APPLICATION OF LEXISEARCH ALGORITHM TO VEHICLE ROUTING PROBLEM WITH TIME WINDOWS

Achaku D. T.[1*] and Sani B.[2]

[1]Department of Mathematics, Federal University of Lafia, Nassarawa.

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

[*]Corresponding Author's Email: achakudt66@gmail.com

Cite this article:

Achaku D. T., Sani B. (2024), Application of Lexisearch Algorithm to Vehicle Routing Problem with Time Windows. African Journal of Mathematics and Statistics Studies 7(2), 96-111. DOI: 10.52589/AJMSS-IBKASWQH

Manuscript History

Received: 11 Jan 2024

Accepted: 15 Mar 2024

Published: 23 Apr 2024

ABSTRACT: *Vehicle Routing Problem with Time Windows (VRPTW) is an NP hard combinatorial scheduling optimization problem in which a minimum number of routes have to be determined so as to serve each of the destinations within their specified time windows. In this paper, the mail delivery of the Nigerian Postal Services (NIPOST) is modelled as a VRPTW in order to address the problem of delay in mail delivery occurring regularly in NIPOST. The Abuja Post Office is used as a case study and the Model of a related literature is applied with modifications to solve the problem. The problem is solved by applying Lexisearch algorithm using data that was obtained from Abuja Post Office and computational results on Solomon's 100 instances were used to validate the algorithm.*

KEYWORDS: Vehicle routing problem, Branch and bound algorithm, Time windows, Optimal and mail clearance.

INTRODUCTION

The Vehicle Routing Problem (VRP) is a well-known NP-hard (Non-deterministic polynomial-time hard) combinatorial optimization problem. As explained by Zhou et al. (2017), this problem arises in distribution systems that usually involve scheduling in a constrained environment. Nowadays, most companies are using trucks as their inland distribution mode because of its flexibility and fuel efficiency, especially because in this era of increasing fuel price, the distribution function must be managed efficiently. Inefficient vehicle route means more energy consumption and higher distribution cost. Consequently, the problem of finding an efficient vehicle route for distributing mails from depot to customers, is very critical. In VRP, it is desired to determine minimum distance routes of the vehicles which are such that each location is visited once and only once by one of the vehicles under the restriction that all routes start and ultimately end at the depot (Wang et al., 2018). In this study, we shall consider the VRPTW.

The problem we deal with in this paper is a specific VRP which is commonly found in distribution practice and it is the Vehicle Routing Problem with Time Windows (VRPTW). Customers receive delivery service from a vehicle only at a certain time interval known as the customer time window. If a vehicle arrives too early at a customer destination, it must wait until the time window opens and it is not allowed to arrive late (He et al., 2014). This additional requirement makes the problem more difficult to solve. In terms of computational complexity, this problem is well known to be an NP-hard problem. As a consequence, exact solution methods can only find optimal solutions in a reasonable amount of time even for small size problems. Depending on each variant of the VRP, there can be different approaches to finding a solution, so the objectives and restrictions to be found in practice are very broad, offering the opportunity to apply different models and algorithms in the search for a solution that guarantees to secure the lowest cost possible based on the efficient use of the resources available in the distribution area. There are currently numerous research studies being carried out by different authors, in which they propose different solution algorithms of this problem (Farid, 2019).

In the case of NIPOST, the problem of delay in mail delivery is mainly due to the fact that NIPOST is not engaging modern logistic delivery chains and techniques. This is even as there have been reformations and reorganisation of the sector over the years. Nigerians using the services of NIPOST especially those receiving and sending mails raise various complaints. The aim of this study is to route NIPOST fleet of vehicles which have limited capacity that will service the Post Offices in the study according to time windows for each Post Office. This will be achieved by modifying the Tompkins et al. (2017) model, to formulate and solve the VRPTW for the Abuja Post Office and the other 22 Post Offices depending on it for mail delivery.

The Abuja Post Office and 22 Post Offices under it use three vehicles but contract 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 of transportation of N3, 563,000:00. Reformations and restructuring were carried out at NIPOST over the years which were aimed at achieving a 24 hour target for intra-city mail delivery in Abuja. The problem is that of the delay in mail delivery mainly due to the fact that NIPOST is not applying the current trend of the logistic delivery chain.

Article DOI: 10.52589/AJMSS-IBKASWQH
DOI URL: https://doi.org/10.52589/AJMSS-IBKASWQH

The remainder of this paper is organised as follows. In Section 2, the review of related literature is taken. Section 3 looks at the problem formulation. In Section 4, the Lexisearch algorithm is discussed. In Section 5, discussion of the results is considered. In Section 6, the conclusion of the paper is presented.

