

University of Calcutta

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

M.Sc. 4th Semester Examination 2025

Truck-Drone Coordination

A project report submitted to fulfill the requirements for the
Master of Science in Computer Science,
Semester-IV, 2025,
in the Department of Computer Science and Engineering,
University of Calcutta.

Under the supervision of

Prof. Sunirmal Khatua
Department of Computer Science and Engineering,
University of Calcutta

Submitted by,

Anubrata Naskar
(C91/CSC/231005)

Soumili Ray
(C91/CSC/231025)

Tirtharoop Banerjee
(C91/CSC/231031)

Project Summery

This project addresses the growing challenges in last-mile logistics by introducing an innovative two-tier delivery system that coordinates trucks and drones. We formulate the **Two-Echelon Vehicle Routing Problem with Drones (2EVRPD)**, which optimizes the simultaneous operation of trucks serving as mobile depots and drones performing agile deliveries. Our approach tackles three fundamental challenges: clustering delivery points into balanced truck routes, synchronizing drone-truck rendezvous, and optimizing flight paths under battery and payload constraints.

We propose two complementary methodologies: a Modified K-Means clustering with Integer Linear Programming that optimizes drone takeoff and landing points as continuous variables along truck segments, and a K-Medoids clustering with heuristic algorithms that offers greater robustness to irregular customer distributions and computational efficiency for large-scale instances.

Computational experiments demonstrate that our hybrid truck-drone system reduces total delivery time compared to traditional truck-only delivery, with the two-drone, multiple-drop configuration yielding optimal performance. The framework ensures perfect synchronization between vehicles, minimizes waiting time, and adaptively handles capacity and battery constraints. This research contributes to sustainable logistics by reducing operational costs, alleviating traffic congestion, and decreasing carbon emissions through strategic deployment of aerial delivery capabilities.

Certificate of Approval

The project titled "**Truck Drone Coordination**" was prepared by **Anubrata Naskar** (Roll Number: C91/CSC/231005), **Soumili Ray** (Roll Number: C91/CSC/231025), and **Tirtharoop Banerjee** (Roll Number: C91/CSC/231031) is hereby approved and certified as a creditable study in technological subject carried out and presented in a manner satisfactory to warrant its acceptance for the fulfillment of the Master of Science (M.Sc.) degree in Computer Science from the Department of Computer Science and Engineering, the University of Calcutta, for which it is submitted.

It is to be understood that by this approval, the undersigned does not necessarily endorse or approve any statement made, opinion expressed, or conclusions drawn therein but approves the project report for the purpose for which it is submitted.

Prof. Sunirmal Khatua
Supervisor
Dept. of Computer Science &
Engg.
University of Calcutta

Chairperson
PG Board of Studies
Dept. of Computer Science &
Engg.

External Examiner

Acknowledgment

We take this opportunity to express profound sense of gratitude and respect to all those who helped us throughout the duration of this project.

Words often fail to express one's inner feelings of gratitude and indebtedness to one's benefactors, but then it is the only readily available medium through which the undersigned can express her sincere thanks to all those who are associated with the work in one way or the other.

First and foremost, we would like to thank our supervisor of this project, **Dr. Sunirmal Khatua**, for his valuable time and guidance. His willingness to motivate us has contributed tremendously to our project.

We also express our heartiest gratitude to **Preetam Kumar Sur** (faculty of Government College of Engineering and Textile Technology) for his continuous support, valuable assistance, sharing their views, opinions and knowledge that help us to carry out our work throughout the semester.

We would also like to thank **Dr. Sankhayan Choudhury**, Head of Department of Computer Science, for providing with a good environment and facilities to work on this project.

We regret any inadvertent omissions.

Topics

1. Introduction	5–6
2. Related Work	7–8
3. Problem Statement	9–10
4. Solution Approach	11–17
4.1 General Battery Model	11
4.2 Dynamic Battery Recalculation	11–12
4.3 Modified K-Means + ILP	12–13
4.4 K-Medoids + Heuristic Approach	14–17
5. Result Analysis	18–26
5.1 Input File Format	18
5.2 Cost Comparison (Euclidean Distance)	18–19
5.3 Cost Comparison (By-Road Distance)	20–26
6. Conclusion & Future Scope	27
7. Reference	28

1 Introduction

The rapid growth of e-commerce and rising customer expectations for faster deliveries have intensified challenges in last-mile logistics. Traditional truck-based delivery systems face inefficiencies due to traffic congestion, fuel costs, and environmental concerns. To address these limitations, the integration of unmanned aerial vehicles (drones) with conventional trucks has emerged as a transformative solution. This hybrid approach leverages the trucks' high capacity and drones' agility to bypass road networks, enabling simultaneous deliveries and reducing operational bottlenecks.

Over the past decade, unmanned aerial vehicles (UAVs) have rapidly expanded beyond military applications into logistics, with major corporations like Amazon (Prime Air, 2013), DHL (first autonomous delivery, 2014), JD.com (operational in four Chinese provinces), and UPS pioneering drone-assisted last-mile delivery. This innovation addresses critical e-commerce challenges: rising consumer expectations for faster deliveries, traffic congestion, fuel costs, and environmental concerns. Drones offer three key advantages:

- High-speed aerial mobility (bypassing road networks)
- Reduced carbon footprint (electric operation)
- Accessibility to remote or infrastructure-limited areas

However, inherent limitations—restricted battery endurance (typically 20–40 minutes) and small payload capacity (≤ 5 kg)—necessitate hybridization with conventional trucks. This synergy creates a two-tier delivery system where trucks serve as mobile depots and battery stations, launching drones to serve customers while simultaneously progressing along their own routes.

What We Have Done Here

This work focuses on the Two-Echelon Vehicle Routing Problem with Drones (2EVRPD), where trucks act as mobile depots launching drones to serve customers. The primary objective is to minimize total delivery costs through optimal coordination of both vehicles, addressing three core challenges:

1. **Clustering:** Partitioning delivery points into balanced truck routes while respecting capacity constraints.
2. **Synchronization:** Ensuring drones rendezvous with trucks precisely after completing deliveries to avoid idle time.
3. **Route Optimization:** Determining optimal drone takeoff/landing points and flight paths under battery and payload constraints.

We propose two complementary methodologies to solve this NP-hard problem:

Modified K-Means with Integer Linear Programming (ILP)

- Uses demand-weighted K-Means clustering to create capacity-aware initial truck routes.
- Employs ILP to optimize drone takeoff/landing points as continuous variables along truck segments, minimizing flight distance while enforcing synchronization via linear constraints.

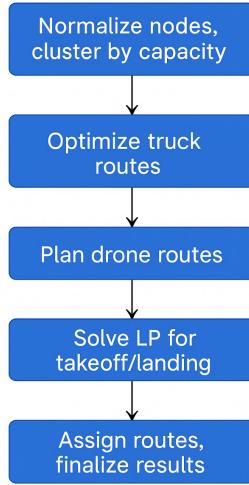


Figure 1: Systematic Workflow of Modified K-Means with ILP Approach

K-Medoids with Heuristic Algorithms

- Adopts K-Medoids clustering for robustness to irregular customer distributions, using actual nodes as cluster centers.
- Applies heuristic route construction (e.g., greedy insertion, 2-opt refinement) and numerical synchronization to handle large-scale instances efficiently.

