Optimizing Bus Routes in Lucknow Region using VRPTW

Rahul Kumar(210804)

06/04/2024

Abstract

This report describes the application of the Vehicle Routing Problem with Time Windows (VRPTW) to optimize bus routes in Lucknow, India. The model aims to minimize the total travel distance and operational costs associated with deploying buses while adhering to service times and passenger demand. The report provides a detailed overview of the methodology, data sources, and results obtained from the optimization process.

1 Introduction

Public transportation systems play a crucial role in urban mobility, providing essential services for millions of people worldwide. In cities like Kanpur, India, efficient management of bus routes is essential for ensuring timely and cost-effective transportation services. The Vehicle Routing Problem with Time Windows (VRPTW) is a mathematical optimization problem well-suited for optimizing bus routes by determining the most efficient routes and schedules while considering time constraints at bus stops.

2 Objective

The objective of this study is to optimize bus routes in Lucknow region using the VRPTW model to minimize total travel distance and operational costs. By leveraging mathematical optimization techniques, we aim to design efficient bus routes that meet passenger demand, adhere to service times, and reduce the number of buses required for operation.

3 Data Description

The optimization model utilizes the following data sources:

- **Locations**: A list of 18 locations in Lucknow region, including starting and ending depots (Charbagh and Fatehpur), bus stops, and important transit points.

Location	**Latitude**	**Longitude**
Charbagh	26.8477	80.9470
Raibareli	26.2325	81.2331
Tiloi	26.1711	81.2155
Dewan	28.6187	79.8087
Salon	26.1522	81.2830
Bachrawan	26.4681	81.3406
Shivgarh	26.2636	81.3070
Safedabad	26.4172	80.7297
Haidergarh	26.7194	81.1133
Lalganj	25.8706	81.7020
Alambagh	26.8462	80.9331
Barabanki	26.9353	81.1951
Mohanlalganj	26.8215	81.0406
Ramnagar	26.5734	81.3752
Kaiserbagh	26.8483	80.9190
Barabanki Naya	26.9255	81.1934
Awadh Bus Station (Kamta)	26.9449	80.9164
Fatehpur	25.9304	80.8133

- **Distance Matrix**: A matrix representing the travel distance between each pair of locations, calculated using the Haversine formula.

