Applied Mathematics and Computational Intelligence
Volume 12, No. 4, Nov 2023 [140-166]

Distance-Constrained Capacitated Vehicle Routing Problem for The Abuja Post Office Using Lexi-Search Algorithm

D. T. Achaku[1], B. Sani[2]

[1]Department of Mathematics, Federal University of Lafia.
[2]Department of Mathematics, Ahmadu Bello University, Zaria.

*Corresponding author: achakudt66@gmail.com

Received: 26 July 2023
Accepted: 14 Sept 2023

ABSTRACT

Transportation problems are generally modelled as vehicle routing problems (VRPs) since they consist of minimizing the overall cost while satisfying routing constraints and customers' orderings. In this paper, the mail delivery of the Nigerian Postal Services (NIPOST) is modeled as a distance- constrained capacitated variant (DCVRP) in order to address the problem of delay in mail delivery occurring frequently experienced in NIPOST. The Lexi-search algorithm was applied and implemented on the spreadsheet solver to solve the capacitated vehicle routing problem of Abuja Post Office. The algorithm was tested on Christofides Benchmark and found to perform well with computational analyses coded in c++ and statistical analyses of goodness fit performed using R Software. The problem has been formulated as a mixed integer programming problem based on a modification of the Tompkin's et al., model. Appropriate data was obtained from Abuja Post Office where the data is used to come up with the distance matrix. The annual cost savings for NIPOST using three vehicles is N1,950,054.66, N1,708,127.73 and N1,140,148.50 for N142, N163 and N213 per litre respectively. The annual cost savings for four vehicles, is N1,494,466.25, and therefore the purchse of a fourth vehicle will reduce delay in mail delivery though it might cost them more to aquire a new bus as already indicated. In the long run however, revenue will increase because of high patronage.

Keywords: Distance constrained and capacitated, mail delivery, mixed integer, Lexi-search algorithm and Vehicle Routing problem.

1 INTRODUCTION

The conveyance of goods to customers is considered to be one of the most challenging activities in logistic sectors. It has a key effect on the overall costs of industrial firms as well as on the environmental resources. Transportation and distribution problems are generally modelled as vehicle routing problems (VRPs) since they consist of minimizing the overall cost while satisfying routing constraints and customers' orderings. The VRP is concerned with finding a set of vehicle routes to serve known customers' demands from a single depot at minimum cost [1]. By involving additional requirements on routes construction, various VRP variants can be addressed. The capacitated VRP, (CVRP), firstly evoked by [2], has received much attention since it models a wide range of applications like fuel consumption optimization [3] and school bus routing problems. The CVRP consists of determining several constrained vehicle routes with minimum cost for serving a set of customers, whose geographical coordinates and demands are known in advance ([4], [5] and [6]). One variant of the CVRP of particular importance is the DCVRP, where both maximum weight and maximum distance constraints are imposed, that is, for each vehicle; a prefixed threshold of run distance is to be respected. Despite its great importance, the DCVRP has been evoked only in few studies such as ([7], [8] and [9]).

For solving CVRP variants, authors mainly focused on metaheuristics, which can find quite good solutions with a simultaneously running time. Among the most challenging metaheuristic methods for solving the CVRP variants, are genetic algorithms [10], greedy randomized adaptive search procedure [6], Tabu search [11], local search [12] ant colony optimization [13].

The Abuja Post Office uses three vehicles but contracts in two more vehicles for mail delivery. The three Vehicles cover a daily total distance of 537.6Km and a total Fuel cost of N11,000:00 excluding Sundays while the annual expenditure for the two contracted vehicles is N120,000. This translates to an annual cost on transportation of N3, 563,000:00. Reformations and restructuring were carried out at NIPOST over the years which were aimed at achieving 24 hour target for intra-city mail delivery in Abuja. The problem is that of the delay in mail delivery due to the fact that NIPOST is not applying the current trend of logistic delivery chain. In this paper, the capacitated vehicle routing problem with distance constraint is applied to the NIPOST case. This comprises of getting the minimum vehicle's travelled distance from the depot to other post offices around Abuja subject to system requirements. The vehicles have a maximum weight capacity and can travel up to an allowed distance, *L*. This is solved by modifying the [14] model, as a distance-constrained and capacitated model (DCVRP) for the Abuja Post Office and solved it using the Lexi-search algorithm.

The remainder of this paper is organized as follows. In the second section considers a review literature of related to this study while, in the third section, DCVRP was considered and its mathematical formulation. In the fourth section is the study the details of the Lexi-search algorithm. Application and analysis of the algorithm are shown in the fifth section and section six forms the conclusion. Figure 1 below shows the study site while the distance matrix between the Abuja Post Office and the other Post Offices can be found in appendix A.

Figure 1: The Study Site

2 LITERATURE REVIEW

A review of exact and approximate algorithms has been given by [15] (heuristic methods) for the VRP, focusing keenly on CVRP and DCVRP. Extensive surveys conducted on exact methods, such as the branch-and-bound, branch-and-cut and set-covering-based algorithms for these problems found relatively little literature on DCVRP. It was found that the lower bounds obtained by earlier relaxations for both the symmetric and asymmetric CVRP generally only allowed for the optimal solution of small instances. Several improved bounding techniques which were later proposed considerably increased the size of instances solvable by branch-and-bound. [16], developed additive bounding procedures for the asymmetric CVRP, while [17] and [18] developed bounding procedures based on Lagrangian relaxation for the symmetric CVRP.

A metaheuristic – Tabu search algorithm has been applied by [19] to solve the distance-constrained and capacitated VRP with split deliveries by order (DCVRPSDO). They showed that the customer demand, which can't be split in the classical VRP model, can only be discrete split deliveries by order. A model of double objective programming was constructed by taking the minimum number of vehicles used and minimum vehicle traveling cost as the first and the second objective, respectively. This approach contained a series of constraints, such as single depot, single vehicle type, distance-constrained and load capacity limit, split delivery by order, and so on.

A formulation based on partial paths has also been proposed in [20], for the VRP with time windows. Similar to the p-step formulation, the partial paths can start and end at the depot and must visit exactly a given number of customers. The authors obtain this formulation by applying Dantzig-Wolfe decomposition to a modified vehicle flow formulation of the problem, which relies on a modified graph to represent the solution as a giant tour. They proved that the linear relaxation of the resulting model provided a bound that is larger than or equal to the bound provided by the standard two-index flow formulation.

A formulation based on horizon decomposition for the capacitated VRP formulations with setup times was proposed by [16]. The paper partitioned the time horizon in several subsets, possibly with overlap, to have smaller subproblems, so that it can be quickly solved by a black-box optimization solver. In the column generation framework, columns become associated to production plans defined for only one of the partitions of the time horizon. These partial production plans are then combined in the master problems, as in the p-step formulation.

The Spreadsheet Solver developed and applied by [21], is capable of solving many variants of VRPs. Case studies of two real-world applications of the solver from the healthcare and tourism sectors that demonstrated its use were presented. It was also shown that the software is capable of solving instances up to 200. The computational results on benchmark instances from the literature were provided.

The South African Post Office problem was formulated as a mixed integer programming model to solve the capacitated *VRP* with time windows by [14]. They used Emalahleni mail centre with satellite locations around it as their pilot model and solved it using a heuristic technique. Distances from Emalahleni mail centre to each of the retail outlets were obtained using the latitude and longitude of each of the retail outlets and these locations fed into the heuristic. Route formulation was done and model validated.

For the Abuja Post Office problem, it will be formulated as a mixed integer programming model by modifying the [14] model.

3 MATHEMATICAL FORMULATION OF DCVRP

In this section, the problem formulation for the Abuja Post Office is considered by modifying the Tompkins's et al., (2017) model. The following notations will be defined now as

Denote: $i \neq j; i, j \in \{0,1,2,\dots,N\}$ where N is the total number of retail outlets
Denote: $k \in \{0,1,2,\dots,K\}$ where K is the total number of vehicles used by NIPOST, Abuja Post Office.
The decision variables include:

$$X_{ijk} = \begin{cases} 1, \text{if vehicle } k \text{ travels from node } i \text{ to node} j \\ 0, \quad \text{otherwise} \end{cases}$$

The model parameters include:

T_i : The arrival time at node i
t_{ij} : The travel time between node i to node j
m_i : Demand at node i
q_k : The carrying capacity of vehicle k
f_i : Service time at node i
e_i : The earliest arrival time at node i
l_i : The latest arrival time at node i
r_k : Maximum route time allowed for vehicle k
d_{ij} : The distance covered in moving from node i to node j for the arc
V The set of vertices: $V = \{0,\dots,N\}$ where 0 refers to the depot
E The set of edges: $E = \{(i, j) : i, j \in V\}$

The DCVRP is defined as an undirected graph $G = (V, E)$ where a node $j \in V$ corresponds to a customer and an edge $e \in E$ expresses a directed route between a pair of customers. Let n be the number of customers and a central depot denoted as 0. Each order is distinguished by a weight q_j and a volume u_j for $j \in \{1,\dots,n\}$ is to be delivered to its corresponding customer by a vehicle k. Every vehicle from the fleet is characterized by a maximum volume $m_{\max}$, a capacity weight in the range $[q_{\min}, q_{\max}]$ and can travel a distance in $[d_{\min}, d_{\max}]$. Given a depot and a set of geographically scattered customers, the objective of the DCVRP is to determine the lowest cost delivery route, under capacity and distance constraints, without flexible assignment. Cost considerations here include both travel costs as well as vehicle utilisation costs.
The problem definition of the variant is as described below.
(a) Each customer is visited exactly once by exactly one vehicle.
(b) Each vehicle route starts and ends at the depot.
(c) Each vehicle's route can only pass through one depot exactly once.
(d) A non-negative demand is associated with each customer and the sum of demands on any vehicle route may not exceed the vehicle capacity.
(e) The length of each vehicle route consists of loading time at the depot, inter-customer travel times and service time at each customer. The total length of any vehicle route may not exceed the specified vehicle range.
The formulation is as follows,

