, *International Journal of Basic and Applied Sciences IJBAS-IJENS*, **11**, 167-177.

Sangaiah, A. K., Han, M. and Zhang, S. (2014), “An investigation of Dijkstra and Floyd algorithms in National City traffic advisory procedures”, *International Journal of Computer Science and Mobile Computing*, **3** (2), 124-138.