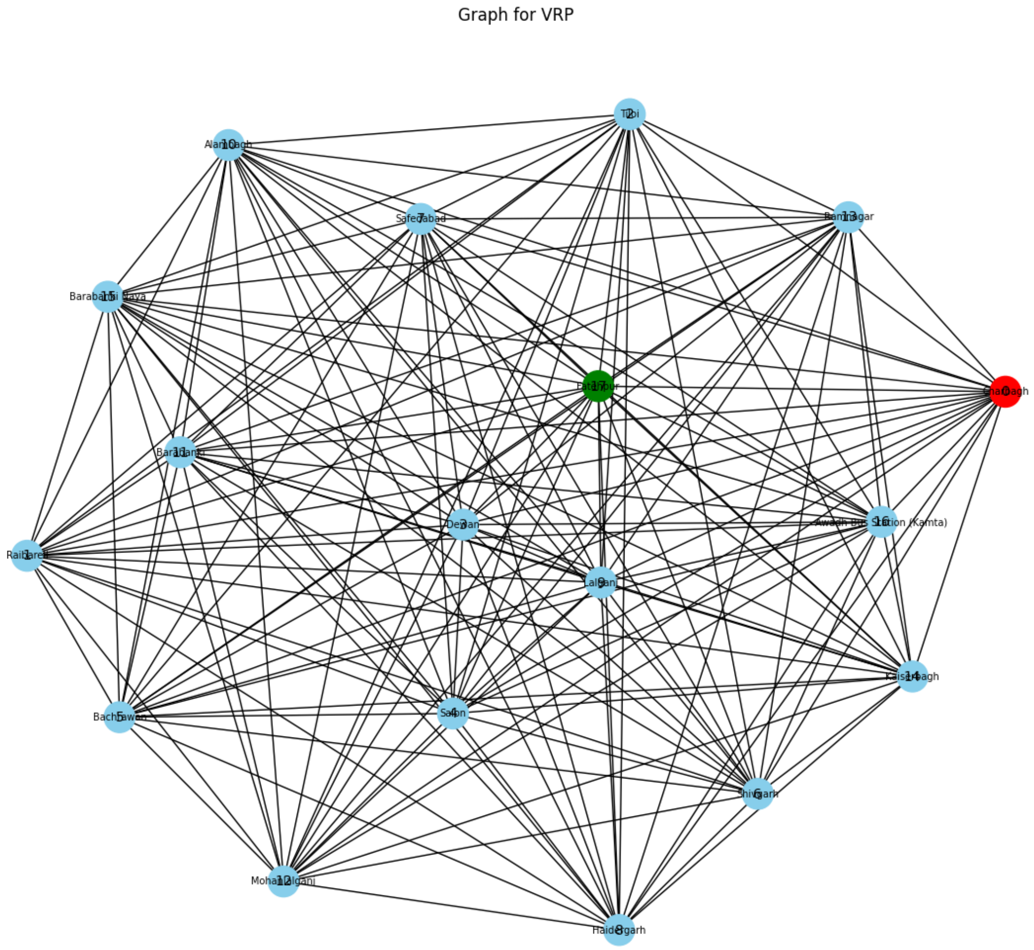

Figure 1: Routes for Vehicle Routing Problem with Time Windows (VRPTW)

- **Node Labels**: A dictionary mapping each location index to its corresponding name for easy reference and visualization.
- **Demand**: An estimate of the number of passengers expected to board at each bus stop, excluding depots. In our case we have dynamic demand of 10-15 people boarding the bus from each bus stops.
- **Time Windows**: Time windows specifying the opening and closing times for service at each bus stop, excluding depots.

Stop Name	Arrival Time Window	
	Open	Close
Charbagh	06:00	07:30
Raibareli	19:00	20:00
Tiloi	19:00	20:30
Dewan	19:30	20:30
Salon	19:00	21:00
Bachrawan	18:00	19:00
Shivgarh	19:00	20:00
Safedabad	17:00	18:00
Haidergarh	13:00	14:30
Lalganj	19:30	20:30
Alambagh	08:00	09:30
Barabanki	15:00	16:30
Mohanlalganj	11:00	12:30
Ramnagar	16:00	17:30
Kaiserbagh	10:00	11:30
Barabanki Naya	14:00	15:30
Awadh Bus Station (Kamta)	12:00	13:30
Fatehpur	21:50	23:30

- **Vehicle Capacity**: The maximum number of passengers that each bus can accommodate. All buses have same capacity of 50 passengers.

4 Methodology

The VRPTW model is formulated to minimize total travel distance and operational costs while satisfying capacity constraints and time windows for service. The model includes the following components:

- **Objective Function**: Minimize the total travel distance and operational costs associated with deploying buses.
- **Constraints**: Ensure that each bus route starts and ends at the designated depots, each bus serves each bus stop exactly once, capacity constraints are met, and service times adhere to specified time windows.
- **Optimization Algorithm**: The optimization problem is solved using the Gurobi Optimization Software, employing exact optimization algorithms to identify globally optimal solutions within a reasonable computational time frame.

5 The Model

The Vehicle Routing Problem with Time Windows (VRPTW) is defined by a fleet of vehicles, V, a set of customers, C, and a directed graph G. Typically, the fleet is considered to be homogeneous, meaning all vehicles are identical. The graph consists of $|C| + 2$ vertices, where the customers are denoted $1, 2, ..., n$, and the depot is represented by the vertices 0 ("the starting depot") and $n + 1$ ("the returning depot"). The set of all vertices, N, includes $0, 1, ..., n + 1$, and the set of arcs, A, represents direct connections between the depot and the customers and among the customers. No arcs end at vertex 0 or originate from vertex $n + 1$. Each arc (i, j), where $i \neq j$, is associated with a cost c_{ij} and a time t_{ij}, which may include service time at customer i.

Parameters:

- V: Set of vehicles.
- C: Set of customers.
- G: Directed graph representing connections between the depot and customers, and among customers.
- c_{ij}: Cost associated with arc (i, j).

- t_{ij}: Time associated with arc (i,j), including service time at customer i.
- Q: Capacity of each vehicle.
- D: Operational cost of each vehicle.
- d_i: Demand of customer i.
- a_i, b_i: Time window for customer i, representing the earliest and latest time for service.
- a_0, b_{n+1}: Time window for the depot, representing the scheduling horizon.