Minimise,

$$\sum_{i=0}^{N} \sum_{j=0}^{N} \sum_{j \neq i, k=1}^{K} d_{ij} X_{ijk} \tag{1}$$

Subject to,

$$\sum_{k=1}^{K}\sum_{j=1}^{N} X_{ijk} \leq K \qquad \forall i = 0 \qquad (2)$$

$$\sum_{j=1}^{N} X_{ijk} = 1 \qquad \forall i = 0 \ , k \in \{1, \dots, K\} \qquad (3)$$

$$\sum_{k=1}^{K}\sum_{j=0, j\neq i}^{N} X_{ijk} = 1 \qquad \forall i \in \{1, \dots, N\} \qquad (4)$$

$$\sum_{k=1}^{K}\sum_{i=0, i\neq j}^{N} X_{ijk} = 1 \qquad \forall j \in \{1, \dots, N\} \qquad (5)$$

$$\sum_{i=1}^{N} m_i \sum_{j=0, j\neq i}^{N} X_{ijk} \leq q_k \qquad \forall k \in \{1, \dots, K\} \qquad (6)$$

$$d_{iN}X_{Ni} + u_i \leq L \quad i = 1,2, \dots, N \qquad (7)$$

$$U_i - U_j + nX_{ij} \leq n - 1 \text{ for } \ i,j = 2, .., n \qquad (8)$$

$$1 \leq U_i \leq n - 1 \quad i = 2, .., n \qquad (9)$$

$$X_{ijk} \in \{0,1\} \qquad \forall i,j \in \{1, \dots, N\} \qquad (10)$$

The objective function (1) consists of minimizing the total routes distance, while constraint (2) is to ensure that at most k vehicles are used. Constraint (3) guarantees that each customer is served by exactly one vehicle. Constraints (4) and (5) are ensuring that each route must start and end at the depot. The maximum capacity of a vehicle should not exceed q_k and is stated in constraint (6). While constraint (7) ensures that the allowed distance range is not exceeded. The set of requirements (8) and (9) are the subtour elimination constraints. Lastly, constraint (10) ensures that solutions are integers.

Note that in the case of [14] model, the objective function minimises cost and will be replaced with minimising the total distance travelled. Other constraints are the same in our case except the time windows constraints which will not be used in this model.

The Capacitated Vehicle Routing Problem (CVRP) is a well-known combinatorial optimization problem in the field of operations research and transportation. Numerous studies have been conducted to address various aspects of the CVRP, aiming to develop efficient and effective solution approaches. Here is a summary of some of the related studies to CVRP and their advantages and disadvantages:

Table 1: Summary of some Related Studies, advantages and disadvantages

s/no	Details	Advantages	Weakness
1.	Dinh, T., Fukasawa, R. and Luedtke, J. (2017) Exact algorithms for the chance-constrained vehicle routing problem.	guarantee finding the optimal solution	computationally expensive, for large-scale instances
2	] G. Erdogan. An open source Spreadsheet Solver for Vehicle Routing Problems. Computers and	provide good-quality solutions in reasonable time	do not guarantee finding the optimal solution

	Operations Research, 2017		
3	Xia Y, Fu Z, Pan L, Duan F (2018) Tabu search algorithm for the distance-constrained vehicle routing problem with split deliveries	explore large search spaces efficiently and often find near-optimal solutions	can be more computationally demanding than heuristics
4	Sabar, N. R., Bhaskar, A., Chung, E., Turky, A., and Andy Song. (2019). A self-adaptive evolutionary algorithm for dynamic vehicle routing problems with traffic congestion.	combine the advantages of different algorithms to improve solution quality and computational efficiency.	the design and implementation of hybrid algorithms can be complex and may require careful parameter tuning

4 A LEXI-SEARCH ALGORITHM FOR THE PROBLEM

The Lexisearch algorithm is an algorithm that is used in solving exact problems for VRP's and was first proposed by [22]. From 1963 onwards this approach has been used to solve various combinatorial problems efficiently, and it performed more efficiently than the branch-and-bound algorithm. The Branch-and-Bound approach can be viewed as a particular case of lexicographic search, [23]. The set of all possible solutions to a combinatorial programming problem is arranged in hierarchy-like words in a dictionary, such that each incomplete word represents the block of words with this incomplete word as the leader. Each node is considered as a letter in an alphabet and each tour can be represented as a word with this alphabet. Thus the entire set of nodes in this tour (the set of solution) is portioned in to blocks. A block B with a leader $(\alpha_1, \alpha_2, .\alpha_3)$ of length three consists of all the words beginning with $(\alpha_1, \alpha_2, .\alpha_3)$ as the string of first three letters. The block A with leader (α_1, α_2) of length is the immediate super block of B and includes B as one of its sub blocks. The block C with leader$(\alpha_1, \alpha_2, .\alpha_3, \beta)$, implies that block B is the immediate super block of block C.

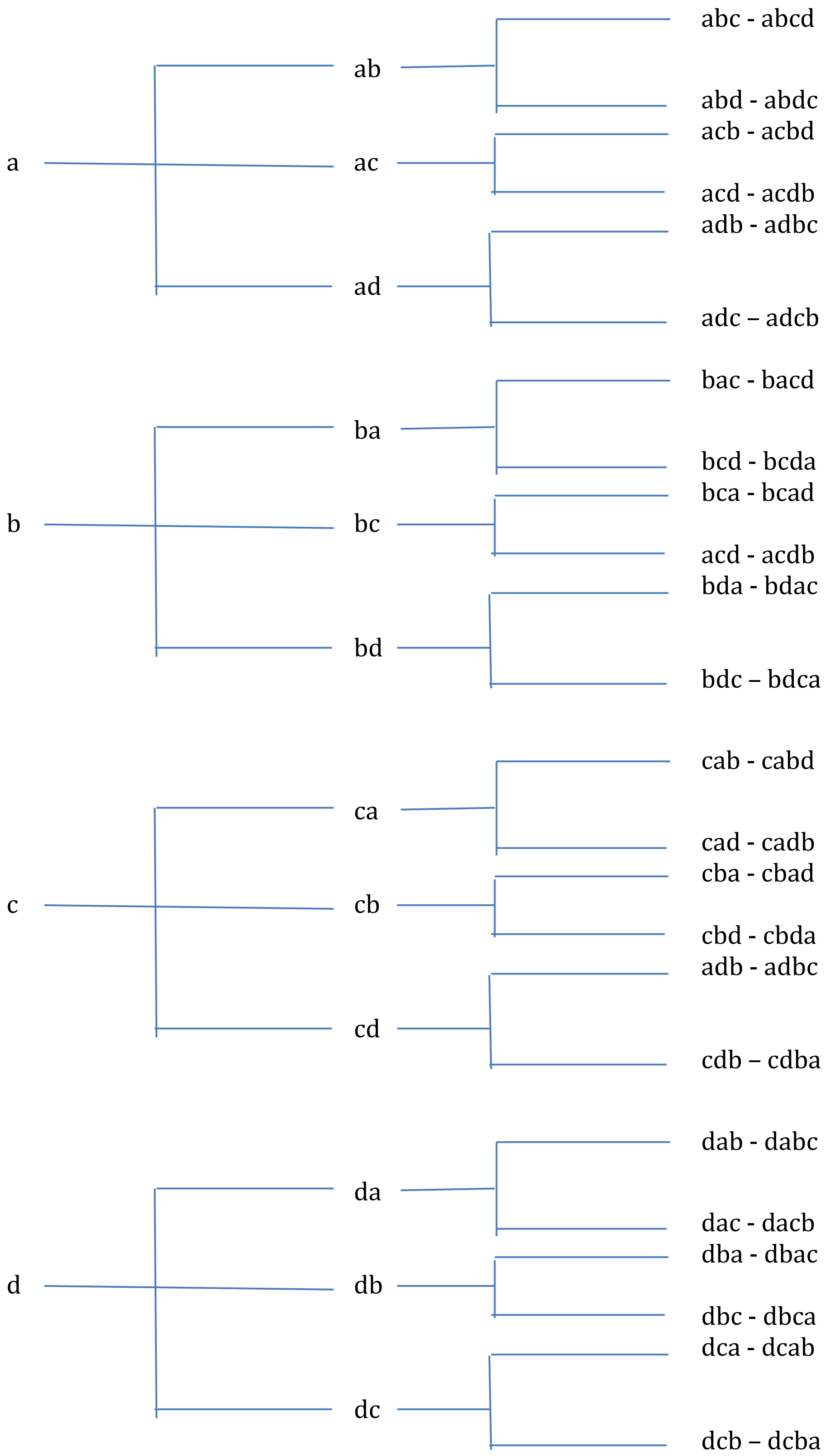

Figure 2: Arrangement of the travel routes

Let a, b, c, d, be the four cities to be travelled by a salesman. Then the set of possible partial and complete word is listed lexicographically as follows in figure 2. The word a with constitute a block with a as its block leader. In a block, there can many sub-blocks, for instance 'ab', 'bc', and 'bd' are the sub-blocks of b. These could be blocks with only one word for instance. The block with leader 'abd' has only one word '$abdc$'. All the incomplete words can be used as leaders to define blocks. For each of the blocks with leader 'ab' 'ac' and 'ac'; the block with leader a, is the immediate super-block.

4.1 Alphabet Table

When the distance matrix is augmented $D = d_{ij}$ which can be found in appendix A, the resultant effect is a matrix of order n by (n *+ m -1*) and is called the alphabet matrix, $A = [a(i, j)]$. This is a matrix whose *i*th row consists of rearranging the *i*th row of the distance matrix in an increasing order. Thus if $a(i, p)$ represents the *p*th element in the *i*th row of *A*, then $a(i,1)$ corresponds to the position of smallest element in *i*th row of the matrix D, [23]. Alphabet table " $[a(i, j) - d_{i,\alpha(i,j)}]$" is the combination of elements of matrix A and their values as shown in appendix B.