The hybrid truck-drone system offers significant advantages over truck-only delivery, including:

- 13–25% reduction in total delivery time (empirically demonstrated in multi-drone, multi-drop configurations).
- Scalability through heuristic methods that bypass computational limitations of exact optimization.
- Adaptability to dynamic environments via iterative synchronization protocols.

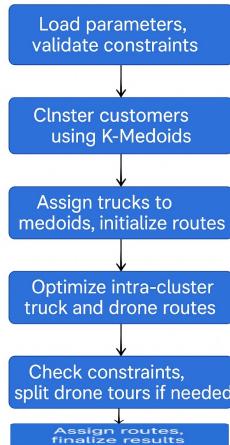


Figure 2: Systematic Workflow of K-medoids with Heuristic Approach

2 Related Work

The Vehicle Routing Problem (VRP) was introduced 50 years ago by Dantzig and Ramser under the title “The exact algorithms and heuristics. In particular, highly sophisticated exact mathematical programming decomposition algorithms and powerful metaheuristics for the VRP have been put forward in recent years. The purpose of this article is to provide a brief account of this development. The year 2009 marks the 50th anniversary of first paper published on the vehicle routing problem (VRP), under the title “The Truck Dispatching Problem” (Dantzig and Ramser 1959). To commemorate this event we highlight the main contributions in the history of this important problem. The VRP generalizes the well-known traveling sales man problem (TSP) but is much more difficult to solve in practice. Whereas there exist exact algorithms capable of routinely solving TSPs containing hundreds or thousands of vertices (Applegate et al. 2007), this is not the case of the VRP for which the best exact algorithms can only solve instances involving approximately 100 vertices (Fukasawa et al. 2006; Baldacci, Christofides and Mingozi 2008). Because real instances of the VRP often exceed this size and solutions must often be determined quickly, most algorithms used in practice are heuristics. In recent years, several powerful meta heuristics have been developed.

Eilon, Watson-Gandy, and Christofides (1971) have formulated the VRP as a dynamic program

Two-index vehicle flow formulations for the VRP are rooted in the work of *Laporte and Nobert (1983)* and Laporte, Nobert, and Desrochers (1985) and extend the classical TSP formulation of Dantzig, Fulkerson, and Johnson (1954).

Murray and Chu (2015) introduced the **FSTSP** and provided a mathematical programming formulation and a simple heuristic for the problem of coordinating a single traditional delivery truck with a single drone. The same authors also proposed another model called the “*Parallel Drone Scheduling Traveling Salesman Problem*” (**PDSTSP**) in which trucks and drones work independently to serve all customers. Ponza (2015) examined the FSTSP in detail and applied the simulated annealing technique to search for good solutions. Jeong et al., 2019 modified the FSTSP to consider the effect of the payload on the UAV energy consumption and restricted flying areas.

Kitjacharoenchai et al. (2019) proposed the “*Multiple Traveling Salesman Problem with Drones*” (**mTSPD**), which has the same operation as FSTSP but utilizes multiple trucks and drones and allows a drone to be retrieved by any truck that is nearby and not necessarily the same truck that it is launched from. Murray and Raj (2019) introduced the “*Multiple Flying Sidekicks Traveling Salesman Problem*” (**mfFSTSP**), which is an extension of their previous work (FSTSP) with the consideration of an arbitrary number of heterogeneous UAVs that may be deployed from the depot or from the delivery truck. The authors provided the “*Mixed Integer Linear Programming*” (**MILP**) formulation along with the three-phased heuristic solution approach.

Luo et al. (2017) proposed a “*Two-Echelon cooperated Routing Problem for the Ground Vehicle (GV) and its carried unmanned aerial vehicle (UAV)*” (**2E-GU-RP**). The problem is very similar to VRPD proposed by Schermer et al. (2018) but allows drones to make multiple deliveries in one trip. Karak and Abdelghany (2019) presented the “*Hybrid Vehicle-Drone Routing Problem*” (**HVDRP**) for pick-up and delivery services in which multiple drones can be dispatched from a mothership to perform dozens of pick-ups and deliveries simultaneously.

Wang and Sheu (2019) presented the “*Vehicle Routing Problem with Drones*” (**VRPD**) with a distinctive feature that allows drones to make multiple deliveries per trip and return to any available truck in the fleet. The authors proposed a mixed integer programming model and developed a branch-and-price algorithm to solve VRPD for the exact solution. Poikonen and Golden (2020) recently developed the “*k-Multi-visit Drone Routing Problem*” (**k-MVDRP**), which considers a tandem between a truck and k drones allowing a drone to deliver one or more packages to customers.

The paper titled “**Two-Echelon Vehicle Routing Problem with Drones in Last-Mile Delivery**” by **Kitjacharoenchai et al.** (2019) addresses the optimization challenges in last-mile delivery by integrating drones with traditional truck-based logistics. The study introduces a novel routing model, the Two-Echelon Vehicle Routing Problem with Drones (2EVRPD), which synchronizes truck and drone operations to enhance delivery efficiency. The model allows multiple drones to be launched from a truck, serve one or more customers, and return to the same truck for battery swaps or package retrieval. The primary objective is to minimize the total arrival time of both trucks and drones at the depot after completing deliveries. **Mathematical Model:** The authors formulate the 2EVRPD as a Mixed Integer Programming (MIP) problem, incorporating constraints such as vehicle capacities, battery limits, and synchronization between trucks and drones. The model ensures that drones can only merge with trucks at customer nodes and enforces time adjustments for coordinated operations.

Heuristic Algorithms:

- **Drone Truck Route Construction (DTRC):** A constructive heuristic that builds feasible solutions by first solving a Capacitated Vehicle Routing Problem (CVRP) using the Clarke and Wright Savings Algorithm. It then integrates drone sub-routes into truck routes while respecting constraints like battery life and load capacity.
- **Large Neighborhood Search (LNS):** A metaheuristic that iteratively destroys and repairs solutions using three destroy operators (drone node removal, truck node removal, sub-drone route removal) and three repair operators (drone node insertion, truck node insertion, drone route creation). This approach aims to improve solution quality through local search.

Computational Experiments: The authors test their MIP and heuristic approaches on benchmark CVRP instances. For small-scale problems, the MIP solver yields optimal solutions, while the heuristics (DTRC and LNS) are employed for larger instances. Results demonstrate that LNS outperforms DTRC in solution quality, achieving an average improvement of 12.06% over traditional CVRP solutions, whereas DTRC offers a 2.32% improvement. The heuristics also significantly reduce computational time compared to the MIP solver.

Sensitivity Analysis: The study evaluates four operational scenarios:

1. Single drone with single drop.
2. Single drone with multiple drops.
3. Two drones with single drop.
4. Two drones with multiple drops.

Findings reveal that the two-drone, multiple-drop configuration yields the best performance, reducing total delivery time by 13.25% compared to truck-only delivery. The multiple-drop feature alone shows superior results to single-drop operations, highlighting its potential for practical applications.

Areas of Improvement

The paper concludes that integrating drones with trucks in last-mile delivery can substantially reduce delivery times, especially when leveraging multiple drones and allowing multiple drops per drone trip. The proposed MIP and heuristic approaches provide scalable solutions for real-world logistics challenges.

