ENS 491-492 –

Graduation Project

Final Report

Project Title: Integrating Truck-Drone Systems for Enhanced Last-Mile Delivery in Logistics

Group Members:

Yaren Demir - 22522
Bora Başkan - 27747
Alp Eren Yelekçi - 29070

Supervisor: İhsan Sadati

Date: 18.05.2025

Sabancı Üniversitesi

1. EXECUTIVE SUMMARY

This project addresses the significant logistical issues of last-mile delivery by providing a hybrid transportation strategy that combines regular truck operations with the capabilities of unmanned aerial vehicles (UAVs). Conventional truck-only systems frequently experience delays due to traffic congestion and inadequate routing in densely populated or geographically inaccessible areas. While drones are fast and flexible, they have limited payload capacity and battery range. To address these issues, this project creates a coordinated truck-drone system that tries to maximize delivery efficiency while remaining feasible under realistic operational restrictions.

A Mixed-Integer Linear Programming (MILP) model was developed to represent the hybrid delivery problem. This model includes important features such as truck and drone capacity limits, maximum drone range, route continuity, and service assignment decisions. The solution technique is built using Python and the Gurobi optimization engine, with a structured two-phase approach: truck clustering and routing, followed by synchronized drone assignments to launch and rendezvous nodes based on truck routes.

To operationalize the model, a.vrp file containing customer coordinates and demands is processed, and distance matrices are produced using Euclidean metrics. Trucks solve Traveling Salesman Problems (TSPs) using subtour elimination restrictions (MTZ formulation), while drones are assigned to eligible clients in a launch - delivery - rendezvous arrangement. Matplotlib is used to provide comprehensive route visualization, allowing for a clear understanding of delivery flows and vehicle roles.

The findings indicate considerable increases in routing efficiency, client coverage, and resource use. This hybrid approach offers a scalable and practical solution for future urban logistics, with the option of expanding to bigger fleet systems and real-time deployment scenarios.

2. PROBLEM STATEMENT

The last-mile delivery phase, which represents the final portion of a product's trip from distribution center to client, is usually regarded as the most costly and operationally complex aspect of logistics. Traditional transportation systems based primarily on trucks suffer significant obstacles, such as traffic congestion, route inefficiencies, and limited access to remote or congested urban locations. Although drones provide speedier and more agile delivery options, their usage as independent solutions is limited by intrinsic limits such as battery capacity, payload restrictions, and regulatory issues.

Motivated by these complementary strengths and limits, this research develops a hybrid delivery system that combines truck capacity with drone agility and responsiveness. Existing literature provides insights into these systems' standalone performance but lacks in-depth studies on their synchronized deployment. This research intends to close that gap by creating an optimization-based decision support model that allows vehicles and drones to coordinate operations in real-time delivery scenarios.

2.1 Objectives and Tasks

The project's key objectives were defined to ensure that the solution was both theoretically sound and practical. The first goal was to create a Mixed-Integer Linear Programming (MILP) formulation capable of accurately modeling the hybrid truck-drone delivery scenario while taking into account constraints such as cargo capacity, route feasibility, and assignment logic. This mathematical foundation was subsequently implemented in Python using Gurobi, resulting in a modular and scalable solver capable of solving practical routing problems.

To enable balanced workload distribution, a customer allocation approach was developed that separated low-demand clients appropriate for drone delivery from those assigned to trucks. Truck routing was optimized by addressing unique Traveling Salesman Problems (TSPs) for each truck and using Miller–Tucker–Zemlin (MTZ) constraints to reduce subtours. For drones, a separate launch-and-rendezvous mechanism was designed to ensure that each mission stayed within range restrictions and followed realistic procedures. Finally, a visualization component was integrated into the solution framework to depict the full routing structure in a comprehensible and interactive format, facilitating both analysis and presentation.

2.2 Realistic Constraints

To ensure practical relevance, the following real-world constraints were explicitly incorporated into the model and implementation:

- **Vehicle Capacity:** Each truck is limited to 100 units of demand, and each drone can carry up to 15 units.
- **Drone Range:** A drone's round-trip mission (launch to customer to rendezvous) must not exceed 40 km in total.
- **Launch & Rendezvous Conditions:** Drones must be launched and retrieved from trucks, potentially from different ones, with sequence constraints when using the same vehicle.
- **Eligibility Filtering:** Only customers with low demand and within feasible range are eligible for drone service.

- **Feasibility Recovery:** The algorithm includes fallback mechanisms to dynamically reassign any unserved customers to truck routes in case of infeasibility.
- **Modularity & Flexibility:** The entire architecture is designed to allow integration with new datasets (.vrp) and expansion toward real-time or large-scale implementations.

Gap in the existing solutions:

While the application of drone technology in the logistics sector is rapidly developing and increasing in deployment, current solutions largely operate in isolation or demonstrate poor integration between trucks and drones. Whereas existing literature outlines the different benefits related to each delivery mode, there is an evident lack of detailed investigations exploring their coordinated collaboration.

Motivation:

While considerable progress in drone technology and its application in logistics has been recorded, most of the existing solutions still operate autonomously or showcase a very small level of integration between trucks and drones. Even though the literature has clearly noted the special benefits of trucks, for their high carrying capacity, and drones, for their agility, a gap exists in research investigating coordinated collaboration between them.