4.2 Lower bound

The Lexisearch algorithm [23], posit that the solution does not depend on lower bound, unlike branch-and-bound algorithm. The lower bound for each block leader on the objective function value is set to skip as many subproblems in the search procedure as possible. A subproblem is skipped if its lower bound exceeds the 'best solution value' found so far (i.e., upper bound) in the process. The higher the lower bound the larger the set of subproblems that are skipped. Following method is used for setting lower bound for each leader. Suppose the partial tour is $(1 = \alpha_1, \alpha_2, .\alpha_3)$ and 'city α_4' is selected for concatenation. Before concatenation, the bound is then checked for the leader $(\alpha_1, \alpha_2, .\alpha_3, \alpha_4)$. For that, computation is initiated from 2nd row of the 'alphabet table' and traverse up to the nth row and sum up the values of the first 'legitimate' city (the city which is not present in the tour), including 'city 1', in each row, excluding α_2-th and α_3-th rows. This sum is the lower bound for the leader $(\alpha_1, \alpha_2, .\alpha_3, \alpha_4)$.

4.3 The Lexisearch Algorithm

Step 0: Suppose, n be the number of cities including depot, m be the number of vehicles, distance travelled by the vehicles are Dist[i] for $1 \le i \le m$, 'city 1' be the depot, and $D_{\max}$ be the maximum distance allowed by each vehicle. Set 'best solution value (BS)' as large as possible, Dist[i]=0 for $1 \le i \le m$. Since 'city 1' is the starting city, computation is initiated from 1st row of the 'alphabet table'. Initialize 'partial tour value (PT)'=0, $k = 1$ and go to step 1.

Step 1: - Go to k^{th} element of the row (say, city p) with value as present city value (W). If (PT + W) ≥ (BS or $D_{\max}$), go to step 7, *else,* go to step 2.

Step 2: - If all the vertices are visited, add an edge connecting the 'city p' to 'city 1', compute the complete tour value (CT) and go to step 3, *else* go to step 4.

Step 3: - If $CT \ge BS$, go to step 7, *else,* BS = CT and go to step 7.

Step 4: - Calculate the lower bound (LB) for the present leader on the objective function value and go to *step 5*.

Step 5:- If (LB+PT+W) ≥ BS, drop the 'city p', increment k by 1, and go to step 6; *else,* accept the 'city p', compute PT = PT + W, update Dist[k] for the present vehicle k, and go to step 1.
Step 6: - For any vehicle k, if (Dist[k] < $D_{\max}$) go to step 1, *else,* go to step 7.
Step 7: - Jump this block, i.e., drop the present city, go back to the previous city in the tour (say, city q), i.e., go to the qth row of the 'alphabet table' and increment k by 1, where p was the index of the last 'checked' city in that row. If vertex q = 1 and k = n, go to step 8, *else,* go to step 1.
Step 8: Now BS is the optimal solution value and calculate the maximum distance travelled by any vehicle (Max), and then *stop*.

4.4 Illustration of the Algorithm

Working of the above algorithm is explained through a six-city and two-vehicle example with part of the distance matrix given in Appendix A. The augmented distance matrix is obtained by adding 1 copy of the depot (city 1) row and column (i.e., 1st row and 1st column) to the given matrix, which is reported in Appendix B. For the six city example the augmented matrix will rather be depicted instead of the alphabet matrix. Then two problems with this dataset will be considered, i.e., they will have two different values of the parameter $D_{\max}$.

The logic-flow of the algorithm at various stages is indicated in a search table that sequentially records the intermediate results, with decision taken (i.e., remarks) at these steps. The symbols used therein are listed below:
GS: Go to sub-block.
JB: Jump the current block.
JO: Jump out to the next, higher order block.

		A10	Nya	A1	Krv	Fsec	A11	Mam
		0	1	2	3	4	5	6
A10	0	0	17.2	1.9	19.4	3.7	3.9	8.51
Nya	1	17.2	0	12.2	2.3	15	12	12.5
A1	2	1.9	12.1	0	14.4	5.6	5.5	12.2
Krv	3	19.4	2.3	14.4	0	12.5	12.3	13
Fsec	4	3.7	15	5.6	12.5	0	3.7	5.3
A11	5	3.9	12	5.5	12.3	3.7	0	6
Mam	6	8.5	12.5	12.2	13	5.3	6	0

Problem 1: First, the value of $D_{\max}$ is set to very large number, say, 999.

Illustration of the example: Initialize BS = 999 and PT = 0, and Dist[0] = Dist[2] = 0. Starting from 1st row of the 'alphabet table'. Here, a(1,1) = 2 with $W = d_{0,2} = 1.9$, and (PT + W < BS) and (Dist[0] + W < $D_{\max}$). Next, the bound calculated for the present leader (0, 2). The bound will guide us whether the city 2 will be accepted or not.

$$LB = d_{1,3} + d_{2,7} + d_{3,1} + d_{4,5} + d_{5,4} + d_{6,4} + d_{7,4}$$
$$LB = 3.3 + 1.9 + 2.3 + 3.7 + 3.7 + 5.3 + 3.2 = 23.4$$

Now, since (LB + PT + W = 23.4 + 0 + 1.9 = 25.3 < BS), city 2 is accepted that leads to the partial tour {1→2} with PT = PT + W = 0 + 1.9 = 1.9, Dist [0] = 1.9. Note that in the search table, 0→2 (1.9) (0) +18, means W = 1.9, PT = 0, LB = 23.4 and remark is GS. Next step is to go to 2nd row of the 'alphabet table'. Since a (2, 1) = 3 with W = $d_{1,3}$ =2.3, and (PT + W < BS) and (Dist [0] + W < $D_{\max}$), and go for bound calculation for the present leader (0, 2, 1).

$$LB = d_{1,3} + d_{3,1} + d_{4,5} + d_{5,4} + d_{6,4} + d_{7,4}$$
$$LB = 3.3 + 1 + 2.3 + 3.7 + 3.7 + 5.3 + 3.2 = 21.5$$

Since, (LB + PT + W = 21.5 + 2 + 2.3 = 25.5 < BS), accept the city 3 that leads to the partial tour {0→2→1} with PT = PT + W = 1.9 + 2.3 = 4.2. Proceeding in this way the complete tour is obtained for the first vehicle as {0→2→1→4} with Dist [1] = 18.8, with a total demand of 1,130kgs and the tour for the second vehicle is {0→6→3→5→0} with Dist [2] = 40.9 and a carrying capacity of 1,140kgs.

Problem 2: The longest tour in the optimal solution of Problem 1 is 40.9. The new value of $D_{\max}$ is calculated as $D_{\max}$ = 0.97 x 40.9 = 39.673 with respect to the matrix. Next, choose $D_{\max}$ = 39.673 and run the same example. Then our algorithm obtains the Optimal tours for the vehicles as {0→2→4→5→0} and {0→6→3→0} with their distances 13.1 and 30.1 respectively.

5 ANALYSIS

In this section, focus is made on the structure of the worksheets and the menu of VRP Spreadsheet Solver. Focus will be on its usability rather than the technical details, for which the interested user can refer to the user's manual since solving this problem is NP- hard.

5.1 Locations

The details about the locations including their names, addresses, coordinates, demands are kept in the worksheet below in Table 2.

Table 2- List of Post Offices with Latitude and Longitude and their demands in kg.

Location ID	Name	Address	Latitude (y)	Longitude (x	Time window start	Time window end	Must be visited?	Service time	Demand
0	Depot	A10.FCT	9.0667000	7.4833000	00:00	23:59	Starting location	0:00	0
1	Customer 1	Nya,FCT	9.0561000	7.5789000	00:00	23:59	Must be visited	0:00	410
2	Customer 2	A1,FCT	9.0579000	7.4951000	00:00	23:59	Must be visited	0:00	360
3	Customer 3	Krv,FCT	9.0469000	7.7636000	00:00	23:59	Must be visited	0:00	430
4	Customer 4	Fed,FCT	9.0627000	7.4983000	00:00	23:59	Must be visited	0:00	360
5	Customer 5	A11,FCT	9.0241000	7.4783000	00:00	23:59	Must be visited	0:00	380
6	Customer 6	Mam,FCT	9.0899000	7.5197000	00:00	23:59	Must be visited	0:00	440
7	Customer 7	Nas,FCT	9.0671000	7.5098000	00:00	23:59	Must be visited	0:00	320
8	Customer 8	Old,FCT	9.0722600	7.4913000	00:00	23:59	Must be visited	0:00	460
9	Customer 9	Jik,FCT	9.1009000	7.2680000	00:00	23:59	Must be visited	0:00	480
10	Customer 10	Mog,FCT	9.0500000	7.5396000	00:00	23:59	Must be visited	0:00	400
11	Customer 11	Wz3,FCT	9.0596000	7.4719000	00:00	23:59	Must be visited	0:00	360
12	Customer 12	Bhr,FCT	9.2180000	7.4080000	00:00	23:59	Must be visited	0:00	320
13	Customer 13	Kub,FCT	9.2020000	7.3990000	00:00	23:59	Must be visited	0:00	380
14	Customer 14	Dei,FCT	9.1142000	7.2598000	00:00	23:59	Must be visited	0:00	460
15	Customer 15	Lsb,FCT	9.2760000	7.3593000	00:00	23:59	Must be visited	0:00	320
16	Customer 16	Gwa,FCT	8.9508000	7.0767000	00:00	23:59	Must be visited	0:00	460
17	Customer 17	Sul,NIG	9.1806000	7.1794000	00:00	23:59	Must be visited	0:00	360
18	Customer 18	Kuj,FCT	8.6590000	7.2705000	00:00	23:59	Must be visited	0:00	480
19	Customer 19	Kwl,FCT	8.7356000	6.9678000	00:00	23:59	Must be visited	0:00	360
20	Customer 20	Uab,FCT	8.8508000	7.0667000	00:00	23:59	Must be visited	0:00	420
21	Customer 21	Abj,FCT	8.8921000	6.8182000	00:00	23:59	Must be visited	0:00	400

5.2 Distance

This worksheet contains the distances and travel durations between every two points that are specified in the 4.2 Locations worksheet. A screenshot of part of the distance matrix is depicted in Table 3.