Sets:

- V: Set of vehicles.
- C: Set of customers.
- N: Set of all vertices, including depot and customers.

Decision Variables:

- x_{ijk}: Binary decision variable indicating if vehicle k travels directly from vertex i to vertex j.
- s_{ik}: Time at which vehicle k starts servicing customer i.

Objective: Minimize the total travel cost:

$$\min \sum_{i\in N}\sum_{j\in N}\sum_{k\in V} c_{ij}x_{ijk} + D\sum_{j\in N}\sum_{k\in V} x_{0jk}$$

Constraints:

1. Each customer is visited exactly once:
$\sum_{k\in V}\sum_{j\in N} x_{ijk} = 1 \quad \forall i \in C$
2. Capacity constraint:
$\sum_{i\in C} d_i \sum_{j\in N} x_{ijk} \leq Q \quad \forall k \in V,$
3. Leaving depot constraint:
$\sum_{j\in N} x_{0jk} \leq 1 \quad \forall k \in V$
4. Returning depot constraint:
$\sum_{j\in N} x_{i,n+1,k} \leq 1 \quad \forall k \in V$
5. Time relationship between consecutive vertices:
$x_{ijk}(s_{ik} + t_{ij}s_{jk}) \leq 0 \quad \forall i,j \in N, \forall k \in V$
6. Time window constraint:
$a_i \leq s_{ik} \leq b_i \quad \forall i \in N, \forall k \in V$
7. Flow conservation at each customer
$\sum_{i\in N} x_{ihk} - \sum_{j\in N} x_{hjk} = 0 \quad \forall h \in C, \forall k \in V$
8. Binary decision variable constraint:
$x_{ijk} \in \{0,1\} \quad \forall i,j \in N, \forall k \in V$

Additional Constraints:

- Upper bound on the number of vehicles:

$$\sum_{j\in N}\sum_{k\in V} x_{0jk} \leq \ |V|$$

Note: The nonlinear restrictions can be linearized, and large constants can be adjusted accordingly.

6 Results

The optimization model yields optimal bus routes and schedules that minimize total travel distance and operational costs while meeting service requirements. The results demonstrate significant improvements in route efficiency and cost-effectiveness compared to baseline scenarios.The model is solved using Gurobi, a powerful optimization software. The solution provides optimal bus routes and their corresponding travel times. The results include:

- Routes: A list of stops traversed by each bus, forming individual routes.
- Buses: The specific bus assigned to each route.
- Optimal solution found (tolerance 1.00e-04) in 11.85 seconds
  Best objective 7.621282583518e+04, best bound 7.621282583518e+04, gap 0.0000

Bus	**Route (Start Depot → Stops → End Depot)**
Bus 0	Charbagh (0) → Alambagh (10) → Salon (14) → Dewan (16) → Tiloi (3) → Fatehpur (17)
Bus 1	Charbagh (0) → Ramnagar (13) → Bachrawan (5) → Raibareli (1) → Tiloi (2) → Fatehpur (17)
Bus 2	Charbagh (0) → Mohanlalganj (12) → Barabanki (11) → Barabanki Naya (15) → Safedabad (7) → Fatehpur (17)
Bus 3	Charbagh (0) → Haidergarh (8) → Shivgarh (6) → Salon (4) → Lalganj (9) → Fatehpur (17)