LITERATURE REVIEW

Moradi (2020) presented a new multi-objective discrete learnable evolution model (MODLEM) to address the vehicle routing problem with time windows (VRPTW). Learnable evolution model (LEM) includes a machine learning algorithm, like the decision trees, that can discover the correct directions of the evolution leading to significant improvements in the fitness of the individuals. The proposed MODLEM was tested on the problem instances of Solomon, (1987) VRPTW benchmark. The performance of this proposed MODLEM for the VRPTW was assessed against the state-of-the-art approaches in terms of both the quality of solutions and the computational time. Experimental results and comparisons indicated the effectiveness and efficiency of the proposed intelligent routing approach.

Differently, using the fuzzy approach for VRPTW, Li et al. (2017) solved the VRP with fuzzy demands using a hybrid differential evolution algorithm. In the study, a fuzzy chance-constrained program model was designed and a hybrid intelligent algorithm was proposed to solve the model. In this regard, Haitao et al. (2018) used a similar approach and designed a fuzzy chance-constrained program model based on fuzzy credibility theory and applied it for the open VRP. Earlier, Tanga et al. (2009) proposed and solved a VRP with fuzzy time windows (VRPFTW). Service level issues associated with violation of time windows in a vehicle routing problem were described using fuzzy membership functions, and the concept of fuzzy time windows was proposed. In their paper, they stated that the proposed fuzzy time windows can reflect the customers' satisfaction for service and for different kinds of customers. Another efficient use of the fuzzy approach in routing networks is related to Gedik et al. (2017) who developed a hybrid evolutionary fuzzy learning algorithm that automatically determines the near optimal travelling path in large-scale travelling salesman problems.

The problem of *VRP* with soft time windows arose in the dispatching of technicians to customer sites to repair broken equipment. Baldaci et al. (2012) proposed an optimization model under service-time uncertainty to address this problem. The authors argued that service-time uncertainty exists significantly due to the high potential for misdiagnosis of the failure at the time when the client requests a repair. The authors then solved the problem using the method of branch-and-price by applying their approach on real data sets from an industry.

Walid (2020) solved vehicle routing problems under the Time Window constraint using Microsoft Excel and used equations and algorithms to optimise the transportation route. The primary data for the study was gathered from an interview with Turkish Transportation Company in order to optimise the routes of several vehicles.

Ahmed (2013) considered the vehicle routing problem with simultaneous pick-up and delivery with time windows. Simultaneous pick-up and delivery (also called reverse logistics) allows one to address the problem of removing goods after they have been labelled as obsolete. To solve this complex combinatorial optimisation problem, the author used the Excel solver which aims to find a set of paths that minimises the total distance of the paths while serving

Article DOI: 10.52589/AJMSS-IBKASWQH
DOI URL: https://doi.org/10.52589/AJMSS-IBKASWQH

simultaneously, customers' delivery and pick-up demands. Further, time window constraints were also considered in the paper, which made the problem harder to solve. However, including time windows makes the problem more realistic, where results show that the Lexisearch algorithm can find solutions that are quite competitive with respect to previously reported algorithms in the literature.

Tompkins et al. (2017) presented a heuristic technique to solve the South African Post Office problem. Formulated as a mixed integer programming model to solve the capacitated *VRP* with time windows, they used the Emalahleni mail centre with satellite locations around it as a pilot study. Using the latitude and longitude of each of the retail outlets, the distances from the mail centre to each of the retail outlets was calculated and fed into the heuristic. Then the optimal routes were formulated and the model validated.

In this paper, we formulate our VRPTW as a mixed integer programming model by modifying the Tompkins et al. (2017) model. The Tompkins et al. (2017) model is suitable since the Abuja Post Office also has satellite Post Offices that depend on it just like the South African Post Office. The aim here is to model the Abuja Post Office as a VRP with time windows as was done for the South African Post Office.

METHODOLOGY

We define the following notations for the purpose of our model

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 the South African Post Office. The decision variables include:

$$X_{ijk} = \{1, if\ vehicle\ k\ travels\ from\ node\ i\ to\ nodej\ 0, \quad otherwise$$

The model parameters include:

T_i : The arrival time at node i

C_{ij} : Total cost incurred from node i to node j for the arc (i, j)

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

Article DOI: 10.52589/AJMSS-IBKASWQH
DOI URL: https://doi.org/10.52589/AJMSS-IBKASWQH

Based on modifications of the Tompkins et al., (2017) model, our model will be defined as follows:

Parameter definitions will be the same as for Tompkins's, et al. (2017) model wherever required except for,

d_{ij} : The distance covered in moving from node i to node j for the arc (i, j)

u_i : The set of retail Post Offices on that route

We define the VRPTW on a directed graph $G = (N, A)$ where node set N consists of a subset $C = \{1,2,\ldots,n\}$ of customers and node 0 corresponds to the depot. Each customer is associated with a demand d_i and a time interval, $[a_i, b_i]$ which is expected to be serviced within this time called a time window. Each arc $(i, j) \in A$ is associated with a travel distance d_{ij} and a travel time t_{ij} that usually includes a service time for node i (it is assumed that the triangular inequality is satisfied for both). A fleet of K vehicles each with a capacity Q is due to service customers through a path starting from depot 0 and ending at depot 0. The VRPTW can be described as the following model.