Table 3: Distance in Km

H309

	A	B	C	D	E	F
1	From	To	Distance	Duration	Method:	Bing Maps Distances (Km)
2	Depot	Depot	0.00	0:00		
3	Depot	Customer 1	17.20	0:14		
4	Depot	Customer 2	1.90	0:02		
5	Depot	Customer 3	19.40	0:17		
6	Depot	Customer 4	3.70	0:03		
7	Depot	Customer 5	3.90	0:03		
8	Depot	Customer 6	8.51	0:07		
9	Depot	Customer 7	5.90	0:05		
10	Depot	Customer 8	2.60	0:02		
11	Depot	Customer 9	22.70	0:19		
12	Depot	Customer 10	9.10	0:08		
13	Depot	Customer 11	5.00	0:04		
14	Depot	Customer 12	50.00	0:43		
15	Depot	Customer 13	29.90	0:26		
16	Depot	Customer 14	42.40	0:36		
17	Depot	Customer 15	47.80	0:41		
18	Depot	Customer 16	51.30	0:44		
19	Depot	Customer 17	65.00	0:56		
20	Depot	Customer 18	40.60	0:35		
21	Depot	Customer 19	74.30	1:12		
22	Depot	Customer 20	48.90	0:42		
23	Depot	Customer 21	114.50	1:38		
24	Customer 1	Depot	17.20	0:15		
25	Customer 1	Customer 1	0.00	0:00		

VRP Solver Console | 1.Locations | 2.Distances | 3.Vehicles | 4.Solution | 5.

5.3 Solution

This worksheet is generated to contain the list of stops for each vehicle specified in 3.Vehicles, and it uses the information in 4.2 Locations to regarding service times and pickup / delivery amounts, as well as the distance and duration in 4.3 Distances to compute the departure / arrival times the cost of traveling between customers. The worksheet computes the net profit rather than cost, to accommodate variants of the VRP that accumulate profits when customers are selectively visited. A screenshot of the 4.4 Solution worksheet is provided in Table 4 below depicting part of solution for three vehicles.

Table 4 – Solution worksheet

=IFERROR(IF(AND(ISNUMBER($P$12),$P$12<=21),INDEX('2.Distances'!$D$2:$D$486,$P$11*22+$P$12+ 1) +$X$11," ")," ")

rofit: -498.90

	B	F	G	I	J	K	L
	A 1	Stops:	4	Net profit:	-109.40		
	Location name	Distance	Delivered	Driving time	Arrival	Departure	Working time
0	Depot	0.00	0	0:00	08:00	08:00	0:00
1	Customer 8	2.60	460	0:02	08:02	08:02	0:02
2	Customer 11	9.20	820	0:08	08:08	08:08	0:08
3	Customer 20	60.50	1240	0:52	08:52	08:52	0:52
4	Depot	109.40	1240	1:34	09:34	09:34	1:34
5							
6							
7							
8							
9							

N	O	S	T	V	W	X	Y
Vehicle:	A 2	Stops:	7	Net profit:	-148.40		
Stop count	Location name	Distance	Delivered	Driving time	Arrival	Departure	Working tim
0	Depot	0.00	0	0:00	08:00	08:00	0:
1	Customer 2	1.90	360	0:02	08:02	08:02	0:
2	Customer 18	37.60	840	0:33	08:33	08:33	0:
3	Customer 19	65.50	1200	0:57	08:57	08:57	0:
4	Customer 17	71.80	1560	1:02	09:02	09:02	1:
5	Customer 14	94.40	2020	1:21	09:21	09:21	1:
6	Customer 13	108.50	2400	1:33	09:43	09:43	1:
7	Depot	148.40	2400	2:07	10:17	10:17	2:
8							
9							

5.4 Visualisation

The locations and the routes of the vehicles can be visually inspected by generating this optional worksheet. Options in the VRP Solver Console may be set to display various details about the locations including their delivery amounts or service times. This worksheet simply contains a scatter graph with the map of the region retrieved from the GIS web service. Figure 3 displays a screenshot of the worksheet for three vehicles as reflected in the different colours.

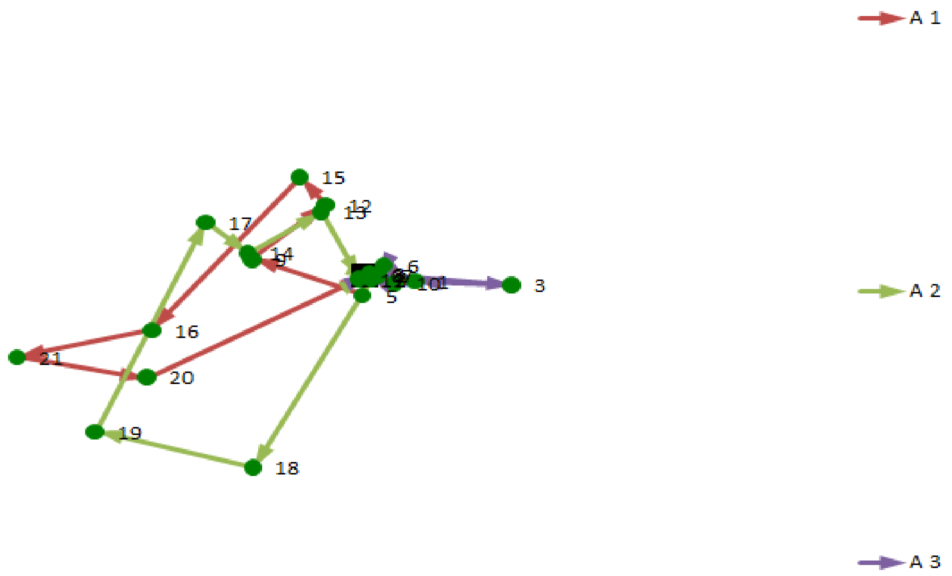

Figure 3: Showing the routes for the three vehicles

6 RESULTS AND DISCUSSION

In this study, the Lexi-search algorithm which Erdogan incorporated in an Excel spreadsheet solver version 3.0, is applied to solve the VRP since it is NP-hard to generate optimal results for more than or equal to three vehicles. The package was applied by Erdogan as reviewed in Erdogan, (2017). The Lexi-search algorithm as also applied by [23] and [24] in a TSP, is a special case of the branch-and-bound algorithm. The spreadsheet solver was applied to analyse the NIPOST delivery chain using the distance matrix in Appendix A and the demands for the Post Offices as depicted in Table 1.

The Abuja Post Office uses three homogeneous capacitated vehicles of 3000 Kilograms each and set the distance at [0,200], (that is no vehicle shall cover a distance of more than 200Km). Based on the demands from the Post Offices, the algorithm will first consider the demands then the distance restriction. Figure 2 below, shows the routes for the three vehicles. As a result of capacity constraint solution with one or two vehicles will not be feasible. The optimal routes for the three vehicles starting with the first is **A1 -** A10 => A11 => Jik => BHr => Lsb => Gwa => Abj => UAb => A10. This vehicle covers a distance of 196.9 Km with a total delivery of 2,780 Kg with the latest vehicle returning at 11:41 a.m. The route for this vehicle is shown in Figure 2 below.

Next, application of MatLab version 2016b is carried out to obtain the routes in respect of each vehicle separately. This is for the purpose of clarity in the route formation in moving from one Post Office to the other.

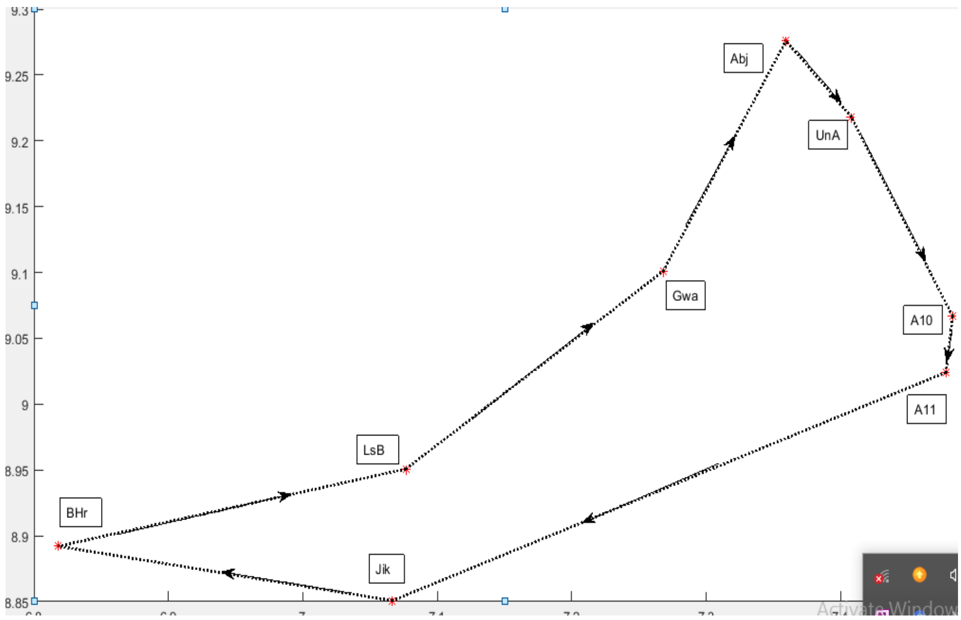

Figure 4: Showing the route for the first vehicle

Figure 5 below shows the optimal route of vehicle **A2** - A10 => A1 => Kuj => Kwl => Sul => Dei => Kub => Wz3 => A10. This vehicle covers a distance of 141.80 Km with a total delivery of 2,760 Kg.