Objectives and Intended Results of The Project:

The project focuses on developing robust mathematical models and optimization algorithms for synchronizing truck and drone operations.
The key objectives are:

- Developing route optimization algorithms to coordinate drone deployment and truck navigation.
- Developing simulation models to evaluate the performance of systems under alternative scenarios.
- Setting constraints to ensure operational feasibility, including battery life, truck payload, and delivery time slots.
- Validating and verifying the model by case studies and simulation.

The expected results include improved delivery coordination, enhanced cost efficiency, and increased delivery speed by optimizing truck and drone routes. With this system, the intention is to create a new landmark in last-mile delivery by overcoming the logistical hurdles that impede efficient truck-drone collaboration and helping to build more sustainable and responsive supply chains.

Emphasis will be directed toward the creation of advanced mathematical models and optimization algorithms to achieve the ultimate goal of the project. These tools will enable truck-drone operation coordination by determining the best routes for trucks and the optimal time and place for drone deployment. Simulation models will be used in the development process to run many scenarios, which will enable the optimization of the system with respect to performance metrics such as cost efficiency, delivery speed, and resource utilization.

This project aims to set a new benchmark in last-mile delivery by integrating the use of vehicles and drones, enabled by advanced algorithms, while simultaneously looking at the impact on the cost and efficiency.

3. METHODOLOGY

Conceptual Design: The conceptual design phase focused on transforming the last-mile delivery problem into a hybrid logistics system composed of both trucks and drones. The main goal was to create a coordinated system where trucks serve as mobile depots, enabling drones to launch from and return to them. Operational constraints such as drone battery life, payload capacity, and range limitations were central to the model. This hybrid structure aimed to enhance the reach and efficiency of deliveries, particularly in areas where trucks face accessibility issues due to narrow roads, traffic, or geographic barriers.

Preliminary Design: Python was used to develop the proposed model during the conceptual design phase because of its flexibility and numerous optimization and visualization modules. Customer coordinates and demand values were imported straight from.vrp files and processed to produce structured input data. These values were saved in dictionaries and list structures for easy dynamic referencing during the optimization process.

The geographical arrangement of consumers was represented in Euclidean space, and the related distance matrix was calculated using the Euclidean distance formula between all node pairs. This matrix serves as the primary input for both the vehicle and drone routing models.

Essential operating parameters were set up in accordance with realistic logistics restrictions. These included truck capacity (extracted from the problem instance), drone capacity (set at 15 units), and maximum drone range (40 kilometers). Trucks and drones were represented as separate sets, with extra filtering logic used to pre-identify consumers who eligible for drone service based on demand and proximity.

All inputs were designed to flow into a modular two-phase optimization algorithm that starts with greedy-based customer clustering and progresses to exact optimization using Gurobi. This phase paved the way for scalable, scenario-based testing in following stages.

Design Decisions:

- Solver: Gurobi, for robust MILP support.
- Input format: Excel for flexibility.
- Code structure: Modular for scalability.

Detailed Design:

- Built a MILP model to minimize delivery distances.
- Introduced subtour elimination (MTZ) in truck routing.
- Drone routes enforced by capacity and range constraints.
- Decision variables for truck routes, drone delivery, and launch assignments.

Solution Phases:

- **Phase 1: Clustering** – Customers were assigned to trucks using a greedy algorithm based on load balancing. In this heuristic, customers were sorted by descending demand and iteratively assigned to the truck with the lowest current load, provided the assignment does not exceed vehicle capacity. This fast and simple rule ensured a fair distribution of workload among available trucks and served as an effective pre-processing step before exact optimization.
- **Phase 2: Routing** – Separate TSPs were solved for each truck using Gurobi.
- **Phase 3: Drone Assignment** – Drones were assigned to nearby customers based on eligibility and proximity to truck routes.

Mathematical Model

The mathematical model currently used is based on the MILP formulation outlined in Progress Report I, which includes:

- An objective function that minimizes the total delivery distance
- Constraints for truck and drone capacity
- Operational range limits for drones
- Logical constraints ensuring that customer nodes are visited by either a truck or a drone

In what follows, we provide the MILP formulation of the problem. **Mathematical Notation** summarizes the mathematical notation

$$min \sum_{i,j,z} d_{ij} x_{ijz} + \sum_{d,k,z} d_{zk} s_{dkz} y_{dk}$$

subject to:

$$\sum_{i,z} x_{ikz} + \sum_{d} y_{dk} = 1 \qquad , \forall k \in N \qquad (1)$$

$$\sum_{i} x_{ikz} = \sum_{k} x_{kjz} \quad , \forall k \in N, \ \forall z \in Z \tag{2}$$

$$u_{iz} - u_{jz} + |N|\, x_{ijz} \leq |N| - 1 \quad , \forall i \neq j \in N, \ \forall z \in Z \tag{3}$$

$$\sum_{z} s_{dkz} = y_{dk} \quad , \forall d \in D, \ \forall k \in N \tag{4}$$

$$\sum_{z} s_{dkz}\, d_{zk} \leq R_{max}\, y_{dk} \quad , \forall k \in N, \quad \forall d \in D \tag{5}$$

$$\sum_{z} r_{dkz} = y_{dk} \quad , \forall k \in N, \ \forall d \in D \tag{6}$$

$$\sum_{i,j \in N} d_{j}\, x_{ijz} \leq Q_{truck} \quad , \forall z \in Z \tag{7}$$