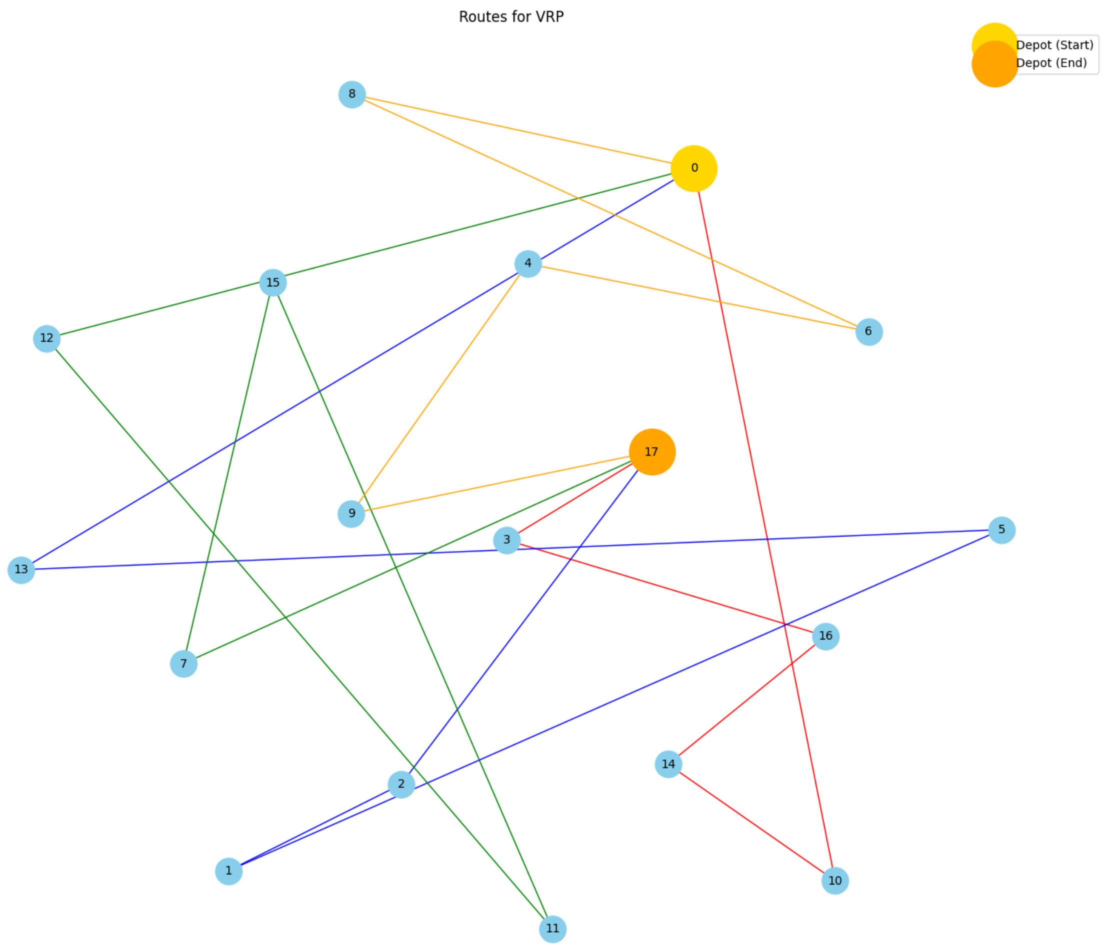

Figure 2: Routes for Vehicle Routing Problem with Time Windows (VRPTW)

7 Conclusion

The study demonstrates the effectiveness of applying the VRPTW model to optimize bus routes in Lucknow region, India, using mathematical optimization techniques. By minimizing travel distance and operational costs while meeting service requirements, the optimized bus routes contribute to enhanced efficiency and service quality in the public transportation system. Future research may explore additional factors such as dynamic demand forecasting and real-time route optimization to further improve system performance and responsiveness.

8 Analysis

The VRPTW model effectively optimizes bus routes by minimizing the total travel cost while adhering to time windows and capacity constraints. The following factors significantly impact the objective function:

- Travel Distances: Longer distances between stops increase the overall travel cost. Optimizing routes to minimize travel distance is crucial.
- Time Window Flexibility: Stricter time windows at stops may necessitate more buses or less efficient routes. Balancing time window adherence with route efficiency requires careful consideration.
- Passenger Demand Distribution: Uneven passenger demand across stops can lead to imbalanced routes. Spreading demand more evenly might improve route efficiency.
- Bus Capacity: Lower bus capacity restricts the number of passengers served in a single route, potentially requiring more buses. Finding the optimal balance between bus capacity and number of buses is essential.

By analyzing the impact of these factors, policymakers and transportation authorities can make informed decisions. Strategies like adjusting bus schedules, stop locations, or even passenger fares could be explored based on the model's insights.

9 References

1. VRPTW Optimization: various Research Papers of VRP
2. Exact Optimization Methods: Research Papers of VRP
3. Gurobi in Transportation Optimization: Gurobi seminars.
4. Code : code for haversine formula implementation, some use of generative AI models for plotting only.

10 Contributions

1. All work is done by me.

11 Appendix

- libraries used- numpy,matplotlib,networkx,gurobipy,math
- PLEASE INSTALL PYTHON THESE LIBRARIES
- Python Code