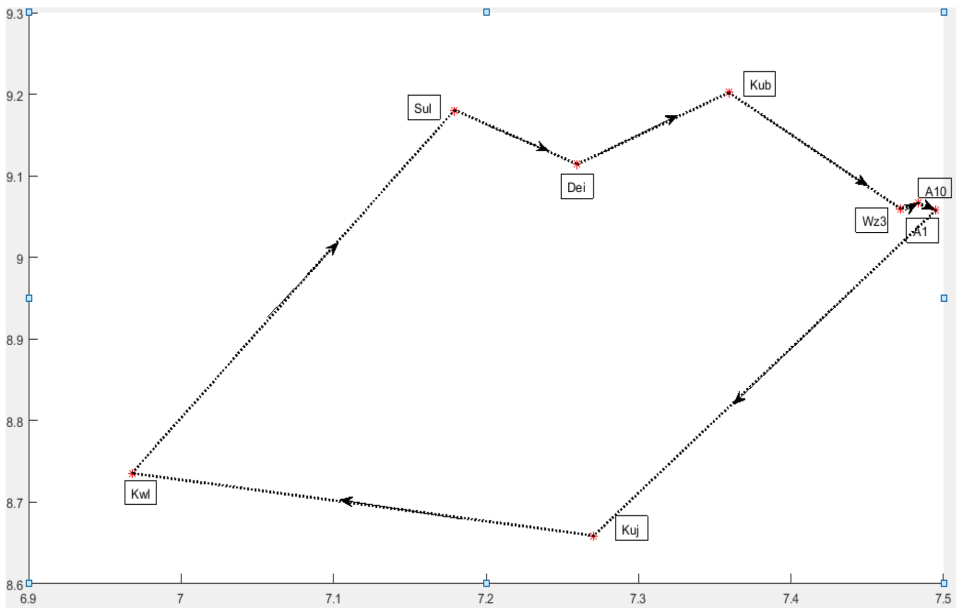

Figure 5: Showing the route for the second vehicle

Figure 6 below shows the optimal route of vehicle **A3** - A10 => Osec => Krv => Nya => Mog => Mam => Nass => Fsec => A10. This vehicle covers a distance of 44 Km with a total demand of 2,820 Kg.

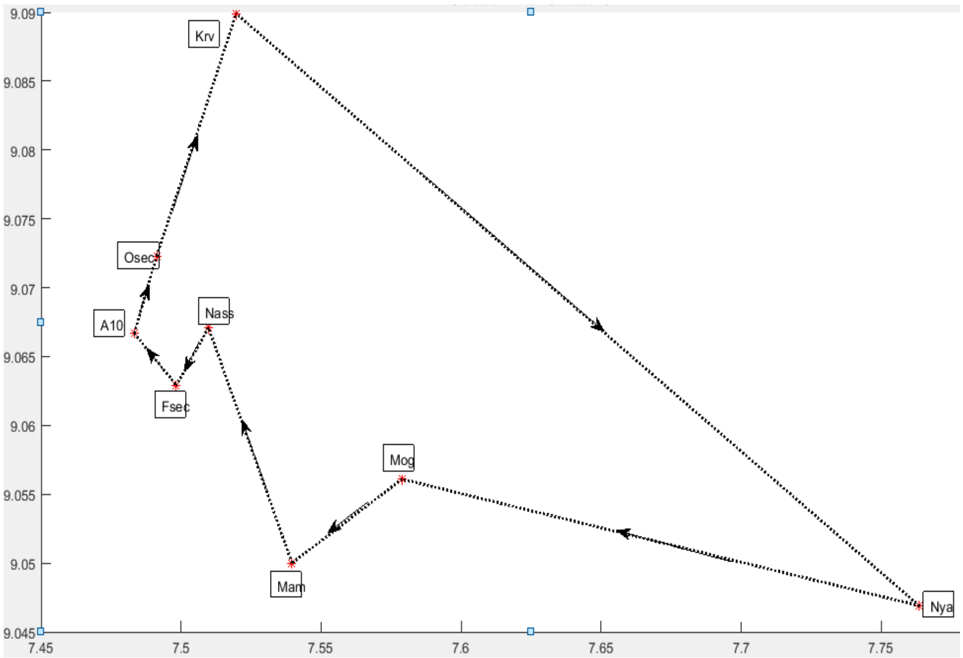

Figure 6: Showing the route for the third vehicle

Though, NIPOST is currently using three vehicles, however, the analysis was carried out for both three and four vehicles, If the analysis for the four vehicles is to be adopted, then NIPOST is supposed to obtain another vehicle. Figure 7 below shows the routes for the four vehicles.

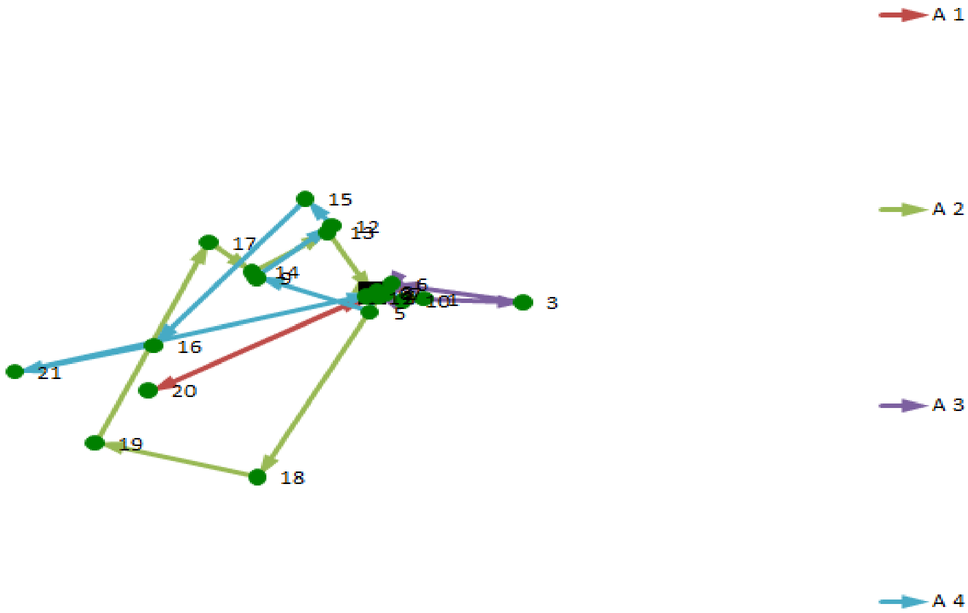

Figure 7: Showing the routes for using four vehicles

Next, MatLab version 2016b is applied again in order to obtain the routes in respect of each vehicle separately. This is for the purpose of clarity in the route formation in moving from one Post Office to the other for the case of four vehicles.

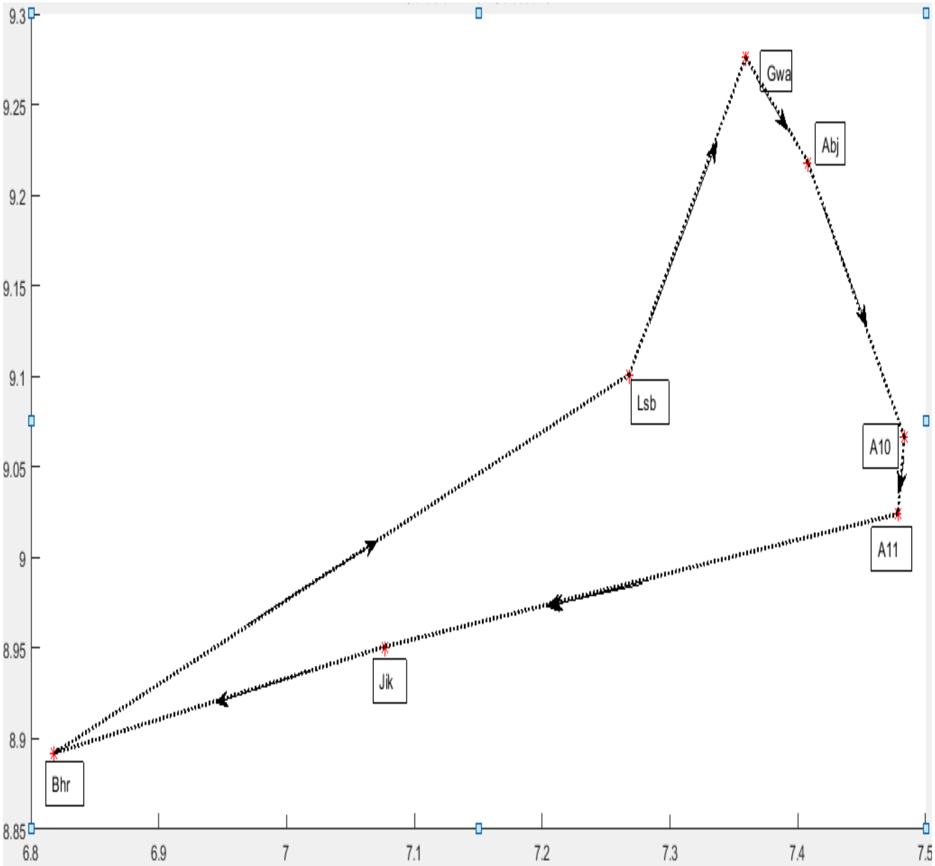

Figure 8: Showing the route for vehicle A1

Figure 8 shows the optimal route of vehicle **A1** - A10 => A11 => Jik => Bhr => LsBc => Gwa => Abj => A10. This vehicle covers a distance of 201.70 Km with a total delivery of 2,360 Kg, with a working time of 3:45 hours.