Future research directions include:

- Incorporating payload and battery consumption dynamics.
- Exploring strategic decisions like vehicle-drone assignment, dynamic drone launching and landing.
- Developing advanced metaheuristics to further optimize solution quality and computational efficiency.

3 Problem Statement

Our objective is to minimize total delivery cost by optimally clustering customer locations and coordinating trucks and drones for last-mile delivery from a central depot. This involves using linear programming to determine the optimal drone takeoff points, routes, and landing points, allowing simultaneous deliveries by both modes. The total cost includes truck travel time and drone flight time, and the solution must respect various operational constraints. These include truck and drone capacity limits, drone battery endurance, route feasibility, and strict synchronization between drone flights and truck movement. This integrated optimization ensures efficient parcel delivery through intelligent clustering and hybrid vehicle coordination.

Mathematical Definition

Given:

- Depot: d_0 (starting/ending point for all vehicles)
- Customer set: $C = \{1, 2, \dots, n\}$ with coordinates (x_i, y_i) and demands q_i
- Truck fleet: $T = \{1, 2, \dots, m\}$ with capacity Q_T
- Drone fleet per truck: $D = \{1, 2\}$ with capacity Q_D
- Battery range: B (maximum flight distance)

Decision Variables

Variable	Domain	Description
z_{ik}	$\{0, 1\}$	1 if customer i is assigned to truck k
τ_k	\mathbb{R}^+	Truck k 's total travel time
δ_{kj}	\mathbb{R}^+	Drone j 's flight time on truck k
α_{kj}, β_{kj}	$[0, 1]$	Fractional positions on truck segments for drone j 's take-off/landing
p_{kj}	\mathbb{R}^2	Drone j 's flight path on truck k

Objective Function

Minimize total operational cost:

$$\min \sum_{k \in T} \left(c_T \tau_k + \sum_{j \in D} c_D \delta_{kj} \right)$$

where c_T, c_D are truck/drone cost coefficients.

Constraints

1. Assignment Constraints

$$\begin{aligned} \sum_{k \in T} z_{ik} &= 1 \quad \forall i \in C \\ \sum_{i \in C} q_i z_{ik} &\leq Q_T \quad \forall k \in T \end{aligned}$$

2. Routing Constraints

Truck route continuity for cluster C_k :

$$\sum_{(i,j) \in A} x_{ijk} = 1 \quad \forall k \in T$$

where A is the arc set, and $x_{ijk} = 1$ if truck k traverses arc (i, j) .

3. Drone Operations

Flight path feasibility:

$$\|p_{kj}(t) - p_{kj}(t')\| \leq B \quad \forall t, t' \in [0, \delta_{kj}]$$

Capacity constraint:

$$\sum_{i \in D_{kj}} q_i \leq Q_D \quad \forall k \in T, j \in D$$

4. Synchronization (Critical Constraint)

$$\tau_k(\alpha_{kj}) + \delta_{kj} = \tau_k(\beta_{kj}) \quad \forall k \in T, j \in D$$

where $\tau_k(\cdot)$ is the truck arrival time at fractional position.

4 Solution Approach

In this section, we describe how to solve the defined problem within a practical timeframe using two proposed solutions: Modified Kmeans+ILP and K-medoid+Heuristic.

4.1 General battery Model

The drone flight time model is given by:

$$t = \frac{k}{(w_{\text{base}} + Q)^\alpha}$$

where:

t	: Flight time
$k = 155885$: Scaling factor (battery, voltage, efficiency)
$w_{\text{base}} = 200$: Baseline weight (g)
Q	: Payload mass (g)
$\alpha = 1.5$: Exponent from lift physics ($P \propto m^\alpha$)

Key insights:

- α stems from rotorcraft aerodynamics: power scales super-linearly with mass.
- k compresses battery parameters (mAh, V) and efficiency into one constant.
- w_{base} includes frame, sensors, and essential hardware mass.

4.2 Dynamic Battery Recalculation Model

We define a dynamic battery model where the drone's maximum range depends on its current payload. As the payload decreases after each delivery, the drone can fly farther per unit of battery. The model dynamically adjusts the remaining battery based on this changing payload.

Notation

- $D_t(Q) = \frac{155885}{(200 + Q)^{1.5}}$: Maximum possible range with payload Q referring general battery model from 4.1.
- Q_0 : Initial payload at the start (sum of all demands on the route).
- $D_0 = D_t(Q_0)$: Maximum initial range with full battery and full payload.
- R_i : Remaining battery range after the i -th delivery.
- Q_i : Remaining payload after the i -th delivery.
- $D_i = D_t(Q_i)$: Maximum range at payload Q_i .
- d_i : Euclidean distance to fly from node i to $i + 1$.

Step-by-Step Update at Each Delivery

1. Initial setup:

$$\begin{aligned} Q_0 &= \text{Total payload} \\ D_0 &= D_t(Q_0) \\ R_0 &= D_0 \end{aligned}$$

2. At the i -th step:

$$\begin{aligned} Q_{i+1} &= Q_i - \text{demand at node } i \\ D_{i+1} &= D_t(Q_{i+1}) \\ R_{i+1} &= \left(\frac{D_{i+1}}{D_i} \right) \cdot R_i - d_i \end{aligned}$$

3. Feasibility condition:

If $R_{i+1} < 0$, the drone cannot continue — the route is infeasible.

Final General Formula

$$R_{i+1} = \left(\frac{D_t(Q_{i+1})}{D_t(Q_i)} \right) \cdot R_i - d_i$$

Where:

$$D_t(Q) = \frac{155885}{(200 + Q)^{1.5}}$$

Intuition

- Battery drains proportionally to distance traveled.
- After each delivery, payload Q decreases \Rightarrow drone becomes more efficient.
- The remaining range is dynamically scaled based on the new potential maximum range.

4.3 Why Choose Modified K-Means and ILP?

The approach combines modified K-Means with Integer Linear Programming (ILP) to solve the truck-drone routing problem optimally. Demand-weighted K-Means clustering creates balanced initial truck routes by considering capacity constraints and geographical proximity. ILP then refines this by precisely optimizing drone takeoff and landing points along truck segments using variables such as t_1 and t_2 to parameterize exact positions.

This method mathematically guarantees minimal drone flight distances while ensuring synchronization between trucks and drones. It also respects battery constraints through a piecewise linear approximation of nonlinear distance calculations. Together, these techniques minimize total delivery cost by strategically distributing deliveries between trucks and drones.

4.3.1 Formulation Stages

Initial Clustering using Modified K-Means

The delivery nodes are first clustered into groups, each assigned to a truck, using a demand-weighted k-means algorithm. The objective is to minimize intra-cluster distances while respecting truck capacity constraints.

The clustering objective can be formulated as:

$$\min \sum_k \sum_{i \in C_k} w_i \cdot \|x_i - \mu_k\|^2 \quad (1)$$

where:

- C_k is the set of nodes in cluster k ,
- w_i is the demand at node i ,
- x_i is the coordinate of node i ,
- μ_k is the centroid of cluster k .