Minimise,

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

Subject to,

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

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

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

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

$$\sum_{i=1}^{N} \sum_{j=0,j\neq i}^{N} X_{ijk}(t_{ij} + f_i) \leq r_k \qquad \forall k \in \{1, \ldots, K\} \tag{6}$$

Article DOI: 10.52589/AJMSS-IBKASWQH
DOI URL: https://doi.org/10.52589/AJMSS-IBKASWQH

$$\sum_{k=1}^{K} \sum_{i=0, j\neq i}^{N} X_{ijk}(t_{ij} + f_i) \leq T_i \qquad \forall i \in \{1, \dots, K\} \qquad (7)$$

$$e_i \leq (t_{ij} + f_i) \leq l_i \qquad \forall i \in \{1, \dots, N\} \qquad (8)$$

$$U_i - U_j + nX_{ij} \leq n - 1 \; for \;\; i, j = 2, .., N \qquad (9)$$

$$1 \leq U_i \leq n - 1 \quad i = 2, .., N \qquad (10)$$

$$x_{ijk} \in \{0,1\} \qquad \forall i, j \in \{1, \dots, N\} \qquad (11)$$

Expression (1) is the objective function which minimises the total distance travelled in delivery of mails to all the Post Offices and back. Constraint (2) ensures that the total number of vehicles does not exceed K. Constraint (3) means that only one vehicle connects any two Post Offices. Constraint (4) states that there is exactly one vehicle which enters each Post Office i. While, constraint (5) states that there is exactly one vehicle leaving each Post Office is in Post Office j once and only once. Constraint (6) is used to ensure that the travel time for each vehicle does not exceed the maximum time allocated to a specific route. Constraint (7) ensures that the arrival time of each vehicle at a Post Office is less than or equal to the arrival at that specific Post Office. Constraint (8) guarantees that the arrival time and the service time for each vehicle at node i is greater or equal to the earliest arrival time. Constraints (9) and (10) are the subtour elimination constraints which guarantee that each optimal route should be feasible and (11) states that the assignment constraints should be integers.

A Lexisearch Algorithm for the Problem

In Lexisearch algorithm, the set of all possible solutions to a problem is arranged in a hierarchy, such that each incomplete word represents the block of words with this incomplete word as the leader. For the VRP, each node is considered as a letter in an alphabet and the tour set can be represented as a word with this alphabet. Thus the entire set of words in this dictionary (namely, the set of solutions) is partitioned into blocks. Bounds are computed for the values of the objective function over the blocks of words, which are then compared with the 'best solution value' found so far. If no word in the block can be better than the 'best solution value' found so far, jump over the block to the next one. If the current block, which is to be jumped over, is the last block of the present super-block, then jump out to the next super-block. Further, if the value of the current leader is already greater than or equal to the 'best solution value'; no need for checking the subsequent blocks within this super-block. However, if the bound indicates a possibility of better solutions in the block, enter into the sub block by concatenating the present leader with appropriate letter and set a bound for the new sub-block so obtained by Ahmed (2016).

Alphabet Table

When we augment the distance matrix $D = d_{ij}$ 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 $a(i,1)$

Article DOI: 10.52589/AJMSS-IBKASWQH
DOI URL: https://doi.org/10.52589/AJMSS-IBKASWQH

corresponds to the position of the smallest element in *i*th row of the matrix D, [23]. Alphabet table "$\lfloor a(i,j) - d_{i,\alpha(i,j)} \rfloor$" is the combination of elements of matrix A and their values as shown in appendix B.

The Lexisearch algorithm which Erdogan incorporated in an Excel spreadsheet solver and left it as an open source solver package was applied by Erdogan as reviewed in related literature. Though the spreadsheet solver comes with a manual, in trying to apply it to solve the Abuja Post Office problem, I had to interact with him personally by Electronic Mail (email) in his University of Bath, United Kingdom. The Lexisearch algorithm as applied by Erdogan (2017) is a special case of the branch-and-bound algorithm. In this section, we apply the spreadsheet solver to analyse the NIPOST delivery chain using the distance matrix in Appendix A.