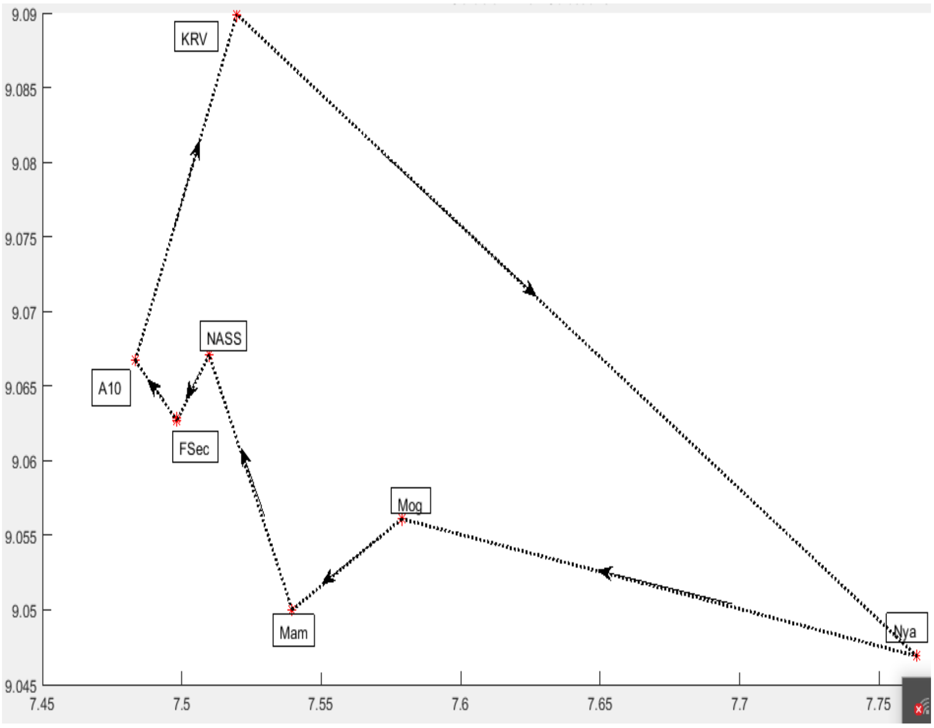

Figure 9: Showing the route for vehicle A2

Figure 9 above shows the optimal route of vehicle **A2** - A10 => Krv => Nya => Mog => Mam => Nass => Fsec => A10. This vehicle covers a distance of 39.40 Km with a total delivery of 2,360 Kg, with a working time of 34 minutes.

Figure 10 below shows the optimal route of vehicle **A3** - A10 => A1 => Kuj => Kwl => Sul => Dei => Kub => A10. This vehicle covers a distance of 148.40 Km with a total delivery of 2,400 Kilograms, with a working time of 2:07 hours.

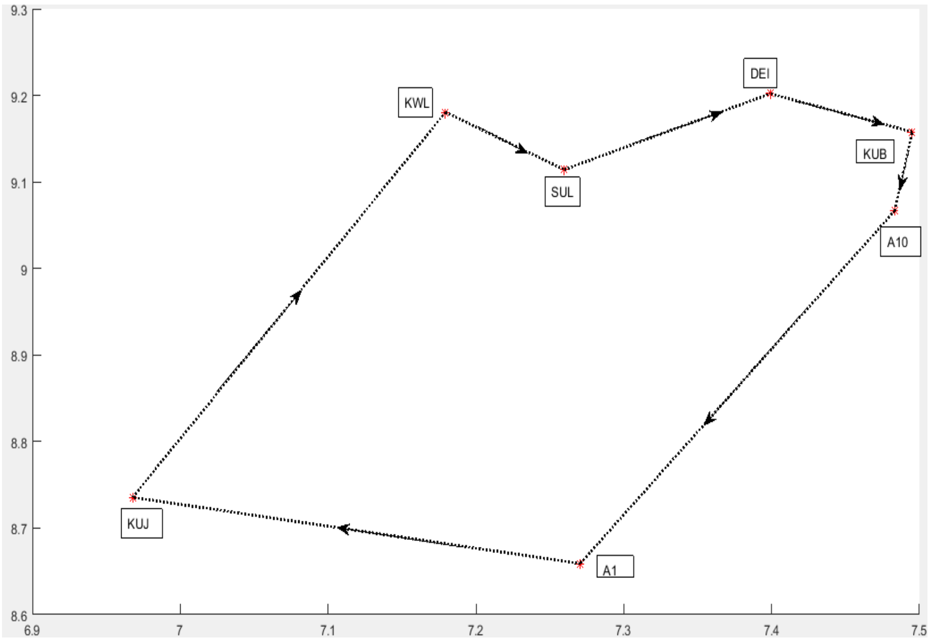

Figure 10: Showing the route for vehicle A3

The Table 5 below, gives the analysis for both the three and four vehicles.

Table 5: Showing number of vehicles with daily and annual fuel consumption

No. of Vehicles			1	2	3	4
Distance covered			-	-	382.7	489.9
Consumption rate (L/100km)			-	-	9.5	9.5
Average Petrol Price (N/litre)	142	Daily Cost (N)	-	-	5,153.18	6,608.75
		Annual Cost (N)	-	-	1,612,945.34	2,068,533.75
	163	Daily Cost (N)	-	-	5,926.11	7,586.10
		Annual Cost (N)	-	-	1,854,872.27	2,374,449.77
	213	Daily	-	-	7,743.93	9,913.13

		Cost (N)				
		Annual Cost (N)	-	-	2,423,851.50	3,102,808.59

The three vehicles cover a total distance of 382.7 Km. The product of the total distance covered, an average fuel consumption of 9.5 litres per 100km and average cost of fuel per litre at N142, N163 and N213 respectively give us the daily and annual cost as shown in Table 3 above. With this, the annual cost savings for three vehicles is also equals to N3,563,000 – N1,612,945.34 = N1,950,054.66, N3,563,000 – N1,854,872.27 = N1,708,127.73 and N3,563,000 – N2,423,851.50 = N1,140,148.50 respectively.

Table 6: Savings analyses for three Vehicles

No. of Vehicles	Average Petrol Price	Annual savings
3	142	N3, 563,000 – N1,612,945.34 = N1,950.054.66
3	163	N3, 563,000 – N1,854,872.27 = N1, 708,127.73
3	213	N3, 563,000 – N2,423,851.50 = N1,140,148.50

Similarly, if NIPOST is to use a fourth vehicle then the annual cost savings for four vehicles is =N3,563,000 – N2,068,533.75 = N1,516,466.25, N3, 563,000 – N2,374,449.77 = N1,188,550.23 and N3, 563,000 – N3,102,808.59 = N460,191.41 based on N142/litre, N163/litre and N213/litre respectively from Table 6. If Abuja Post Office is to use a fourth vehicle, the one advantage for NIPOST in mail delivery will be that, the vehicles return to the depot earlier than when three vehicles are used. However, this has an additional cost-: the cost of purchasing a new pick-up vehicle. Though in the long run, patronage will increase because of timely delivery of mails and consequently an increase in revenue.

Table 7: Savings analyses for four Vehicles

No. of Vehicles	Average Petrol Price	Annual savings
4	142	N3, 563,000 – N2,046,533.75 = N1,516,466.25
4	163	N3, 563,000 – N2,374,449.77 = N1,188,550.23
4	213	N3, 563,000 – N3,102,808.59 = N460,191.41

6.1 Case Of Heterogenous Capacity

The analysis above is that of Homogenous capacitated vehicles used by NIPOST therefore, it was decided to vary the capacities of these vehicles and observe what happens to the routes used by the vehicles. The variation was done in the first, second and third instances and results obtained using Lexisearch algorithm is shown in Table 8 below.

Table 8: Heterogenous Capacity, Demands and Working Time

Variations	Vehicle	Demand (kg)	Capacity	Distance (km)	Latest Working Time
First	A1	3,220	3,300	144.50	2:03pm
	A2	2,780	3,200	196.9	3:41pm
	A3	2,360	2,500	39.40	0:34pm
Second	A1	3,440	4,000	134.50	1:58pm
	A2	2,920	3,000	196.9	3:41pm
	A3	2,000	2,000	51.20	0:44p
Third	A1	3,220	3,500	150.50	2:03
	A2	2,460	2,800	159.9	2:55
	A3	2,680	2,700	76.4	1:46

It can be observed that after varying the capacities of the vehicles, the latest delivery time is 3:41 hours and the total distances covered by the three vehicles are 380.80 km, 382,60 km and 386.50 km for the first, second and third variations respectively. In all the three heterogenous cases, the total distance covered is less than the distance covered in the case of homogeneous capacity because, some have a carrying capacity of more than three tons.

6.2 Computational Results

In order to give evidence of goodness of work performed, firstly, the Excel spreadsheet software was applied and used to produce solutions about the case of the Abuja Post Office. After that, its' performance was run against some of Christofides et al instance files; then the results are compared against the best know solution available in literature for these benchmarks.

Based on the reason stated above, the algorithm was coded and tested with C^{++} using a laptop computer with an Intel i7 CPU running at 2.5 GHz with 8 GB of RAM, a configuration that would reflect the computers used in practice. The CPU time limit is set for the solution algorithm VRP Spreadsheet Solver to 15 min for instances, and increased it linearly with the number of customers for larger instances. It is not claimed that a single algorithm can solve all the variants of the VRP to near-optimality, and believe that a computational experiment to solve all existing variants is beyond the scope of this paper. Hence, it was opted to use the well-known and widely used benchmark data set by Christofides that contains two of the main variants of the VRP, the Capacitated VRP and the Distance Constrained VRP.

The computational results are provided in Table and show that the algorithm performs very well for up to 100 customers, and returns acceptable results for larger instances. The performance of the algorithm is better for instances with a distance constraint, due to the reduced search space. The performance is slightly degraded for the last four instances, which consist of artificially constructed clusters of customers.

Table 9: Computational results on benchmark instances.