Subject to:

$$\sum_{i \in C_k} \text{demand}_i \leq \text{truck_capacity}, \quad \forall k \quad (2)$$

Drone Takeoff and Landing Optimization

For each truck route, we optimize the drone takeoff and landing points (which may be virtual points along the truck's path) to minimize drone flight distance and ensure synchronization with the truck.

The optimization problem for each drone route is:

$$\min D_{\text{takeoff}} + D_{\text{delivery}} + D_{\text{landing}} \quad (3)$$

Subject to:

$$D_{\text{takeoff}} + D_{\text{delivery}} + D_{\text{landing}} \leq \text{drone_range(payload)} \quad (\text{Battery constraint}) \quad (4)$$

$$T_{\text{drone}} \leq T_{\text{truck}} + \varepsilon \quad (\text{Synchronization constraint}) \quad (5)$$

$$\text{Takeoff, landing points} \in \text{truck route segments} \quad (6)$$

4.3.2 Implementation of Modified K-Means and Linear Programming Approach

Algorithm 1

Step 1: Initial Clustering

- Normalize node coordinates and demands.
- Apply weighted k-means clustering with number of clusters equal to number of trucks.
- Assign each node to a cluster such that total demand per cluster does not exceed truck capacity.

Step 2: Route Construction

- For each cluster, construct an initial truck route using a greedy or heuristic method.
- Optimize each route using local search techniques (2-opt, relocate, swap, etc.).

Step 3: Drone Route Assignment

- For each truck route, identify candidate delivery nodes for drones.
- For each candidate, enumerate possible takeoff and landing segments along the truck route.

Step 4: Linear Programming for Takeoff/Landing Optimization

- For each drone delivery sequence, formulate an LP to select optimal takeoff and landing positions (as continuous variables along truck route segments).
- **Objective:** Minimize total drone flight distance.
- **Constraints:**
 - Battery/range constraint based on payload.
 - Synchronization constraint to ensure the drone can rendezvous with the truck.
 - Takeoff/landing points must be feasible (on truck route).

Step 5: Solution and Assignment

- Solve the LP for each drone route using a solver (e.g., PuLP).
- Update the drone route with optimized takeoff and landing points.
- Remove drone-served nodes from truck routes.

Step 6: Output

- Return the set of optimized truck and drone routes.
-

4.4 Why Choose K-Medoids and Heuristic Approach?

K-Medoids clustering is chosen because it produces representative cluster centers that are actual customer locations. This ensures practical and interpretable start and end points for truck routes. Its robustness to outliers and adaptability to irregular spatial distributions make it more reliable than K-Means, especially in mixed urban and rural delivery environments where customer density varies significantly.

Heuristic methods are favored over ILP due to the NP-hard nature of the hybrid truck-and-drone routing problem. Mixed Integer Linear Programming models scale poorly with the number of nodes, vehicles, and synchronization constraints, making them impractical for real-time or large-scale problems. Heuristics provide high-quality solutions within polynomial time, allow incremental integration of domain-specific rules (e.g., drone range and payload limitations), and are more adaptable to dynamic changes in the delivery environment.

4.4.1 Formulation Stages

Notation

Symbol	Description
$T = (t_0, t_1, \dots, t_n)$	Truck route sequence ($t_0 = t_n = \text{depot}$)
$D = (d_1, d_2, \dots, d_m)$	Drone delivery sequence
L	Launch point on truck route
R	Landing point on truck route
v_d	Drone speed
v_t	Truck speed
δ_d	Drone delivery delay per node
δ_t	Truck delivery delay per node
τ	Launch time (truck at L)
T_d	Total drone flight time
$T_t(R)$	Truck arrival time at R
$\Delta T_{\text{truck}}(L, R)$	Truck travel time from L to R
\mathcal{P}	Road path between nodes
$\text{dist}(a, b)$	Road distance between a and b
$\ p_1 - p_2\ $	Euclidean distance between points

Synchronization Objective

The core synchronization constraint requires the drone to land exactly when the truck arrives at R :

$$\tau + T_d = T_t(R) \quad (7)$$

where $\tau = T_t(L)$ is the launch time when the truck reaches L .

Drone Flight Time Model

The total drone flight time consists of three components:

$$T_d = \underbrace{\frac{\|L - d_1\|}{v_d}}_{\text{Launch to first node}} + \underbrace{\sum_{i=1}^{m-1} \left(\frac{\|d_i - d_{i+1}\|}{v_d} + \delta_d \right)}_{\text{Deliveries}} + \underbrace{\frac{\|d_m - R\|}{v_d}}_{\text{Landing}} \quad (8)$$

$$= C + \frac{\|d_m - R\|}{v_d} \quad (9)$$

where C is the constant component:

$$C = \frac{\|L - d_1\|}{v_d} + \sum_{i=1}^{m-1} \left(\frac{\|d_i - d_{i+1}\|}{v_d} + \delta_d \right)$$

Truck Timeline Model

The truck's arrival time at R is computed as:

$$T_t(R) = T_t(L) + \Delta T_{\text{truck}}(L, R) \quad (10)$$

$$\Delta T_{\text{truck}}(L, R) = \int_L^R \frac{d\mathcal{P}}{v_t} + \sum_{k \in \mathcal{I}} \delta_t \quad (11)$$

where \mathcal{I} is the set of delivery nodes between L and R , and \mathcal{P} denotes the road path.

Discrete Path Formulation

For implementation, the truck path is discretized into segments:

$$\Delta T_{\text{truck}}(L, R) = \underbrace{(1 - \alpha) \frac{\text{dist}(t_i, t_{i+1})}{v_t}}_{\text{Remaining segment}} + \underbrace{\sum_{k=i+1}^{j-1} \left(\frac{\text{dist}(t_k, t_{k+1})}{v_t} + \delta_t \right)}_{\text{Full segments}} + \underbrace{\beta \frac{\text{dist}(t_j, t_{j+1})}{v_t}}_{\text{Final segment}} \quad (12)$$

where:

- L lies on segment (t_i, t_{i+1}) at fraction α
- R lies on segment (t_j, t_{j+1}) at fraction β

Synchronization Constraint

Substituting (9) and (12) into (7):

$$(1 - \alpha) \frac{\text{dist}(t_i, t_{i+1})}{v_t} + \sum_{k=i+1}^{j-1} \left(\frac{\text{dist}(t_k, t_{k+1})}{v_t} + \delta_t \right) + \beta \frac{\text{dist}(t_j, t_{j+1})}{v_t} = C + \frac{\|d_m - R(\beta)\|}{v_d} \quad (13)$$

where $R(\beta) = (1 - \beta)t_j + \beta t_{j+1}$.

Solution Approach

Solve (13) numerically for $\beta \in [0, 1]$:

1. For each candidate segment (t_j, t_{j+1}) :

$$A_j + \beta \frac{\text{dist}(t_j, t_{j+1})}{v_t} = C + \frac{\|d_m - R(\beta)\|}{v_d}$$

where A_j is the truck travel time to segment start.

2. Use root-finding methods to solve for β :

$$f(\beta) = A_j + \beta \frac{\text{dist}(t_j, t_{j+1})}{v_t} - C - \frac{\|d_m - R(\beta)\|}{v_d} = 0$$

3. If $\beta^* \in [0, 1]$ exists, set $R = R(\beta^*)$

Iterative Refinement

For cases requiring higher precision:

$$R^{(0)} = \text{Initial solution} \quad (14)$$

$$T_d^{(k)} = C + \frac{\|d_m - R^{(k)}\|}{v_d} \quad (15)$$

$$R^{(k+1)} = \text{Truck position at } \tau + T_d^{(k)} \quad (16)$$

Terminate when $\|R^{(k+1)} - R^{(k)}\| < \epsilon$.