Incomplete Word and Block

An incomplete route $(\alpha_1, \alpha_2, ..., \alpha_k)$ of *k* cities, where $k \le n$, is called an incomplete word or a leader of length *k* and is also called a leader of length *k*. A block *Q* 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 the first three letters. The block *P* with leader (α_1, α_2) of length 2 is the immediate super-block of *Q* and includes *Q* as one of its sub-blocks. The block *R* with leader $(\alpha_1, \alpha_2, .\alpha_3, \alpha_4)$ is a sub-block of *Q*. The block *Q* consists of many sub-blocks (α1, α2, α3, αr) one for each αr. The block *Q* is the immediate super-block of block *R* (Ahmed, 2016).

Lower Bound

The Lexisearch algorithm Ahmed, (2016), posits 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 bounds 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, we check bound for the leader $(\alpha_1, \alpha_2, .\alpha_3, \alpha_4)$. For that, we start our computation 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 $(\alpha_1, \alpha_2, .\alpha_3, \alpha_4)$.

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, we start our computation from 1st row of the 'alphabet table'. Initialize 'partial tour value (PT)'=0, $k = 1$ and go to step 1.

Article DOI: 10.52589/AJMSS-IBKASWQH
DOI URL: https://doi.org/10.52589/AJMSS-IBKASWQH

Step 1: Go to k[th] element of the row (say, city p) with value as present city value (W). If (PT + W) $\geq$ (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 $CT \geq BS$, 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) $\geq$ 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*.

Table 1: List of Post Offices Location, Time Windows and Service Times

Location ID	Name	Address	Latitude (y)	Longitude (x	Time window start	Time window end	Must be visited?	Service time
0	Depot	A10.FCT	9.0667000	7.4833000	00:00	08:00	Starting location	0:00
1	Customer 1	Nya,FCT	9.0561000	7.5789000	08:00	10:00	Must be visited	0:10
2	Customer 2	A1,FCT	9.0579000	7.4951000	09:30	11:00	Must be visited	0:15
3	Customer 3	Krv,FCT	9.0469000	7.7636000	08:00	10:00	Must be visited	0:15
4	Customer 4	Fed,FCT	9.0627000	7.4983000	08:03	10:20	Must be visited	0:12
5	Customer 5	A11,FCT	9.0241000	7.4783000	08:18	10:30	Must be visited	0:10
6	Customer 6	Mam,FCT	9.0899000	7.5197000	08:33	11:00	Must be visited	0:15
7	Customer 7	Nas,FCT	9.0671000	7.5098000	09:11	11:00	Must be visited	0:20
8	Customer 8	Old,FCT	9.0722600	7.4913000	09:30	11:00	Must be visited	0:10
9	Customer 9	Jik,FCT	9.1009000	7.2680000	09:30	11:00	Must be visited	0:12
10	Customer 10	Mog,FCT	9.0500000	7.5396000	08:48	10:00	Must be visited	0:15
11	Customer 11	Wz3,FCT	9.0596000	7.4719000	09:30	10:00	Must be visited	0:20
12	Customer 12	Bhr,FCT	9.2180000	7.4080000	08:10	09:03	Must be visited	0:10
13	Customer 13	Kub,FCT	9.2020000	7.3990000	09:20	11:00	Must be visited	0:15
14	Customer 14	Dei,FCT	9.1142000	7.2598000	09:30	11:00	Must be visited	0:09
15	Customer 15	Lsb,FCT	9.2760000	7.3593000	08:41	10:00	Must be visited	0:10
16	Customer 16	Gwa,FCT	8.9508000	7.0767000	09:47	10:40	Must be visited	0:15
17	Customer 17	Sul,NIG	9.1806000	7.1794000	09:00	10:40	Must be visited	0:10
18	Customer 18	Kuj,FCT	8.6590000	7.2705000	09:27	11:00	Must be visited	0:15
19	Customer 19	Kwl,FCT	8.7356000	6.9678000	08:30	10:30	Must be visited	0:13
20	Customer 20	Uab,FCT	8.8508000	7.0667000	09:50	11:00	Must be visited	0:10
21	Customer 21	Abj,FCT	8.8921000	6.8182000	09:44	11:53	Must be visited	0:09

Article DOI: 10.52589/AJMSS-IBKASWQH
DOI URL: https://doi.org/10.52589/AJMSS-IBKASWQH

Table 2: Optimal Routes for One and Two Vehicles with Time Windows (TW)