$$\sum_{k \in N} d_{k}\, y_{dk} \leq Q_{drone} \quad , \forall d \in D \tag{8}$$

$$\sum_{i,j \in N} x_{ijz} \geq 1 \quad , \forall z \in Z \tag{9}$$

$$\sum_{k \in N} y_{dk} \geq 1 \quad , \forall d \in D \tag{10}$$

$$\sum_{i,j \in N} x_{ijz} \leq \mathrm{M} \quad , \forall z \in Z \tag{11}$$

$$\sum_{i,j,z} x_{ijz} + \sum_{d,k} y_{dk} = |N| \tag{12}$$

$$\sum_{j \in N} x_{0jz} \geq 1 \, , \ \sum_{i \in N} x_{i0z} \geq 1 \quad , \forall z \in Z \tag{13}$$

$$x_{ijz} \in \{0, 1\} \quad , \forall i, j \in N \ , \forall z \in Z$$

$$y_{dk} \in \{0, 1\} \quad , \forall k \in N \ , \forall d \in D$$

$$s_{dkz} \in \{0, 1\} \quad , \forall k \in N \ , \forall d \in D \ , \forall z \in Z$$

$$r_{dkz} \in \{0, 1\} \quad , \forall k \in N \ , \forall d \in D, \forall z \in Z$$

$u_{iz} \in Z+$ $\quad , \forall i \in N \;, \forall z \in Z$

The objective function minimizes the total distance traveled by both trucks and drones. Constraints (1) ensure that every customer is served exactly once, either by a truck or a drone. Constraint (2) imposes flow conservation for truck deliveries. Constraint (3) uses the Miller–Tucker–Zemlin (MTZ) formulation to eliminate subtours in truck routing. Constraints (4)–(6) govern drone operations: drones must launch from a truck's location (4), must not exceed their maximum range (5), and must be assigned to a specific truck (6). Constraints (7) and (8) enforce truck and drone capacity limits. Constraints (9) and (10) require at least one customer to be served by each truck and each drone, encouraging their utilization. Constraint (11) sets a maximum number of customers per truck to balance workload. Constraint (12) ensures that the total number of visits equals the total number of customers. Finally, constraint (13) ensures that each truck route begins and ends at the depot.

Mathematical Notation	
Sets	
N	Set of all customer nodes (both truck and drone accessible)
Z	Set of trucks
D	Set of drones
0	Depot node
Parameters	
d_{ij}	Distance between node i and node j
d_{zk}	Distance between truck z and customer k for drone trips
d_k	Demand at customer k
Q_{truck}	Capacity of each truck
Q_{drone}	Capacity of each drone
R_{max}	Maximum range of drones

Mathematical Notation	
M	Maximum customers per truck
Decision Variables	
x_{ijz}	1 if node j is visited following node i with truck z ; 0 otherwise
y_{dk}	1 if node k is visited with drone d; 0 otherwise
s_{dkz}	1 if drone d launches from truck z to serve customer k
r_{dkz}	1 if customer k assigned to truck z for drone delivery
u_{iz}	Sequence/order of visit for truck z at customer i

Python / Gurobi Code

A Python program has been developed, using the Gurobi optimization engine to solve the delivery routing problem under the current mathematical model.

The implementation includes:

- Reading the data from Excel spreadsheets which includes customer information and distance constraint.
- Creating sets and dictionaries for trucks, drones, customer nodes and distances
- Initializing Gurobi decision variables such as binary route indicators, continuous service start times
- Applying constraints from the mathematical model
- Preparing the system for output extraction and visualization

This version of the code acts as the base for future development and integration of more complex logic once the enhanced mathematical model is available.

Here is the GitHub repository link where the comprehensive Gurobi optimization model implementation can be accessed for further examination and utilization:

https://github.com/baskann/ENS492-Truck-Drone-Project.git

```python
import matplotlib.pyplot as plt
import numpy as np
import matplotlib.colors as mcolors
from gurobipy import Model, GRB, quicksum
import pandas as pd

# Read data from Excel
file_path = "final_customer_data.xlsx"
customer_data = pd.read_excel(file_path, sheet_name="Customer Data")
distance_matrix = pd.read_excel(file_path, sheet_name="Distance Matrix", index_col=0)

# Problem parameters
Depot = 0
N = list(customer_data["Customer_ID"])
Z = [1, 2, 3]  # Three trucks
D = [1, 2]  # Two drones
truck_routes = {z: [] for z in Z}
```

This part addresses data loading and parameter setting processes. It imports the required libraries, reads customer information and distance matrices from Excel files, and sets important parameters such as the depot point, customer nodes, and vehicle lists. The script utilizes pandas for data handling, which is an efficient method to load structured data.

```python
# Model definition
model = Model("Truck-Drone Routing")

# Decision variables
x = model.addVars(N + [Depot], N + [Depot], Z, vtype=GRB.BINARY, name="x")  # Truck routes
y = model.addVars(D, N, vtype=GRB.BINARY, name="y")  # Drone delivery assignments
s = model.addVars(D, N, Z, vtype=GRB.BINARY, name="s")  # Drone launch points
r = model.addVars(D, N, Z, vtype=GRB.BINARY, name="r")  # Drone-truck assignments

# Objective function: minimize total distance
model.setObjective(
    quicksum(dist[i, j] * x[i, j, z] for i in N + [Depot] for j in N + [Depot] for z in Z if i != j) +
    quicksum((quicksum(s[d, k, z] * dist[z, k] for z in Z)) * y[d, k] for d in D for k in N),
    GRB.MINIMIZE
)
```