Optimization Objective

Minimize waiting time:

$$\min W = \max (0, T_t(R) - (\tau + T_d))$$

with ideal solution $W = 0$ (perfect synchronization).

4.4.2 Implementation of Modified Heuristic Approach

Input Parameters

- **coords**: Coordinates of all nodes (customers + depot)
- **demands**: Package demand for each customer
- **depot**: Depot node index
- **num_trucks**: Number of trucks
- **drone_cap**: Maximum packages per drone tour
- **truck_cap**: Maximum packages per truck
- **road_matrix**: Truck travel times/distances between nodes
- **euclidean_matrix**: Drone flight distances (Euclidean)
- **drone_battery_range**: Maximum flight distance per drone tour

Key Heuristics

- **Zero-Wait Landing**:

$$T_{\text{drone}}(\text{launch} \rightarrow \text{customers} \rightarrow \text{landing}) = T_{\text{truck}}(\text{launch} \rightarrow \text{landing})$$

- **Drone Tour Assignment**: Prioritize high-demand, close customers
- **Perpendicular Intersection Fallback**: Place virtual landing node if synchronization fails

4.4.3 Implementation of K-Medoid and Heuristic Approach

Algorithm 2

1. Initialization

- Load parameters and validate:
 - Each truck carries 2 drones
 - Drone capacity and battery constraints
 - Truck capacity

2. Clustering with K-Medoids

- Cluster customers into `num_trucks` groups
- Choose medoids based on Euclidean distance to depot

3. Truck Assignment to Clusters

- Assign each truck to the nearest medoid
- Initialize route: `truck_route = [depot → medoid]`

4. Intra-Cluster Route Construction

(a) Truck Sub-Route Initialization:

- Solve TSP to get `truck_subroute`

(b) Drone-Truck Synchronization:

- Set launch point: `launch_node = truck_subroute[0]`
- Assign customers to drone tours using greedy method:
 - \sum demands \leq `drone_cap`
 - total flight distance \leq `drone_battery_range`

(c) Landing Point Selection (Zero-Wait Heuristic):

- Compute $T_{\text{truck}}(\text{launch} \rightarrow \text{landing})$ using `road_matrix`
- Compute $T_{\text{drone}} = d(\text{launch}, c_1) + \sum d(c_i, c_{i+1}) + d(c_k, \text{landing})$
- Find landing node such that:

$$T_{\text{drone}} = T_{\text{truck}}(\text{launch} \rightarrow \text{landing})$$

- If no match, reduce customers and retry

(d) Update Truck Route:

- Remove drone-served nodes from `truck_subroute`
- Extend final route

5. Battery & Capacity Constraints

- Split drone tours if battery range is violated
- Reject tours exceeding `drone_cap` or `truck_cap`

6. Return to Depot

- For each truck, append depot to end of route

7. Output

- **Truck Routes:** Sequences for each truck
- **Drone Routes:** Tuples (`launch`, [`customers`], `landing`)
- **Total Cost:**

$$\text{cost} = \sum \text{truck distances} + \sum \text{drone distances}$$

5 Result analysis

5.1 Input File Format

5.1.1 Modified Kmeans + ILP

VRP file containing

1. No of trucks to be allocated
2. Longitudes ,Latitudes
3. Truck max capacity
4. Demand per nodes

5.1.2 K-Medoid + Heuristic

1. VRP file containing
 - (a) No of trucks to be allocated
 - (b) X,Y coordinates
 - (c) Truck max capacity
 - (d) Demand per nodes
 - (e) Drone max capacity
2. Euclidian distance matrix
3. Road distance matrix
4. Road connection summary by edge coordinates

5.2 Cost comparison based on euclidean Distance

1. Input instances are taken from paper :
2. n denotes no of nodes and k denotes the no of trucks

Table 1: based on Instance: A-n45-k6.vrp

	DTRC	Modified Kmeans + ILP	Kmedoid + Heuristic
Total truck cost	1078.43	911.64	797.44
Total drone cost	200.96	268.30	250.60

Table 2: based on Instance : B-n34-k5.vrp

	DTRC	Modified Kmeans + ILP	Kmedoid + Heuristic
Total truck cost	860.23	646.30	798.19
Total drone cost	103.03	181.81	32.12

Table 3: based on Instance : P-n60-k10.vrp

	DTRC	Modified Kmeans + ILP	Kmedoid + Heuristic
Total truck cost	886.28	796.90	725.53
Total drone cost	258.69	318.78	317.43