	1 Vehicle	2 Vehicles			
		A1		A2	
Route	TW	Route	TW	Route	TW
0	0:00 – 8:00	0	0:00 – 8:00	0	0:00 – 8:00
4	8:03 – 8:15	15	8:41 – 8:51	4	8:03 – 8:15
5	8:18 – 8:28	12	8:53 – 9:03	5	8:18 – 8:28
6	8:33 – 8:48	13	9:21 – 9:36	3	8:39 – 8:54
10	8:55 – 9:10	14	9:48 – 9:57	1	8:56 – 9:06
1	9:26 – 9:41	16	10:38 – 10:53	10	9:10 – 9:25
3	9:10 – 9:18	20	10:55 – 11:05	6	9:32 – 9:47
9	9:45 – 9:57	19	11:17 – 11:30	7	9:51 – 10:11
12	10:01 – 10:11	17	11:35 – 11:45	11	10:16 – 10:36
15	10:13 – 10:23	18	12:29 – 12:44	8	10:42 – 10:50
14	10:45 – 10:54	0	13:18	2	10:54 – 11:09
13	11:06 – 11:21			9	11:23 – 12:05
11	11:45 – 12:05			21	12:46 – 12:55
7	12:10 – 12:30			0	14:33
8	12:38 – 12:48				
2	12:50 – 13:05				
18	13:36 – 13:51				
16	14:10 – 14:25				
21	15:20 – 15:29				
20	16:21 – 16:31				
19	16:43 – 16:50				
17	17:01 – 17:11				
0	18:07				
Working Time	**10.07 Hrs.**	**4:46 Hrs**		**6:03 Hrs.**	
Dist. Covered	**392.7 Km.**	**174.30 Km**		**303.0 Km**	

Article DOI: 10.52589/AJMSS-IBKASWQH
DOI URL: https://doi.org/10.52589/AJMSS-IBKASWQH

Table 3: Optimal Routes for Three Vehicles with Time Windows (TW)

Vehicles					
Vehicle A1		**Vehicle A2**		**Vehicle A3**	
Route	**TW**	**Route**	**TW**	**Route**	**TW**
0	0:00 – 8:00	0	0:00 – 8:00	0	0:00 – 8:00
15	8:02 – 8:09	3	8:13 – 8:32	4	8:03 – 8:15
12	8:15 – 8:23	1	8:34 – 8:44	5	08:18-08:28
19	8:25 – 8:33	10	8:48 – 9:03	6	8:33 – 8:48
11	8:35 – 8:40	7	9:11 – 9:31	18	9:27 – 9:42
14	9:08 – 9:12	11	8:36 – 9:56	16	10:01 – 10:16
13	9:34 – 9:44	8	10:02 – 10:12	20	10:18 – 10:28
0	9:46 – 9:56	2	10:14- 10:29	21	11:20 – 11:29
		9	10:43 – 10:55	0	13:07
		0	11:14		
Working time	**3:18 Hours.**		**1:10 Hours.**		**5:02Hours**
Distance	**153.7 Km**		**88.20 Km**		**259.5 Km**

Table 4: Optimal Routes for Four Vehicles with Time Windows (TW)

Vehicles							
Vehicle A1		**Vehicle A2**		**Vehicle A3**		**Vehicle A4**	
Route	**TW**	**Route**	**TW**	**Route**	**TW**	**Route**	**TW**
0	0:00 – 8:00	0	0:00 – 8:00	0	0:00 – 8:00	0	0:00 – 8:00
18	8:35 – 9:42	3	8:17 – 8:32	4	8:03 – 8:15	15	8:41 – 8:51
16	10:01 – 10:16	1	8:34 – 8:44	5	8:18 – 8:28	12	8:53 – 9:03
20	10:18 – 10:28	10	8:48 – 9:03	6	8:33 – 8:42	19	9:21 – 9:36
21	10:30 – 10:39	7	9:11 – 9:31	0	8:55	17	9:39 – 9:49
0	12:07	11	9:36 – 9:56			14	10:08 – 10:17
		9	10:12 – 10:24			13	10:29 – 10:44
		2	10:38- 10:53			0	11:18
		8	11:05 -11:15				
		0	11:25				
Working time	**4:15 Hours**	**3:08 Hours**		**0:55 Hours**		**3:18 Hours**	
Distance	**242.8 Km**	**99 Km**		**44.9 Km**		**153.90km**	

Article DOI: 10.52589/AJMSS-IBKASWQH
DOI URL: https://doi.org/10.52589/AJMSS-IBKASWQH

ISSN: 2689-5323
Volume 7, Issue 2, 2024 (pp. 96-111)

RESULTS

It is worthy of note from the results that, if the Abuja Post Office were to use one vehicle, then with the optimal route as shown in Table 2, the vehicle will return to the depot at about 18:07 p.m. after covering a distance of 392.7km and a working time of about 10 hours. Whereas, when two vehicles are used for mail clearance, the latest vehicle A1 returning to the depot at 14:33 p.m. implies a working time of about 6 hours. But both will cover a total distance of 477.30 km. That is not all, for using three vehicles the latest vehicle will come back to the depot at 13:07p.m. implying a working time of about 5 hours and all three vehicles will cover a distance of 501.4 km. Lastly, the analysis for four vehicles shows that the latest vehicle will come back to the depot at 12:07 p. musing about 4 hours of working time. All four vehicles cover a total distance of 540.60 km.