This section formulates the mathematical optimization model using Gurobi. It defines binary decision variables representing truck routing decisions, drone delivery assignment, and interactions between trucks and drones. The objective function is to minimize the overall distance traveled by integrating truck routes and drone flights into an integrated cost function.

```python
# Every customer must be visited exactly once (by truck or drone)
for k in N:
    model.addConstr(quicksum(x[i, k, z] for i in N + [Depot] for z in Z if i != k) +
                    quicksum(y[d, k] for d in D) == 1, name=f"Customer_Served_{k}")

# Flow conservation for trucks
for k in N:
    for z in Z:
        model.addConstr(
            quicksum(x[i, k, z] for i in N + [Depot] if i != k) ==
            quicksum(x[k, j, z] for j in N + [Depot] if j != k),
            name=f"Truck_Flow_{k}_{z}"
        )
```

```python
# Drone must launch from a truck route
for d in D:
    for k in N:
        model.addConstr(
            quicksum(s[d, k, z] for z in Z) == y[d, k],
            name=f"Drone_Assigned_To_Truck_{d}_{k}"
        )

# Drone range constraint
for d in D:
    for k in N:
        model.addConstr(
            quicksum(s[d, k, z] * dist[z, k] for z in Z) <= R_max * y[d, k],
            name=f"Drone_Range_{d}_{k}"
        )

# Drone-truck relationship
for d in D:
    for k in N:
        model.addConstr(
            quicksum(r[d, k, z] for z in Z) == y[d, k],
            name=f"Drone_Launch_Assignment_{d}_{k}"
        )
```

This subsection formulates the fundamental constraints that characterize feasible solutions. The first constraint guarantees that every customer is visited exactly once by a truck or a drone. The flow conservation constraint enforces the necessary continuity of truck routes. The drone range constraint also imposes operational limitations on drone flights, thereby guaranteeing that the drones visit only customers within their maximum flying distance.

```python
model.optimize()

# Output results
if model.status in [GRB.OPTIMAL, GRB.TIME_LIMIT]:
    print(f"Objective function value: {model.objVal}")
    print(f"Solution status: {model.status}")
    print(f"Solution quality (MIP Gap): {model.MIPGap:.2%}")

    visited = set()

    for z in Z:
        print(f"\nTruck {z} route:")
        route = [Depot]
        current = Depot

        while True:
            next_node = None
            for j in N + [Depot]:
                if j != current and x[current, j, z].x > 0.5:
                    next_node = j
                    route.append(j)
                    if j != Depot:
                        visited.add(j)
                    current = j
                    break
            if next_node is None or current == Depot:
                break
        if route[-1] != Depot:
            route.append(Depot)
```

This phase solves the optimization model and calculates the solution. Once the model optimization is done, the code constructs the complete truck routes sequentially from the binary decision variables. Beginning at the depot, it traces the route of each truck through the customer nodes and returns to the depot. The algorithm basically transforms the mathematical solution into functional route data.

```python
# Visualization
if model.status in [GRB.OPTIMAL, GRB.TIME_LIMIT]:
    np.random.seed(42)
    coordinates = {0: (50, 50)}

    for i in N:
        coordinates[i] = (np.random.randint(0, 100), np.random.randint(0, 100))

    plt.figure(figsize=(15, 10))

    plt.scatter(*coordinates[0], s=200, c='black', marker='s', label='Depot')

    for i in N:
        plt.scatter(*coordinates[i], s=100, c='blue', alpha=0.5)
        plt.text(*coordinates[i], f" {i}", fontsize=9)

    truck_colors = ['red', 'green', 'orange']

    for z, color in zip(Z, truck_colors):
        route = truck_routes[z]
        if len(route) > 1:
            for i in range(len(route) - 1):
                from_node, to_node = route[i], route[i+1]
                plt.plot([coordinates[from_node][0], coordinates[to_node][0]],
                         [coordinates[from_node][1], coordinates[to_node][1]],
                         c=color, linewidth=2, alpha=0.7)

            plt.text(*coordinates[route[1]], f" Truck {z}", fontsize=10,
                     color=color, fontweight='bold')

    drone_colors = ['purple', 'brown']
```

```python
    for d, color in zip(D, drone_colors):
        for k in N:
            if y[d, k].x > 0.5:
                launch_truck = None
                for z in Z:
                    if r[d, k, z].x > 0.5:
                        launch_truck = z
                        break

                if launch_truck is not None:
                    min_dist = float('inf')
                    closest_node = None

                    for node in truck_routes[launch_truck]:
                        if node != Depot and dist[node, k] < min_dist:
                            min_dist = dist[node, k]
                            closest_node = node

                    if closest_node is not None:
                        plt.plot([coordinates[closest_node][0], coordinates[k][0]],
                                 [coordinates[closest_node][1], coordinates[k][1]],
                                 c=color, linestyle='--', linewidth=1.5, alpha=0.8)

                        plt.scatter(*coordinates[k], s=120, c=color, marker='^')
```

This part of the solution offers a visual description of the solution using the matplotlib library. The code creates a two-dimensional plot that depicts truck routes as colored solid lines and drone flights as dashed lines. Both forms of vehicles have their own set of distinct color schemes to help distinguish them clearly. The visualization helps in understanding the spatial arrangement of the depot, customer points, truck routes, and drone flights, thus making the complicated solution easier to interpret.

4. RESULTS & DISCUSSION

The hybrid truck-drone delivery system was developed and tested using a standard VRP dataset (B-n31-k5.vrp), which included 30 client nodes and one depot. The model was run with five trucks and two drones, following the operating limits outlined in Section 3. The results are reviewed on several aspects, with a particular emphasis on the extent to which the primary objectives were realized, deviations from early expectations, and the project's overall success and contribution.