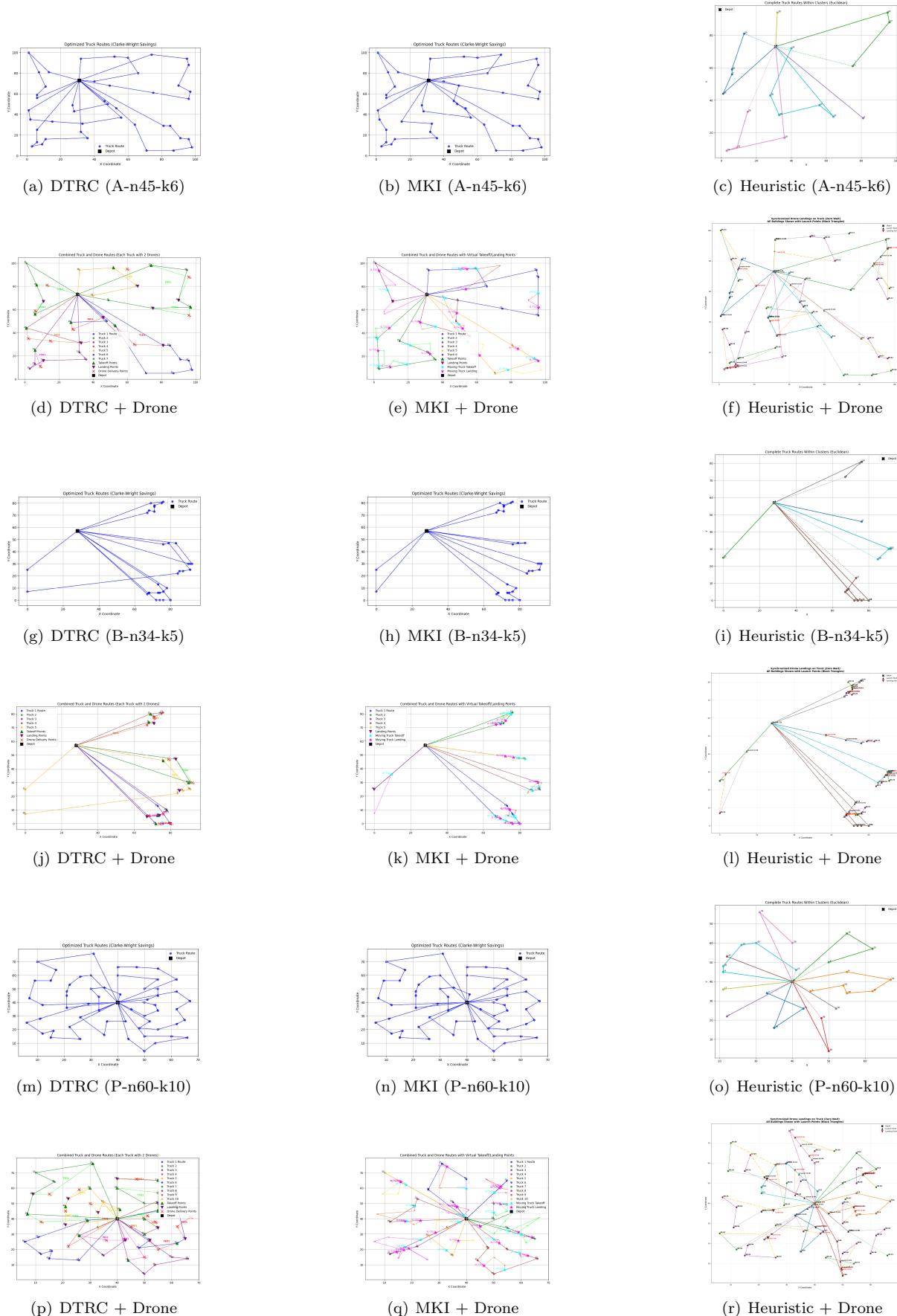


Figure 3: Visual Comparison of Routing Strategies

5.3 Cost comparison based in by-road distance

1. Input instances are generated from Sumo(Simulation of Urban Mobility) open software using real world areas
2. n denotes no of nodes : generated randomly
3. k denotes the no of trucks

5.3.1 for Instance ch-n30-k5.vrp

Modified Kmeans + ILP

Modified Truck Routes

- Truck 1 Route: [0, 6, 7, 31, 28, 0]
- Truck 2 Route: [0, 29, 27, 0]
- Truck 3 Route: [0, 3, 2, 8, 0]
- Truck 4 Route: [0, 25, 17, 14, 22, 0]
- Truck 5 Route: [0, 21, 19, 16, 12, 0]

Drone Routes

1. Truck 1 Drones:
 - V(22.572916,88.347056) → 15 → 11 → V(22.575800,88.346200)
 - V(22.585879,88.354830) → 26 → 10 → V(22.585377,88.357129)
2. Truck 2 Drones:
 - V(22.571628,88.395938) → 23 → 30 → V(22.560900,88.397100)
3. Truck 3 Drones:
 - V(22.583044,88.371845) → 5 → 20 → V(22.585869,88.374458)
 - V(22.584539,88.377318) → 13 → 1 → V(22.581522,88.375455)
 - V(22.577615,88.366821) → 4 → V(22.577974,88.367153)
4. Truck 4 Drones:
 - V(22.561125,88.351234) → 18 → V(22.561189,88.351268)
 - V(22.569650,88.352655) → 9 → 33 → V(22.566429,88.348467)
5. Truck 5 Drones:
 - V(22.597788,88.357021) → 34 → 32 → V(22.595730,88.349910)
 - V(22.595809,88.349908) → 24 → V(22.595780,88.349909)

Routing Costs

Truck	Drone Cost	Waiting Cost
Truck 1	5.3276	0
Truck 2	4.6932	0
Truck 3	3.6976	0
Truck 4	2.6441	0
Truck 5	8.1912	0
Total Drone Cost	38.5538	
Total Truck Cost	125.5170	

Table 4: Routing cost summary for trucks and drones

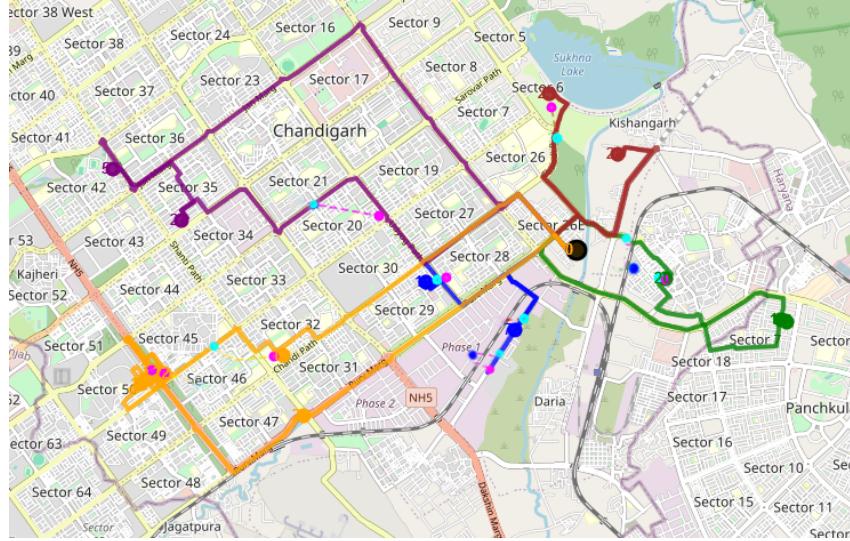


Figure 4: Chandigarh route map(Google API)

K-Medoid + Heuristic

Truck Routes per Cluster

Cluster	Route
0	Depot → 390018186 → 396656792 → 129584253 → 129584139 → Depot
1	Depot → 396001906 → 1190447658 → Depot
2	Depot → 396654025 → 392199678 → 392199702 → 396006695 → 396006666 → Depot
3	Depot → 876485689 → Depot
4	Depot → 391129544 → Depot

Drone Routes per Cluster

Cluster	Drone	Launch Point	Nodes Visited	Landing Point
0	0	(4913.91, 1786.00)	390018341 → 129584283	(5023.20, 1692.40)
0	1	(3922.59, 2063.43)	396657343 → 396656696	(5168.24, 1517.53)
1	0	(2864.51, 1893.44)	653560991	(3114.29, 1471.24)
1	1	(3308.26, 1687.74)	390180324	(2887.50, 1298.62)
2	0	(2562.95, 1354.59)	1282669882 → 388047887 → 396011452	(2573.20, 1361.73)
2	1	(3114.01, 863.50)	396122346 → 392199831	(2384.65, 1221.73)
3	0	(5153.25, 2987.63)	876485671 → 876482648 → 876500749 → 876500753	(4722.94, 2664.96)
4	0	(7225.73, 1333.24)	391129550	(7205.69, 1355.27)

Distance Summary (km)

Cluster	Truck	Drone 0	Drone 1	Drone Total	Cluster Total
0	10.19	3.11	3.46	6.57	16.76
1	10.90	2.29	2.72	5.01	15.91
2	10.45	2.47	2.84	5.31	15.76
3	8.70	3.98	—	3.98	12.68
4	9.54	0.13	—	0.13	9.67
Total Truck Distance					49.78
Total Drone Distance					21.00
Combined Distance					70.77