Table 5: Number of Vehicles with Daily Fuel Consumption

No. of Vehicles			1	2	3	4
Distance covered			392.70	477.30	501.40	540.60
Consumption rate (L/100km)			9.5	9.5	9.5	9.5
Average Petrol Price (N/litre)	142	Daily Cost (N)	5,297.52	6,438.78	6,763.89	7,292.69
		Annual Cost (N)	1,658,124.69	2,015,337.20	2,117,096.32	2,282,613.22
	163	Daily Cost (N)	6,080.96	7,390.99	7,764.18	8,371.19
		Annual Cost (N)	1,903,342.40	2,313,380.03	2,430,188.03	2,620,182.78
	213	Daily Cost (N)	7,646.29	9,658.17	10,145.83	10,939.04
		Annual Cost (N)	2,393,288.77	3,023,005.80	3,175,644.48	3,423,919.83

Since NIPOST is currently using three vehicles, we draw our comparison using three vehicles, see Table 6 below, with the average Petrol price (N/Litre) put at N142, N163 and N213 per litre as shown in Table 6.

Article DOI: 10.52589/AJMSS-IBKASWQH
DOI URL: https://doi.org/10.52589/AJMSS-IBKASWQH

Table 6: Savings Analyses

Average Petrol Price	Annual savings
142	N3, 563,000 – N2,117,096.32 = N1, 445,903.68
163	N3, 563,000 – N2,430,188.03 = N1,132,811.97
213	N3, 563,000 – N2,489,580.09 = N1,073,419.91

Computational Experiments

Almost from the first computational experiments, a set of problems became the test-bed for both heuristic and exact investigations of the VRPTW. Solomon (1987) proposed a set of 168 instances that have remained the leading test set ever since. For the researchers working on heuristic algorithms for the VRPTW need for bigger problems, Homberger and Gehring (1980) proposed a series of extended Solomon problems. These larger problems have as many as 1000 customers and several have been solved by exact methods.

The Solomon Instance

The test sets reflect several structural factors in vehicle routing and scheduling such as geographical data, number of customers serviced by a single vehicle and the characteristics of the time windows (e.g., tightness, positioning and the fraction of time-constrained customers in the instances). Customers are distributed within a $[0,100]^2$ square. The instances are divided into 3 groups (test-sets) denoted R1, R2, and C1. Each of the test sets contain between 8 and 12 instances. In R1 and R2 the geographical data is randomly generated by a random uniform distribution. In the test sets C1 the customers are placed in clusters.

In the test sets R1, C1 and RC1 the scheduling horizon is short permitting approximately 5 to 10 customers to be serviced on each route. This makes the problems very hard to solve exactly and they have not been used until recently to test exact methods. The time windows for the test sets C1 are generated to permit good, maybe even optimal, cluster-by-cluster solutions. For each class of problems the geographical position of the customers is the same in all instances whereas the time windows are changed.

Each instance has 100 customers, but by considering only the first 25 or 50 customers, smaller instances can easily be generated. It should be noted that for the RC-sets this results in the customers being clustered since the clustered customers appear at the beginning of the file. Travel time between two customers is usually assumed to be equal to the travel distance plus the service time at the predecessor customer.

Computational Results

This section reviews the results obtained by the best exact algorithms for the VRPTW. All are based on the coded Lexisearch algorithm using C^{++}. The tables 17 through 22 present the solutions for the six different sets of the Solomon instances that have been solved to optimality. Column *K* indicates the number of vehicles used in the optimal. Kohl, Desrosier, Madsen, Solomon, and Soumis (1999) solved 50 of the 87 Solomon problems with narrow time windows, but with different travel times. Whereas all the above-mentioned papers compute the travel times using one decimal point precision and truncation, time and cost is computed differently in Desrochers, Desrosier, and Solomon (1992).

Article DOI: 10.52589/AJMSS-IBKASWQH
DOI URL: https://doi.org/10.52589/AJMSS-IBKASWQH

Furthermore, solutions to all C1 instances were reported for the first time by Kohl and Madsen (1997), who used a Lagrangian relaxation approach. As discussed in Cordeau, Desaulniers, Desrosier, Solomon and Soumis (2002), the optimal algorithm of Kohl, Desrosier, Madsen, Solomon, and Soumis (1999) solved 69 of the 87 Solomon benchmark short horizon problems to optimality. Eleven additional problems were solved by Larsen (1999), Cook and Rich (1999), and Kallehauge, Larsen, and Madsen (2000). Recently, Irnich and Villeneuve (2016) were successful in closing three additional instances. Four 100-customer instances are still open.