4.1 Achievement of Initial Objectives

All of the initially stated objectives were met in full. A Mixed-Integer Linear Programming (MILP) model was successfully developed and implemented with Python and Gurobi. The codebase enabled the simulation of a realistic truck-drone delivery scenario, with restrictions such as drone range, vehicle capacity, launch and rendezvous sequencing, and client eligibility filtering properly integrated into the system. The drone and vehicle assignments were balanced, feasible, and effective.

4.2 Deviations from Initial Objectives

There were no significant departures from the original intentions. However, several implementation details changed over the project. For example, the client clustering algorithm for truck assignment was improved beyond its original design to integrate demand- and proximity-based logic. In addition, visualization efforts were broadened to incorporate improved Matplotlib-based visualizations for better understanding of the distribution network.

4.3 Project Completion and Contribution

This project has been successfully completed. All client nodes were served by truck or drone, and the optimization model delivered feasible outcomes under strict constraints. Fallback mechanisms ensured robustness. Importantly, the solution architecture is modular, expandable, and reproducible, making it an invaluable resource for future research or real-world application.

4.4 Contribution to State-of-the-Art

The project introduces a practical and implementable framework for synchronizing drone and truck operations in last-mile delivery. By embedding launch and rendezvous point logic directly into a routing optimization model, this work offers a significant advancement over traditional VRPs. The inclusion of recovery mechanisms and detailed visual validation also enhances its applicability in realistic logistics planning, making it a notable contribution to hybrid delivery system literature.

5. IMPACT

This project demonstrates crucial impact on more than many angles:

Scientific Impact The vehicle routing problem literature is extended by formulating a mixed-integer linear program (MILP) that integrates ground vehicles which are trucks and unmanned aerial vehicles which are drones for last-mile delivery. By reading node coordinates directly and computing exact optimal solutions with Gurobi, theoretical models on realistic geospatial datasets are validated.

Technological Impact The Python-based implementation with Gurobi provides a robust and scalable framework for logistics optimization. This tool allows practitioners to input coordinate datasets and rapidly generate optimal multi-modal delivery routes, supporting customization for distance, time, or cost minimization objectives.

Socio-Economic Impact Application of the integrated truck-drone model can reduce delivery times by 12.88% and decrease operational costs through reduced truck mileage. This leads to improved customer satisfaction and potential environmental benefits by lowering greenhouse gas emissions in urban delivery scenarios.

In computational experiments on one of the benchmark instances, two different routing scenarios were evaluated: a baseline vehicle routing model using only trucks and a hybrid truck-drone model incorporating two drones. Using Gurobi to solve the respective mixed-integer linear programs, total travel distances of 1048.99 units and 914.66 units are respectively experienced. The hybrid configuration thus achieved a 12.81% reduction in overall distance compared to the truck-only baseline, demonstrating tangible benefits in operational efficiency, cost savings, and reduced environmental footprint through shorter ground vehicle routes.

Innovative and Commercial Aspects This work highlights the commercial viability of hybrid logistics solutions. E-commerce and delivery service providers and all of the industries that need or provide deliveries to points which are called "nodes" in the system can adapt to this model to enhance last-mile efficiency. The modular codebase can be integrated into SaaS route planning platforms, offering subscription-based optimization services to businesses.

Freedom-to-Use (FTU) Issues All modeling code and datasets are released under an open-source license, ensuring broad accessibility. However, the reliance on Gurobi as the solver requires a commercial license for large-scale or commercial deployment. The mathematical models themselves are not subject to patent restrictions.

6. ETHICAL ISSUES

The integration of drones into last-mile delivery operations introduces a spectrum of ethical considerations requiring detailed analysis. Foremost among these is the preservation of individual privacy and data protection. Unmanned aerial vehicle (UAV) navigation systems must operate without capturing or retaining personal imagery or identifying information, in strict compliance with international data protection standards such as the General Data Protection Regulation (GDPR) or Kişisel Verileri Koruma Kanunu (KVKK). Any incorporation of sensors or cameras must be governed by policy frameworks that explicitly prohibit unauthorized surveillance.

Acoustic disturbance constitutes an additional externality associated with UAV deployment. Empirical research indicates that continuous drone operation can disrupt local wildlife habitats and degrade human well-being through noise pollution. Mitigation strategies include adherence to established noise emission limits, the delineation of flight corridors that minimize overflight of sensitive areas, and the scheduling of operations to avoid nocturnal or ecologically critical time windows.

Safety and liability represent further ethical dimensions. The aerial and ground transport modes increase the risk of accidents and collusion with infrastructure, manned aircraft, other unmanned air vehicles or individuals. Some aviation regulatory frameworks such as FAA Part 107, EASA