Time Summary

Cluster	Vehicle	Travel Time	Delivery Time	Total Time
0	Truck	10.19	12	22.19
	Drone 0	2.07	2	4.07
	Drone 1	2.31	2	4.31
1	Truck	10.90	6	16.90
	Drone 0	1.53	1	2.53
	Drone 1	1.81	1	2.81
2	Truck	10.45	15	25.45
	Drone 0	1.64	3	4.64
	Drone 1	1.90	2	3.90
3	Truck	8.70	3	11.70
	Drone 0	2.65	4	6.65
4	Truck	9.54	3	12.54
	Drone 0	0.08	1	1.08
Total Truck Time				88.78
Total Drone Time				30.00
Combined Operational Time				118.77

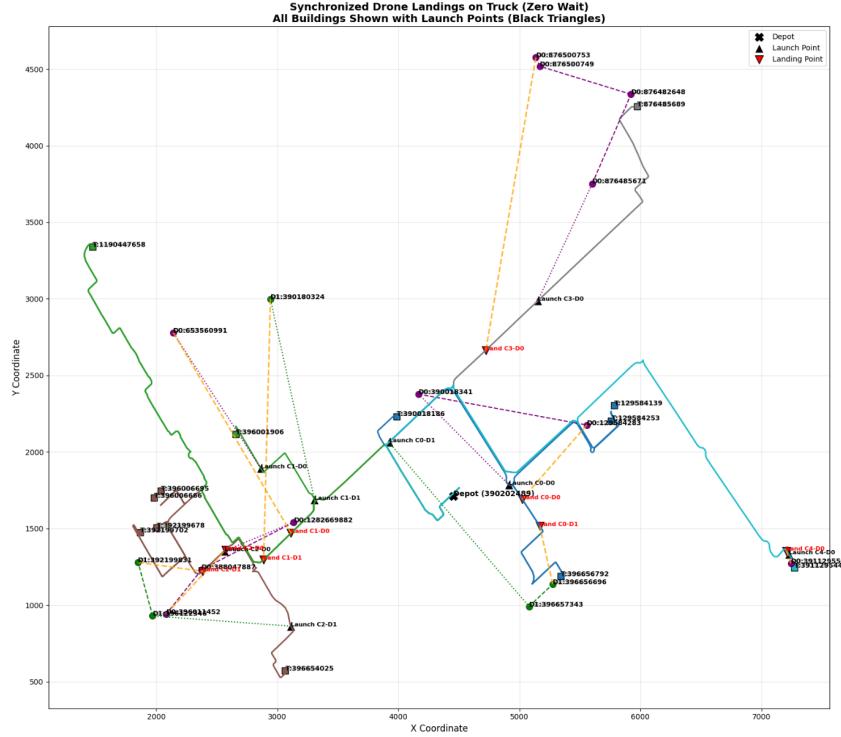


Figure 5: Chandigarh route map(SUMO)

5.3.2 for Instance kol-n35-k5.vrp

Modified Kmeans + ILP

Modified Truck Routes

- Truck 1 Route: [0, 18, 33, 0]
- Truck 2 Route: [0, 26, 10, 12, 16, 19, 28, 0]
- Truck 3 Route: [0, 27, 23, 0]
- Truck 4 Route: [0, 8, 2, 20, 0]
- Truck 5 Route: [0, 14, 7, 31, 11, 6, 0]

Drone Routes

1. Truck 1 Drones:
 - V(18) → 25 → 22 → V(33)
2. Truck 2 Drones:
 - V(26) → 32 → 24 → V(16)
 - V(16) → 21 → 34 → V(28)
3. Truck 3 Drones:
 - V(27) → 30 → V(27)
 - V(27) → 29 → V(23)
4. Truck 4 Drones:

- $V(8) \rightarrow 1 \rightarrow 13 \rightarrow V(2)$
- $V(2) \rightarrow 3 \rightarrow 5 \rightarrow V(20)$
- $V(20) \rightarrow 4 \rightarrow V(0)$

5. Truck 5 Drones:

- $V(14) \rightarrow 9 \rightarrow 15 \rightarrow V(11)$
- $V(11) \rightarrow 17 \rightarrow V(6)$

Routing Costs

Truck	Drone Cost	Waiting Cost
Truck 1	2.4525	0
Truck 2	5.1135	0
Truck 3	4.6324	0
Truck 4	7.6223	0
Truck 5	4.9778	0
Total Drone Cost	80.323	
Total Drone Cost	40.7986	

Table 5: Drone routing cost summary

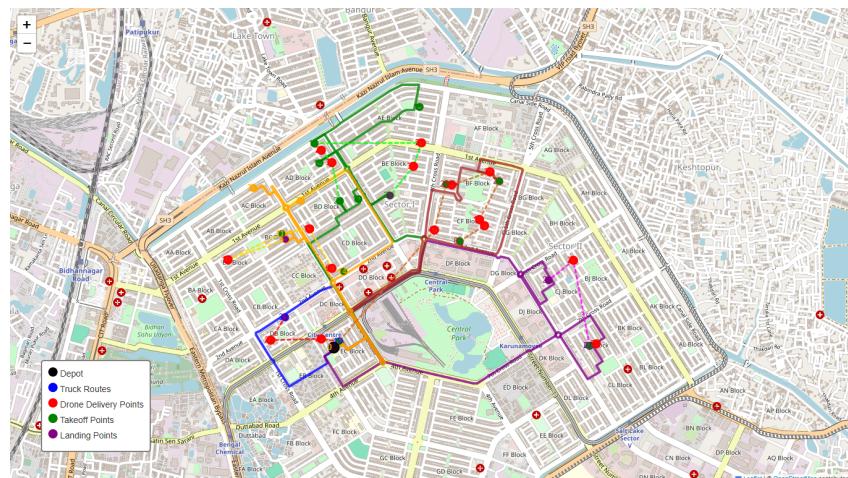


Figure 6: Kolkata route map(Google API)

K-Medoid + Heuristic

Truck Routes per Cluster

Cluster	Route
0	Depot → 358792596 → 358810384 → 235576527 → 235594144 → 235587813 → 239538820 → Depot
1	Depot → 358877576 → Depot
2	Depot → 236873619 → Depot
3	Depot → 358845412 → Depot
4	Depot → 236695867 → 236769904 → 236730723 → 235711701 → Depot

Drone Routes per Cluster

Cluster	Drone	Launch Point	Nodes Visited	Landing Point
0	0	(2409.80, 2486.03)	362741475 → 235599915 → 358811132	(2389.96, 2490.15)
0	1	(2510.77, 2466.34)	239378046 → 235599904 → 358810211	(2522.23, 2503.02)
1	0	(1515.54, 2314.04)	359505018 → 237738983 → 237741062	(1307.62, 2634.64)
1	1	(1568.29, 2168.99)	237741037	(1587.59, 2182.98)
2	0	(1500.79, 3057.32)	236954975 → 236871514	(1510.80, 3067.33)
2	1	(1537.74, 3027.37)	242346066 → 242991789	(1547.75, 3037.38)
3	0	(2615.36, 2429.08)	358786588 → 358845543	(2162.72, 2523.12)
3	1	(2834.09, 2382.78)	358845321	(2467.56, 2460.40)
4	0	(1387.66, 2529.80)	243040099 → 238132897	(1391.39, 2524.08)
4	1	(1599.14, 2306.15)	238132871 → 237092275	(1397.54, 2495.64)