Instance	**Number of**	**Fleet size**	**Vehicle**	**Distance**	**Best known**	**VRP Spreadsheet Solver**			
name	**customers**	Empty Cell	**capacity**	**limit**	**solution value**	**Average**	**Average gap**	**Best**	**Best gap**
vrpnc1	50	5	160	N/A	524.61	**524.61**	**0.00%**	**524.61**	**0.00%**
vrpnc2	75	10	140	N/A	835.26	840.67	0.65%	**835.26**	**0.00%**
vrpnc3	100	8	200	N/A	826.14	841.05	1.80%	831.28	0.62%
Vrpnc4	199	17	200	N/A	1291.29	1341.19	3.86%	1323.08	2.46%
Vrpnc5	50	6	160	200	555.43	556.77	0.24%	**555.43**	**0.00%**
Vrpnc6	75	11	140	160	909.68	913.13	0.38%	**909.68**	**0.00%**
Vrpnc7	100	9	200	230	865.94	876.40	1.21%	**865.94**	**0.00%**
Vrpnc8	120	7	200	N/A	1042.11	1047.82	0.55%	1047.61	0.53%
Vrpnc9	100	10	200	N/A	819.56	821.29	0.21%	821.29	0.21%
vrpnc10	120	11	200	720	1541.14	1565.01	1.55%	1554.51	0.87%
vrpnc11	100	11	200	1040	866.37	886.41	2.31%	869.96	0.41%

6.3 Statistical Analysis

The best known solution and the best solution were analysed on R version 4.3 to obtain the goodness of fit.

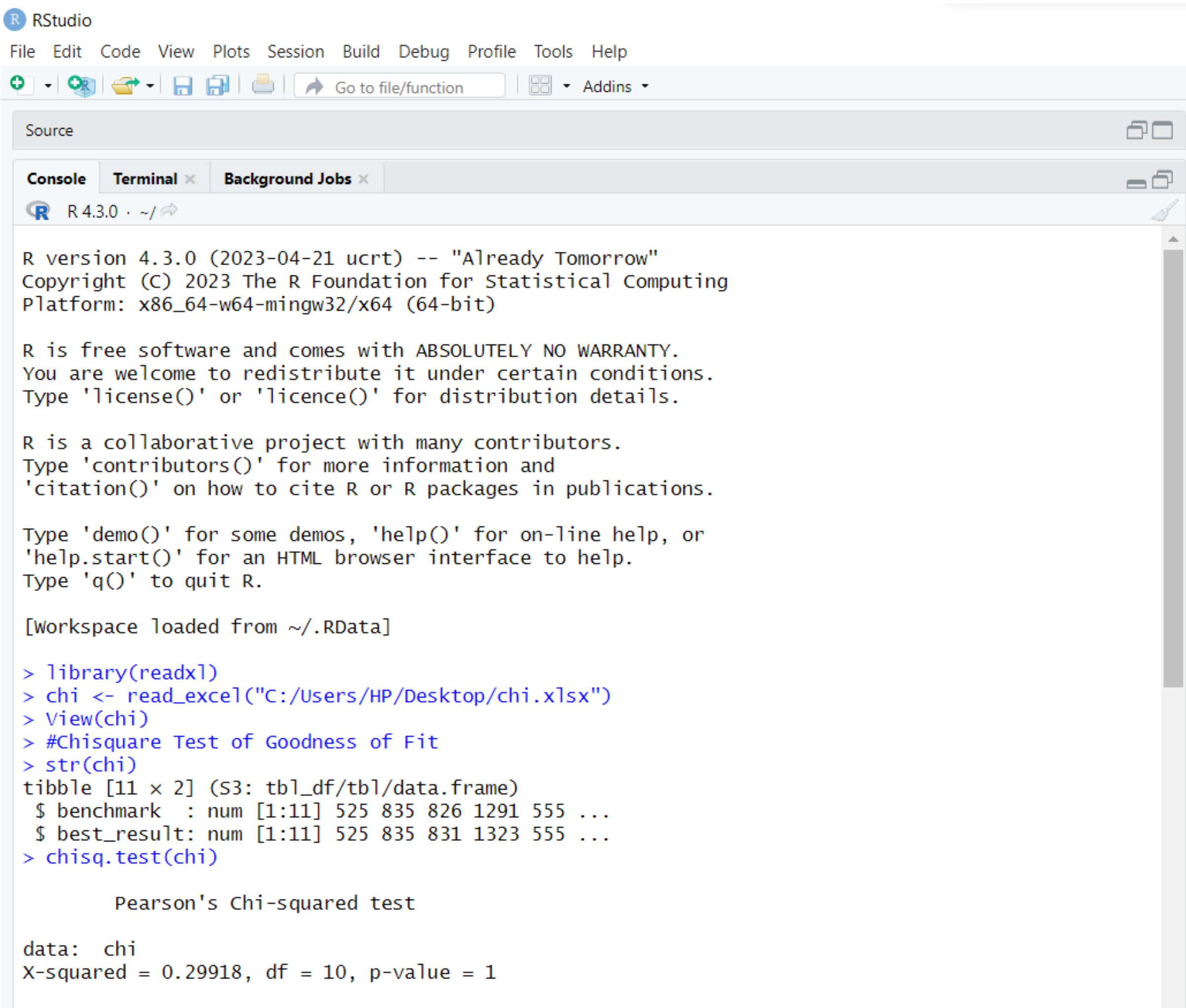

Figure 11: R script for chi-square of goodness fit

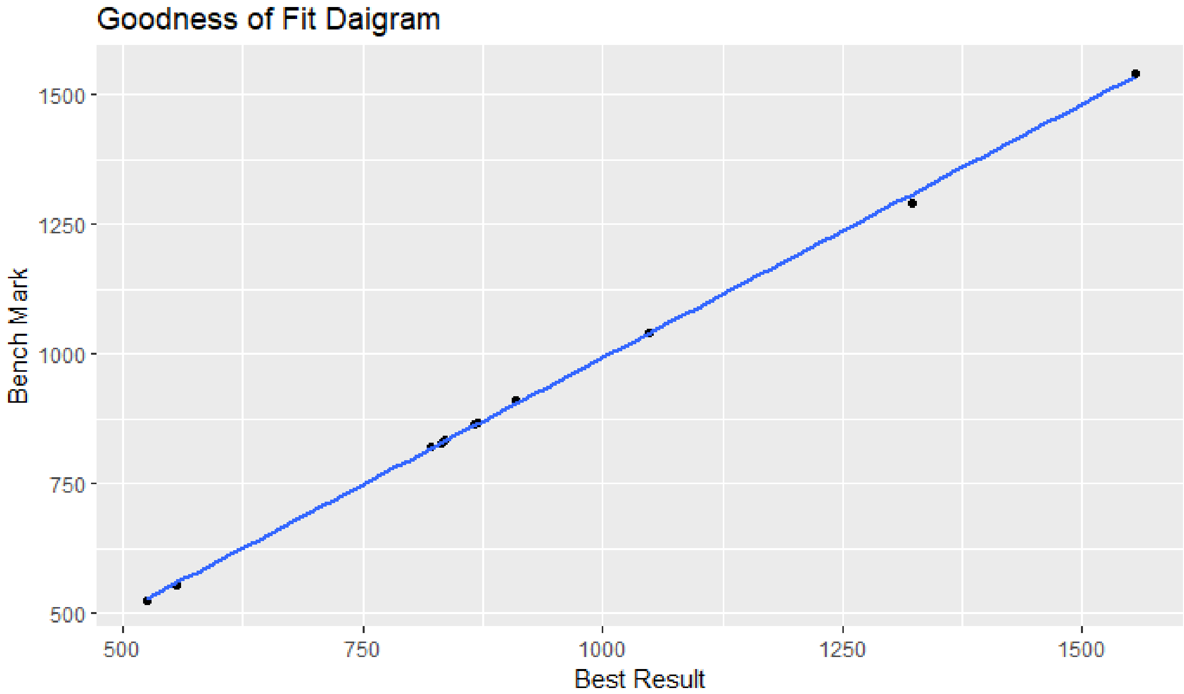

Figure 12: Scatter diagram

Chi square Output:

Table 10 : Goodness of Fit Statistic Values:

X-Squared	Degree of Freedom	P Value
0.29918	10	1

With a chi-square value of 0.29918, degrees of freedom of (df) of 10, and a probability value of 1, it suggest that the observed data is exactly what would be expected under the null hypothesis of no association between the variables. It infers that the chi-square test indicates on significant difference between the benchmark results and the best results obtained in using the coded algorithm on c++

7 CONCLUSION

In this study, the Lexi-search algorithm was applied which was implemented on the spreadsheet solver to solve the capacitated vehicle routing problem of Abuja Post Office. The problem has been formulated as a mixed integer programming problem based on a modification of the [14] model. Formulated on the fact that the vehicles have homogeneous capacity, two vehicles cannot exhaust the total deliveries at once. Therefore, three routes are developed with a total distance of 382 Km. Four routes were developed for four vehicles in case NIPOST may want to acquire a fourth vehicle though, it will cost them more. It is observed that with the current demand, it is not feasible to use two vehicles without split delivery. The annual cost savings for

NIPOST using three vehicles is N1,950,054.66, N1,708,127.73 and N1,140,148.50 for N142, N163 and N213 per litre respectively. For using four vehicles, the annual cost savings is N1,494,466.25. A fourth vehicle will reduce delay in mail delivery though it might cost them more to aquire a new bus as already indicated. In the long run however, revenue will increase because of high patronage.

The savings for three vehicles in the heterogeneous case has been calculated as N1,682,439.66 at a petrol cost of N163 per litre, which was found to be less than the corresponding savings at the same petrol cost for the case of homogeneous capacity case.

REFERENCES

[1] Y. Ren, M. Dessouky and F. Ordonez. "The multi-shift vehicle routing problem with overtime", *Computers and Operations Research*, vol. 37, pp 1987-1998, 2010.

[2] G. Dantzig and J. Ramser. "The travelling salesman problem with capacity constraint", *Management Science*, vol. 8, pp 80-91, 1959.

[3] Y. Xiao, Q. Zhao, I. Kaku and X. Yuchun. "Development of a fuel consumption optimization model for the capacitated vehicle routing problem", *Computers and Operations Research*, vol. 39, pp. 1419-1431, 2017.

[4] A. L. Chen, G. K. Yang and Z. M. Wu. "Hybrid discrete particle swarm optimization algorithm for capacitated vehicle routing problem," *Zhejiang University SCIENCE A*, vol. 7, no. 4, pp 607-614, 2010.