Table 17: Optimal Solutions for the R1 Instances by KDMSS

Problem	K	Dist.	Lexi. Dist.	Problem	K	Dist.	Lexi. Dist.
R101.	25	8 617.1	8615.7	R107.	25	4 424.3	4424.2
R101.	50	12 104.4	11203.5	R107.	50	7 711.1	7711.0
R101.	100	20 1637.7	201,636.7	R107.	100	534.8	543.7
R102.	25	7 547.1	7547	R108.	25	4 397.3	4397.3
R102.	50	11 909	11908	R108	50	6 617.7	6617.7
R102.	100	18 1466.6	181464.7	R108.	100	1235.3	1234.6
R103.	25	5 454.6	5454.4	R109.	25	5 441.3	5441.3
R103.	50	9 772.9	9772	R109.	50	8 786.8	8786.8
R103.	100	14 1208.7	141207.9	R109.	100	13 1146.9	1146.5
R104.	25	4 416.9	4416.9	R110.	25	5 444.1	5444.0
R104.	50	6 625.4	6625.4	R110.	50	7 697	7697
R104.	100	11 971.5	11971.3	R110.	100	12 1068	121,067
R105.	25	6 530.5	6531	R111.	25	4 428.8	4428.5
R105.	50	9 899.3	9899.4	R111.	50	7 707.2	7706.7
R105.	100	15 1355.3	151354.8	R111.	100	12 1048.7	121,048.5
R106.	25	5 465.4	5465.3	R112.	25	4 393	4393
R106.	50	8 793	8793	R112.	50	6 630.2	6630.2
R106.	100	131234.6	131230				

Table 18: Optimal Solutions for the C1 Instances by KDMSS

Problem	K	Dist.	Lexi. Dist	Problem	K	Dist.	Lexi. Dist
C101.	25	3 191.3	3191.3	C106.	25	3 191.3	3191.2
C101	.50	5 362.4	5362.3	C106.	50	5 362.4	5362.3
C101.	100	10 827.3	10826.5	C106.	100	10 827.3	10827.3
C102.	25	3 190.3	3190.2	C107.	25	3 191.3	3191.3
C102.	50	5 361.4	5361.3	C107.	50	5 362.4	5362.4
C102.	100	10 827.3	10826.7	C107.	100	10 827.3	10827.3
C103.	25	3 190.3	3191	C108.	25	3 191.3	3191.1
C103.	50	5 361.4	5361.4	C108.	50	5 362.4	5362.4
C103.	100	10 826.3	10826.2	C108.	100	10 827.3	10827.3
C104.	25	3 186.9	3185.9	C109.	25	3 191.3	3191.2

Article DOI: 10.52589/AJMSS-IBKASWQH
DOI URL: https://doi.org/10.52589/AJMSS-IBKASWQH

African Journal of Mathematics and Statistics Studies
ISSN: 2689-5323
Volume 7, Issue 2, 2024 (pp. 96-111)
www.abjournals.org

C104.	50	5 358	5358	C109.	50	5 362.4	5362.4
C104.	100	10 822.9	10822.8	C109.	100	10 827.3	10827.2
C105.	25	3 191.3	3191.2				
C105.	50	5 362.4	5362.3				
C105.	100	10 827.3	10827.2				

Table 19: Optimal Solutions for the RC1 Instances by KDMSS

Problem	K	Dist.	Lexi. Dist	Problem	K	Dist.	Lexi. Dist.
RC101.	25	4 461.1	44602	RC105.	25	4 411.3	4410.5
RC101.	50	8 944	8940	RC105.	50	8 855.3	8850.3
RC101.	100	15 1619.8	151610	RC105.	100	15 1513.7	151512.4
RC102.	25	3 351.8	3350.5	RC106.	25	3 345.5	3344.9
RC102.	50	7 822.5	7822.0	RC106.	50	6 723.2	6722.6
RC102.	100	14 1457.4	141456.3	RC103.	25	3 332.8	3325.9
RC107.	25	3 298.3	3290.7				
RC103.	50	6 710.9	6710.9	RC107.	50	6 642.7	6642.0
RC103.	100	11 1258	111250	RC107.	100	12 1207.8	121200.4
RC104.	25	3 306.6	33.6.6	RC108.	25	3 294.5	3294.1
RC104.	50	5 545.8	5540.5	RC108.	50	6 598.1	6597.4

CONCLUSIONS

In this study, we have applied the Lexisearch algorithm, an Excel solver to solve the Abuja Post Office VRP with time windows and the results obtained show that the total distance covered increases with increase in the number of vehicles which is expected. This is because they all have to make feasible tours. It can be seen from the analysis that, for three vehicles, the latest vehicle returns to the depot at about 13:03 hours as against one vehicle which returns after about ten hours. It is evaluated and found that, with their current transportation schedule an annual savings of N1, 445,903.68, N1,132,811.97 and N1,073,419.91 for N142, N163, N213 and N660 per litre respectively are realised.