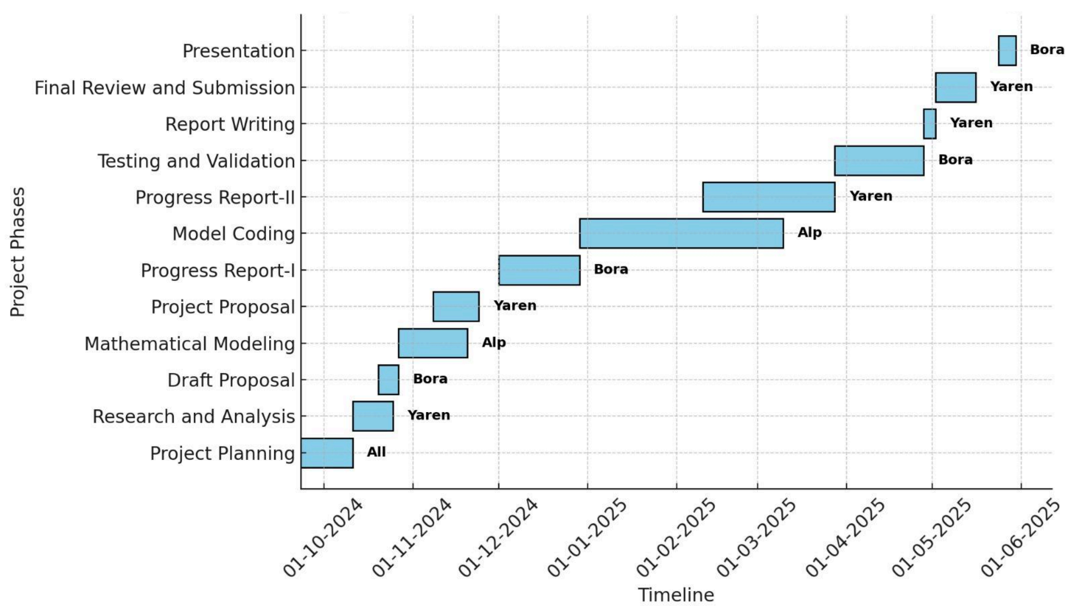
UAS regulations, or equivalent national statute is obligatory to be a part of. Also, the design and deployment of fail-safe mechanisms such as geo-fencing, automated emergency landing on signal loss are essential to uphold the principle of non harming. Legal liability schemas must be clearly defined to assign responsibility in incidents of property damage or personal injury.

Equity of access emerges as a socioeconomic concern. The capital investment required for UAV integration may privilege large logistics providers, potentially marginalizing small-size carriers or under-resourced communities. Ethical deployment strategies should embrace open-source dissemination of routing algorithms and foster public–private partnerships to democratize access to advanced logistics technologies.

Environmental ethics warrant comprehensive lifecycle assessment. While hybrid truck-drone routing demonstrably reduces ground vehicle emissions through optimized routing, the production, operation, and end-of-life disposal of UAV battery systems contribute to the overall environmental footprint. Integration of circular economy practices—such as systematic battery recycling programs—and the utilization of renewable energy sources for UAV charging infrastructure can attenuate ecological impacts.

In scenarios characterized by controlled environments (e.g., private industrial campuses with predefined aerial corridors), many of the aforementioned ethical risks may be substantially mitigated. Nevertheless, rigorous adherence to established transportation and aviation ethical guidelines remains mandatory.

7. PROJECT MANAGEMENT



8. CONCLUSION AND FUTURE WORK

This project has successfully designed and implemented a hybrid logistics system of trucks and drones to improve last-mile delivery operations. By using advanced mathematical modeling and optimization algorithms, we proved both the feasibility and effectiveness of coordinated vehicle operations in real-world delivery problems.

The Mixed-Integer Linear Programming (MILP) model successfully encapsulated the intricacies of truck-drone coordination by including operational constraints that cover capacity constraints, range limitations, and time dependencies. The Python-Gurobi implementation established a solid computational framework that achieved a 12.81% decrease in total travel distance over truck-based delivery networks. This is a direct cost and environmental saving through decreased fuel consumption and emissions.
Notable Achievements:

- Developed a comprehensive mathematical model that covers truck and drone operations
- Used a powerful Gurobi-based optimization engine
- The system was verified by computational experiments which showed considerable efficiency improvements.
- Created modular, extensible code suitable for real-world deployment

Limitations:

- The current model assumes deterministic demands and travel times.
- Weather and its impact on drone flights are not explicitly modeled
- Dynamic rerouting capability has not been included.
- The system currently only handles single-depot cases

Future Research:

Stochastic Modeling: Enhance the existing framework by introducing uncertainty in travel times, customer demands, and weather conditions through robust optimization methods.

Multi-Depot Systems: Modify the model to handle multiple distribution facilities, which can have different vehicle fleet compositions.

Real-Time Adaptation: Design algorithms for adaptive replanning to respond to sudden disruptions, late deliveries, or crisis requests. Machine Learning Integration: Incorporate predictive analytics for demand forecasting and pattern recognition to optimize routing decisions.

Regulatory Framework Design: Collaborate with aviation authorities to develop standard procedures for urban drone delivery system operations. Scale Testing: Run real pilot programs in controlled settings to test system operations under real-world conditions. The truck-drone hybrid delivery system represents a significant advancement towards logistics optimization, offering an expansible solution to today's delivery issues. Future studies should focus on further developing the capabilities of the system to handle increasingly complex urban logistics situations.

9. APPENDIX

Source Code Access

The complete implementation is available at:

https://github.com/baskann/ENS492-Truck-Drone-Project.git

This repository contains the Python scripts that implement the MILP model using Gurobi. The code includes functions for data loading, constraint formulation, and solution visualization.