Distance Summary

Cluster	Truck	Drone 0	Drone 1	Drone Total	Cluster Total
0	6.00	1.32	1.59	2.91	8.91
1	4.10	1.69	0.80	2.48	6.58
2	0.30	0.28	0.75	1.03	1.33
3	5.06	2.37	1.77	4.14	9.20
4	5.31	0.42	1.62	2.04	7.35
Total Truck Distance					20.77
Total Drone Distance					12.60
Combined Distance					33.37

Time Summary

Cluster	Vehicle	Travel Time	Delivery Time	Total Time
0	Truck	6.00	18	24.00
	Drone 0	0.88	3	3.88
	Drone 1	1.06	3	4.06
1	Truck	4.10	3	7.10
	Drone 0	1.13	3	4.13
	Drone 1	0.53	1	1.53
2	Truck	0.30	3	3.30
	Drone 0	0.19	2	2.19
	Drone 1	0.50	2	2.50
3	Truck	5.06	3	8.06
	Drone 0	1.58	2	3.58
	Drone 1	1.18	1	2.18
4	Truck	5.31	12	17.31
	Drone 0	0.28	2	2.28
	Drone 1	1.08	2	3.08
Total Truck Time				59.77
Total Drone Time				49.40
Combined Operational Time				109.17

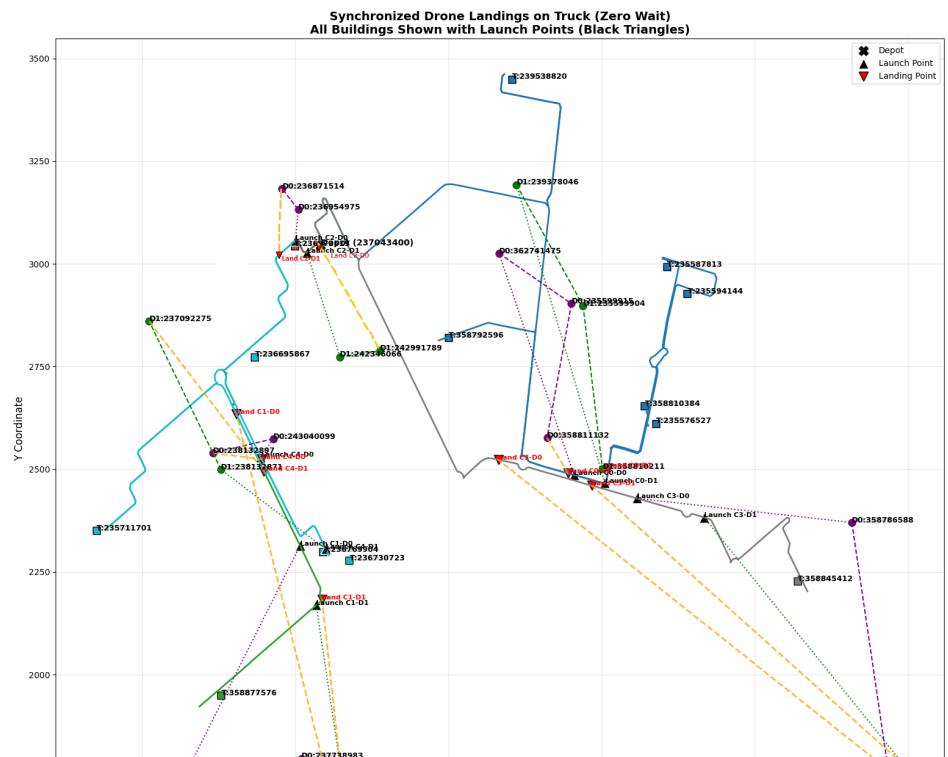


Figure 7: kolkata Route map(SUMO).

6 Conclusion & Future Scope

This study presents an effective solution to the Two-Echelon Vehicle Routing Problem with Drones (2EVRPD), optimizing last-mile logistics through coordinated truck-drone delivery. By combining modified K-Means clustering with Integer Linear Programming (ILP) and a K-Medoids-based heuristic approach, the proposed framework ensures efficient node clustering, route planning, and drone synchronization. The ILP model precisely determines optimal drone takeoff and landing points, while the heuristic approach offers scalability for large instances. Experimental results show an amount of reduction in total delivery time compared to traditional truck-only systems, particularly in multi-drone, multi-drop settings. The hybrid model not only improves delivery speed but also minimizes costs and supports dynamic operational environments. This work highlights the potential of truck-drone coordination in addressing the growing demands of modern e-commerce logistics and lays the groundwork for future exploration into adaptive strategies and real-time optimization under operational constraints.

Future research can enhance the current truck-drone coordination model in several impactful ways. First, the model can be extended to handle multiple demands originating from the same customer node, allowing for partial or repeated deliveries over multiple trips. Second, dynamic drone landing strategies can be incorporated, where drones are permitted to land on any feasible segment of a nearby truck's route, rather than returning strictly to their launching truck. This flexibility could further reduce idle time and increase system adaptability. Third, introducing delivery priority levels for customer nodes would enable time-sensitive parcels to be prioritized in routing decisions. This would require modifying the objective function to include penalty costs for delayed high-priority deliveries or enforcing time window constraints. Together, these enhancements would make the system more robust and applicable to complex, real-world logistics scenarios, especially in urban and heterogeneous delivery environments.

7 References

- A. Zhan, X. Li, J. Ma, Q. Meng. *Multi-objective optimization for drone-assisted delivery considering customer satisfaction and energy consumption*. Annals of Operations Research, 2023.
<https://link.springer.com/article/10.1007/s10479-023-05418-y>
- H. Xu, Y. Li, M. Du, H. Wang. *Optimizing Drone Delivery with Energy Consumption and Time Windows: A MILP Approach*. Drones, 8(1), 27, 2024.
<https://www.mdpi.com/2504-446X/8/1/27>
- R. D'Andrea. *Guest editorial can drones deliver?* Artificial Intelligence, Volume 287, 2020.
<https://www.sciencedirect.com/science/article/abs/pii/S0957417421002955>
- A. Allegretti, F. Dell'Amico, A. Faloppa, A. Hadjicharalambous. *A MILP formulation for the capacitated vehicle routing problem with drones*. Politecnico di Milano Repository (preprint).
https://re.public.polimi.it/bitstream/11311/1228778/4/Manuscript_iris.pdf
- T. Poikonen, S. Golden, B. P. Golden. *The vehicle routing problem with drones: Extended models and connections*. European Journal of Operational Research, Volume 274, Issue 2, 2019.
<https://www.sciencedirect.com/science/article/abs/pii/S0925527319300106>
- H. Y. Jeong, B. D. Song, S. Lee. *Truck-drone hybrid delivery routing: Payload-energy dependency and NoFly zones*. Transportation Research Part C: Emerging Technologies, Volume 102, 2019.
<https://doi.org/10.1016/j.trc.2019.02.022>
- P. Kitjacharoenchai, B.-C. Min, S. Lee. *Two echelon vehicle routing problem with drones in last mile delivery*. International Journal of Production Economics, Volume 225, 2020.
<https://doi.org/10.1016/j.ijpe.2019.107598>
- laporte2009 G. Laporte, “Fifty Years of Vehicle Routing,” *Transportation Science*, vol. 43, no. 4, pp. 408–416, 2009.
<https://doi.org/10.1287/trsc.1090.0301>