Article DOI: 10.52589/AJMSS-IBKASWQH
DOI URL: https://doi.org/10.52589/AJMSS-IBKASWQH

REFERENCES

Ahmed, Z. H. (2016). A new reformulation and an exact algorithm for the quadratic assignment problem, *Indian Journal of Science and Technology* 6(4), 4368–4377.

Ahmed, Z. H. (2013). A new reformulation and an exact algorithm for the quadratic assignment problem, *Indian Journal of Science and Technology*, *6*(4), 4368–4377.

Baldacci, R., Mingozzi, A. and Roberti, R. (2012). Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. *European Journal of Operations Research* 218**(**1), pp 1–6.

Erdoğan, G. (2017) An open source Spreadsheet Solver for Vehicle Routing Problems. *Computers and Operations Research,* 84(4), pp 62-72.

Farid, G. (2019). Evolutionary Approach for Energy Minimising Vehicle Routing Problem with Time Windows and Customers' Priority Seyed, *International Journal of Transportation Engineering*, vol. 6, no. 3, pp. 24 - 32.

Gedik, R., Kirac. E., Milburn, A. B. and Rainwater, C. (2017) A constraint programming approach for the team orienteering problem with time windows. *Computers and Industrial Engineering*, vol. 107, no. 7, pp 178 – 195.

Haitao, X., Pan, P. and Feng, B. (2018). Dynamic Vehicle Routing Problems with Enhanced Ant Colony Optimization, *Journal of Discrete Dynamics in Nature and Society*. https://doi.org/10.1155/2018/1295485.

He, B., Song, R., He, S. and Xu, Y. (2014) High-speed rail train timetabling problem: A time- space network based method with an improved branch-and-price algorithm, *Mathematical Problems in Engineering*, vol. 34, no 2, pp 45 – 51.

Li, M., Liang, L. and Lu, Y. (2017) "A Model of Vehicle Routing Problem Based on Improved Tabu Search Algorithm". *Journal of Highway and Transportation Research and Development, vol.* 34 no. 10, pp. 108-114.

Lakshmi, G.V. (2013). Lexisearch Approach to Travelling Salesman *Problem Journal of Mathematics* (IOSR-JM) e-ISSN: 2278-5728,p-ISSN: 2319-765X, 6(4), PP 01-08.

Moradi, B. (2020) The new optimization algorithm for the vehicle routing problem with time windows using multi-objective discrete learnable evolution models. *Soft Computing* vol. 24, pp 6741–6769. https://doi.org/10.1007/s00500-019-04312-9.

Pandit, S. N. (1963). "*Some quantitative combinatorial search problems*", Indian Institute of Technology, Kharagpur, India.

Solomon, M. M. (1987) Algorithms for the vehicle routing and scheduling problem with time window constraints. *Operations Research*, vol. 35, pp 254–265.

Tanga, J., Pan, Z. H., Fung, R. K. and Lau, H. (2009) Vehicle routing problem with fuzzy time windows. *Fuzzy Sets System*, vol. 160, pp. 683–695.

Tompkin, A., Batista, R. and Vigo, E. (2017). The South African Post Office supply chain Design Route Optimization, BPJ 420 Final Report.

Wang, J., Zhou, L., Yue, Y., Tang, J. and Bai, Z., (2018) Optimising High-Speed Railroad Timetable with Passenger and Station Service Demands: A Case Study in the Wuhan-Guangzhou Corridor. *Journal of Advanced Transportation*, vol. 10, no. 2, pp 3 – 11.

Walid, A. (2020) Vehicle routing Problem with time window constraint. *Cadernos do LOGIS-UF* vol. 2019, no 2 pp 1 - 38

Zhou, L., Tong, L.C., Chen, J., Tang, J. and Zhou, X. (2017) Joint optimization of high-speed train timetables and speed profiles: A unified modelling approach using space-time-speed grid networks. *Transport Research Journal*, Part B Methodology, vol. 97, pp 157–181.

Article DOI: 10.52589/AJMSS-IBKASWQH
DOI URL: https://doi.org/10.52589/AJMSS-IBKASWQH

African Journal of Mathematics and Statistics Studies
ISSN: 2689-5323
Volume 7, Issue 2, 2024 (pp. 96-111)

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

Article DOI: 10.52589/AJMSS-IBKASWQH
DOI URL: https://doi.org/10.52589/AJMSS-IBKASWQH