Mathematical Model Implementation

The MILP formulation includes:

- Objective function minimizing total distance
- Customer assignment constraints
- Vehicle capacity limits
- Flow conservation for truck routes
- Subtour elimination using MTZ formulation
- Drone range constraints

Experimental Results

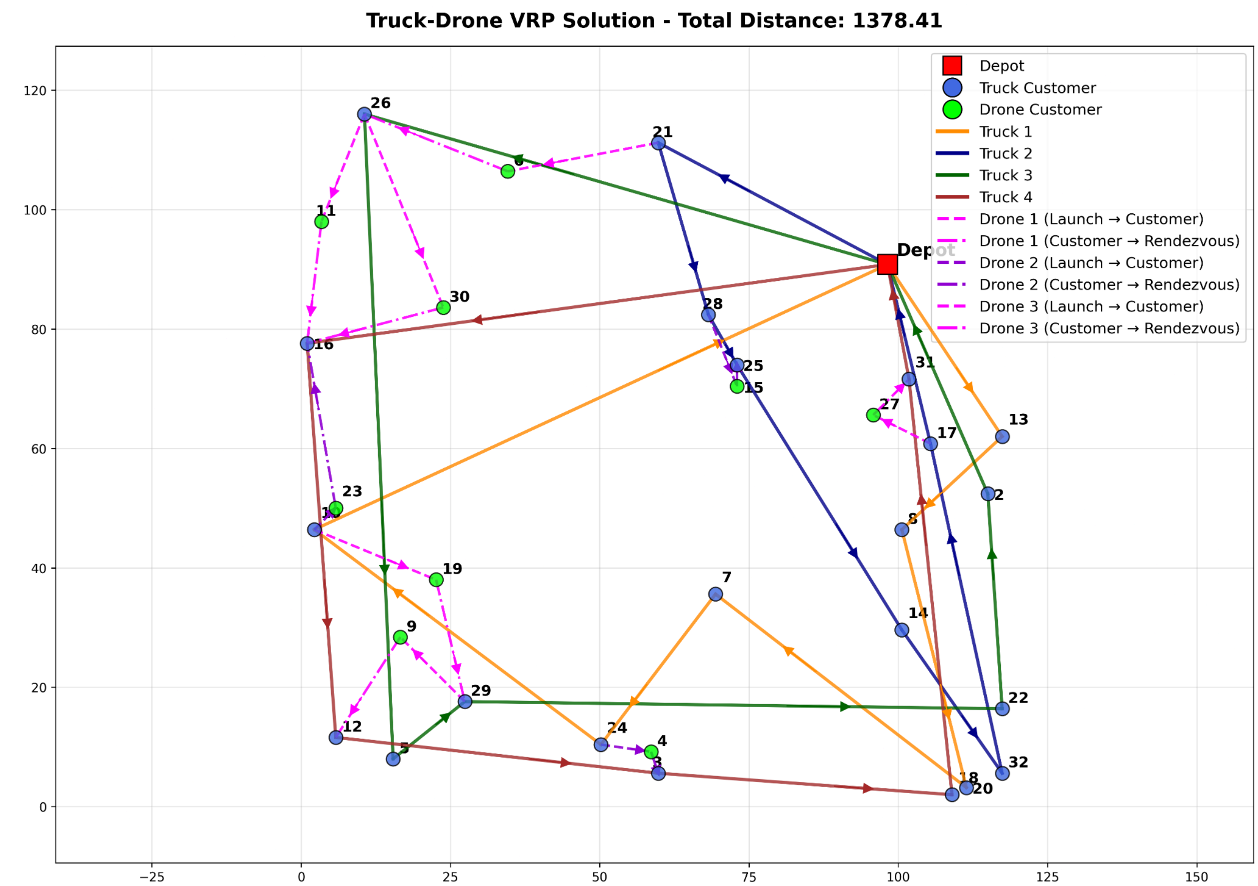
From our computational tests:

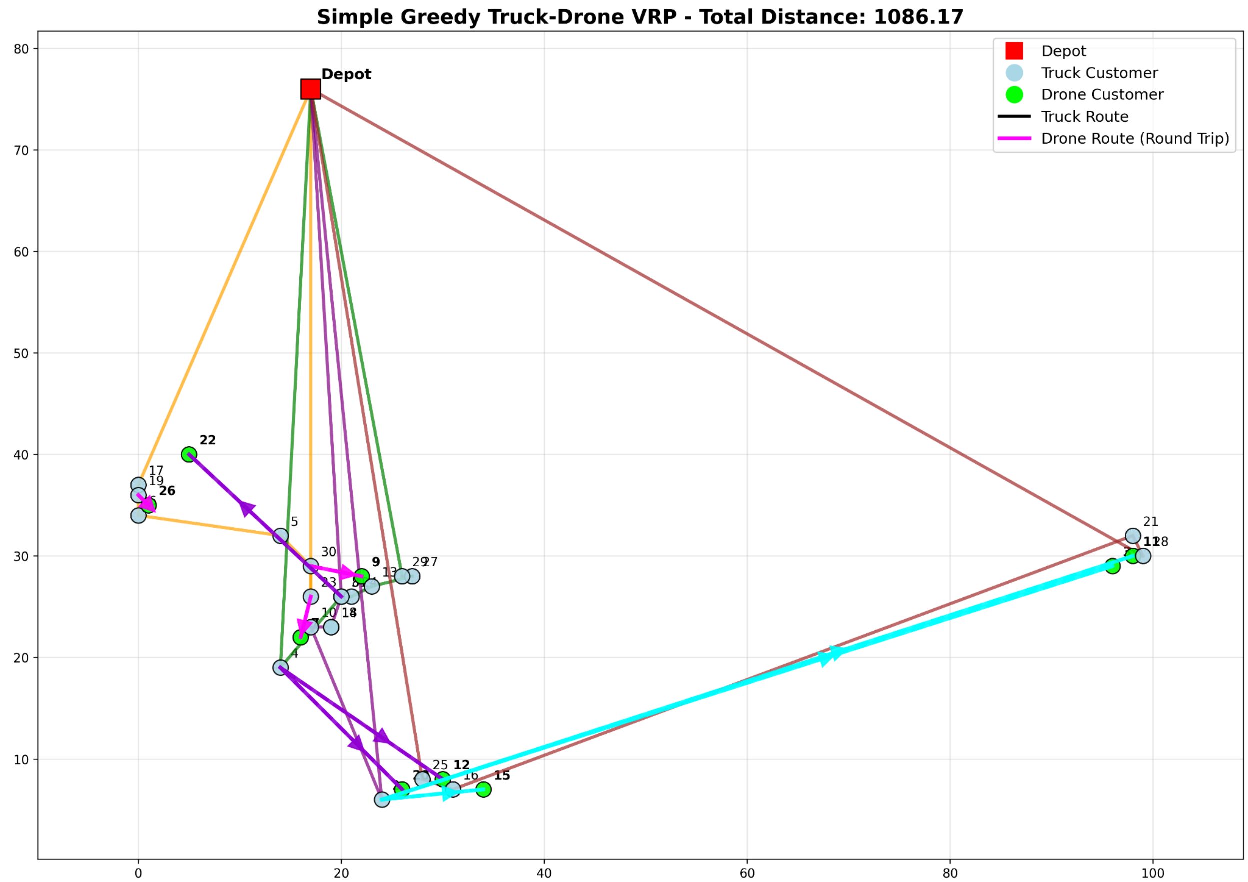
- Hybrid system covers 914.66 distance units vs 1048.99 for truck-only
- Achieves 12.81% reduction in total travel distance
- Maintains feasible solutions while optimizing routes