[5] W. Szeto, Y. Wu and S. Ho. "An artificial bee colony algorithm for the capacitated vehicle routing problem", *European Journal of Operational Research,* vol. 215, pp 126 -135, 2011.

[6] Y. Marinakis. "Particle Swarm Optimization for the Vehicle Routing Problem with Stochastic Demands", *Applied Soft Computing*, vol. 13, pp 1693–1704, 2011.

[7] G. Nagy and S. Salhi. "Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries", European *Journal of Operational Research,* vol. 162, pp 126-14, 2005.

[8] B. Cheang, X. Gao, A. L. Null, H. Qin and W. Xhu. "Multiple pickup and delivery traveling salesman problem with last-in-first-out loading and distance constraints", *European Journal of Operational Research*, vol. 223, no. 3 pp 60-75, 2012.

[9] S. Almoustafa, S. Hana and N. Mladenovic. "New exact method for large asymmetric distance-constrained vehicle routing problem", *European Journal of Operational Research*, vol. 226, no. 5, pp 386 –394, 2013.

[10] H. Nazif and L. S. Lee. "Optimised crossover genetic algorithm for capacitated vehicle routing problem", *Applied Mathematical Modelling*, vol. 36, pp 2110–2117, 2012.

[11] J. Jin, T. G. Crainic and A. Lokketangen. "A parallel multi-neighbourhood cooperative Tabu search for capacitated vehicle routing problems", *European Journal of Operational Research*, vol. 222, pp 441–451, 2012.

[12] L. Ke and Z. Feng. "A two-phase metaheuristic for the cumulative capacitated vehicle routing problem", *Computers and Operations Research*, vol. 40, pp 633-638, 2013.

[13] C. J. Tinga and C. H. Chen. "A multiple ant colony optimization algorithm for the capacitated location routing problem", *International Journal of Production Economics*, vol. 141, 34-44, 2013.

[14] A, Tompkins, R. Batista and E. Vigo. "The South African Post Office supply chain Design Route Optimization", BPJ 420 Final Report, 2017.

[15] G. Laporte. "The vehicle routing problem: an overview of exact and approximate algorithms," *European Journal of Operational Research*, vol. 59, pp 345-359, 1992.

[16] M. Drexl. "Rich vehicle routing in theory and practice", *Logistics Research*, vol. 5, pp. 47-63, 2012.

[17] J. Liu, X. Ren and H. Ma. "A new PSO algorithm with random switching," *Applied Mathematics and Computation*, vol. 218, pp 9579- 9593, 2012.

[18] M. Gang, Z. Wei and C. Xiaolin. "A novel particle swarm optimization algorithm based on particle swarm migration," *Applied Mathematics and Computation*, vol. 218, pp 6620 – 6626, 2012.

[19] Y. Xia, Z. Fu, L. Pan and F. Duan. "Tabu search algorithm for the distance-constrained vehicle routing problem with split deliveries by order", *PLoS ONE*, vol. 13, no. 5, e0195457, 2018.

[20] G. H. Alvina, R. L Kek and Q. M. Cheu. "Distance-c0nstrained capacitated vehicle routing problems with flexible assignment of start and end depots", *Mathematical and Computer Modelling,* vol. 47, no. 6 pp 140–152, 2008.

[21] G. Erdogan. "An open source Spreadsheet Solver for Vehicle Routing Problems", *Computers and Operations Research*, vol. 84, pp 62-72, 2017.

[22] S. N. Pandit. "Some quantitative combinatorial search problems", *Indian Institute of Technology*, Kharagpur, India, 1963.

[23] Z. H. Ahmed. "A new reformulation and an exact algorithm for the quadratic assignment problem," *Indian Journal of Science and Technology*, vol. 6, no 4, pp 4368–4377, 2016.

[24] G.V. Lakshmi, "Lexis-search Approach to Travelling Salesman", *Problem Journal of Mathematics* (IOSR-JM) e-ISSN: 2278-5728,p-ISSN: 2319-765X, vol. 6, no 4, pp 01-08, 2013.

Appendix A: 22 × 22 Distance Matrix in Km

		A10	Nya	A1	Krv	Fsec	A11	Mam	Nas	Osec	Jik	Mog	Wz3	Bhr	Kub	Dei	Bls	Gwa	Sul	Kuj	Kwl	UniA	Abj
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
A10	0	0	17.2	1.9	19.4	3.7	3.9	8.51	5.9	2.6	22.7	9.1	5	50.2	29.9	42	47.8	51.3	65	40.6	74.3	48.9	114.5
Nya	1	17.2	0	12.2	2.3	15	12	12.5	16.2	16.3	6.7	4.5	12.4	52.6	34.2	46	52.2	68	68.8	56.1	84.2	65.6	126.7
A1	2	1.9	12.1	0	14.4	5.6	5.5	12.2	7.6	2	16.5	11.1	6	57.6	37.2	41	35.2	51.3	65.9	35.7	73.5	48.9	114.2
Krv	3	19.4	2.3	14.4	0	12.5	12.3	13	13.1	16.4	4.1	6.8	14.7	51.6	39.2	51	57.2	66.2	73	53.8	86.5	63.9	127.4
Fsec	4	3.7	15	5.6	12.5	0	3.7	5.3	2.2	6.3	20.6	9.7	4.3	50.3	29.9	37	47.9	54.9	59.8	40.8	74.6	52.5	116.9
A11	5	3.9	12	5.5	12.3	3.7	0	6	4.8	7.5	11.8	7.7	7	53.7	33.3	37	51.3	57.3	59.7	43	77.2	55.3	117.5
Mam	6	8.5	12.5	12.2	13	5.3	6	0	4.6	12.9	17.1	7.7	9.5	49.2	28.8	33	46.8	61.4	55.2	45.7	82.9	59	122.6
Nas	7	5.9	16.2	7.6	13.1	2.2	4.8	4.6	0	9.3	20.3	8.1	6.1	52.2	30.9	34	49.9	57.1	56.4	42.7	76.8	54.7	118.4
Osec	8	2.6	16.3	2	16.4	6.3	7.5	12.9	9.3	0	18.5	11.8	6.6	53.4	32.9	36	50.9	68.6	58.8	38	72.3	66.2	127.7
Jik	9	22.7	6.4	16.5	4.1	20.6	11.8	17.1	20.3	18.5	0	15.2	18.5	61.3	40.9	44	58.9	72.7	67.3	62.3	90.6	70.3	137.5
Mog	10	9.1	4.5	11.1	6.8	9.7	7.7	7.7	8.1	11.8	15.2	0	11.1	51.7	31.3	35	49.3	61.5	57.7	47.8	123	59.1	122.7
Wz3	11	5	12.4	6	14.7	4.3	7	9.5	6.1	6.6	18.5	11.1	0	48.7	28.3	32	46.3	53.2	54.7	39.4	64.7	51.3	122.3
Bhr	12	50.2	52.6	57.6	51.6	50.3	53.7	49.2	52.3	53.4	61.3	51.7	48.7	0	20.9	43	2.4	64.8	67.5	74.2	70.6	65.1	183.7
Kub	13	29.9	34.2	37.2	39.2	29.9	33.3	28.8	30.9	32.4	40.9	31.3	28.3	20.9	0	14	22.4	44.4	47.1	53.8	53.9	44.7	163.3
Dei	14	42.4	46.2	41.3	50.7	37.2	37.1	32.6	33.8	36.2	44.7	35.1	32.1	42.8	14.1	0	26.2	48.2	50.9	57.6	57.7	48.5	167.1
Bls	15	47.9	53.2	55.2	56.2	47.9	51.3	46.8	49.9	50.9	58.9	49.3	46.3	2.4	32.1	26	0	61.6	65.1	71.8	71.9	62.7	181.3
Gwa	16	51.3	68	51.3	66.2	54.9	57.7	61.4	57.1	68.6	72.7	61.5	53.7	64.8	37.8	24	61.6	0	40.2	22.7	16.7	2.4	63.2
Sul	17	65	68.8	63.9	73.3	59.8	59.7	55.2	56.4	58.8	67.3	57.7	54.7	67.5	36.7	23	48.8	40.2	0	51.3	63	38.8	189.7
Kuj	18	40.6	56.1	35.2	52.8	40.8	43	45.7	42.7	38	62.3	47.8	39.4	74.2	56.7	43	54.3	23.7	51.3	0	27.9	22.5	158.7
Kwl	19	74.3	84.2	73.5	86.5	74.6	77.2	82.9	76.8	72.3	90.6	123	64.7	70.6	53.9	58	71.9	16.9	63	27.9	0	17.3	137.1
UniA	20	48.9	65.6	48.9	63.9	52.5	55.3	59	54.7	66.2	70.3	59.1	51.3	61.8	35.4	21	61.7	2.4	37.8	20.3	14.3	0	60.8
Abj	21	115	127	114	128	117	118	123	118	128	138	123	122	184	163	167	181	63.2	190	159	137	60.8	0

Appendix B: R scripts For The Analysis

```
#Checking the Accuracy of the Model
accuracy(modeler225)



# mean of the Forecast error
mean(er)
library(ggplot2)

#Saving all The Plot in R
ggsave("timeplot1.png", width = 18, height = 10, units = c("cm"), dpi = 300)

ggsave("timeplot2.png", width = 18, height = 10, units = c("cm"), dpi = 300)

#Chisquare Test of Goodness of Fit

#Importing the Data into R with an Object Name "chi"
library(readxl)
chi <- read_excel("C:/Users/HP/Desktop/chi.xlsx")
View(chi)
#Checking The Structure of the data
str(chi)
#Converting the data into a Matrix Form
chi1= as.matrix(chi)
chi1
#Calling The Chisquare Function
chisq.test(chi1)
#Plotting the Output Showing the Goodness of Fit
library(ggplot2)
#calling the plot Function
ggplot(data = chi, aes(x =best_result , y = benchmark)) +
  geom_point()+
  geom_smooth(method = "lm", se = F) +
  labs(x = "Best Result",
       y = "Bench Mark", title = "Goodness of Fit Daigram")
dev.off()
#End of The Script.
```