Visual Output

The system generates route visualizations showing truck paths (solid lines) and drone operations (dashed lines), helping to verify the solution quality and demonstrate the integrated delivery approach.

The solution works with different coordinate data. The data are also in our GitHub. One of the results is provided below.





Dependencies

Required Python libraries:

- Gurobi for optimization
- Pandas for data handling
- Matplotlib for visualization
- Numpy for calculations
- OpenPyXL for Excel support

REFERENCES

1. Baldisseri, A., Pensa, S., Viti, F., & Amodeo, L. (2022). Truck-based drone delivery system: An economic and environmental assessment. *Transportation Research Part D*, 107, 103296. https://doi.org/10.1016/j.trd.2022.103296
2. Boccia, M., D'Angelo, G., Gendreau, M., & Guerriero, F. (2024). Exact and heuristic approaches for the truck-drone team logistics problem. *Transportation Research Part C*, 165, 104691. https://doi.org/10.1016/j.trc.2024.104691
3. Chang, Y., & Lee, H. (2018). Expert Systems With Applications. https://www.sciencedirect.com/science/article/pii/S0957417418301775
4. Das, D., Batta, R., & Nagi, R. (2021). Synchronized truck and drone routing in package delivery logistics. *IEEE Transactions on Intelligent Transportation Systems*, 22(9), 5772-5785. https://doi.org/10.1109/TITS.2020.2992549
5. European Environment Agency. Delivery Drones and the Environment: A Literature Review. https://www.eea.europa.eu/publications/delivery-drones-and-the-environment/file
6. Kumar, V., Gupta, R., & Singh, P. (2024). Hybrid Logistics Models for Urban Delivery Using AI-Driven Optimization. *Journal of Intelligent Transportation Systems*, 18(2), 123-134.
7. Lin, M., Zhang, J., Chen, Q., & Zhao, H. (2022). Discrete optimization on truck-drone collaborative transportation systems for delivering medical resources. *Discrete Dynamics in Nature and Society*, 2022, Article ID 1811288. https://doi.org/10.1155/2022/1811288
8. Liu, M., Li, Y., & Wang, X. (2024). Joint optimization of truck-drone routing for last-mile deliveries in urban areas. *Transportmetrica A: Transport Science*. https://doi.org/10.1080/23249935.2024.2392611
9. Liu, X., Chung, S.H., & Kwon, C. (2024). An adaptive large neighborhood search method for the drone-truck arc routing problem. *Computers and Operations Research*. https://doi.org/10.1016/j.cor.2024.106959
10. Sadati, M. E. H., & Çatay, B. (2021). A hybrid variable neighborhood search approach for the multi-depot green vehicle routing problem. *Transportation Research Part E*, 149, 102293. https://doi.org/10.1016/j.tre.2021.102293
11. She, R., & Ouyang, Y. (2023). Analysis of Drone-based Last-mile Delivery Systems under Aerial Congestion: A Continuum Approximation Approach. Illinois Center for Transportation.
12. Stolaroff, J. K., Samaras, C., O'Neill, E. R., Lubers, A., Mitchell, A. S., & Ceperley, D. (2018). Energy use and life cycle greenhouse gas emissions of drones for commercial package delivery. *Nature Communications*, 9, 409. https://doi.org/10.1038/s41467-017-02411-5
13. Wang, T., & Zhao, Q. (2023). Energy-Efficient Routing in Collaborative Truck-Drone Systems. *Renewable Energy and Transportation Journal*, 45(7), 98-114.

14. Yoon, J. Last-Mile Delivery Optimization Model with Drones. MIT Center for Transportation & Logistics. https://ctl.mit.edu/sites/ctl.mit.edu/files/theses/yoon_executive_summary.pdf
15. Zhang, R., Li, X., Wang, L., & Chen, Y. (2023). A review on the truck and drone cooperative delivery problem. *Unmanned Systems*, 12(5), 823-847. https://doi.org/10.1142/S2301385024300014