OPTIMIZATION OF CUSTOMER LOCATION CLUSTERING AND DRONE BASED ROUTING FOR LAST MILE DELIVERY

Project Report

Submitted in partial fulfilment of the requirements for the award of the degree of

Bachelor of Technology
in
Mechanical Engineering

by

AMGOTH THARUN (*Roll No*.: **B201235ME)**
BANOTHU GANESH NAIK (*Roll No*.: **B201245ME)**
MOHD FAISAL AFTAB (*Roll No*.: **B201205ME)**
SAIREDDY SHREYAS (*Roll No*.: **B201211ME)**

तमसो मा ज्योतिर्गमय

Department of Mechanical Engineering

NATIONAL INSTITUTE OF TECHNOLOGY CALICUT

May 2024

CERTIFICATE

This is to certify that the report entitled "**OPTIMIZATION OF CUSTOMER LOCATION CLUSTERING AND DRONE BASED ROUTING FOR LAST MILE DELIVERY**" is a bonafide record of the Project done by **AMGOTH THARUN**(*Roll No*.: **B201235ME**), **BANOTHU GANESH NAIK** (*Roll No*.: **B201245ME**), **MOHD FAISAL AFTAB** (*Roll No*.: **B201205ME**) and **SAIREDDY SHREYAS** (*Roll No*.: **B201211ME)** under my supervision, in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Mechanical Engineering** from **National Institute of Technology Calicut**, and this work has not been submitted elsewhere for the award of a degree.

Dr. Ratna Kumar K (MED)
(guide)
Asst. Professor
Dept. of Mechanical Engineering

Dr. Sudarshan C (CSED)
(co-guide)
Asst. Professor
Dept. of Computer Science Eng

Professor & Head
Dept. of Mechanical Engineering

Place : NIT Calicut
Date : 07 May 2024

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to all those who contributed to the successful completion of the project titled "Optimization of Customer Location Clustering and Drone-Based Routing for Last Mile Delivery." This endeavor would not have been possible without the unwavering support and guidance from our guide Dr. Ratna Kumar K and Dr. Sudarshan C, whose expertise and valuable insights significantly enriched the project.

We extend our appreciation to the team members who collaborated diligently throughout the research, bringing diverse skills to the table. Their dedication and teamwork played a pivotal role in achieving the project's objectives.

Furthermore, We acknowledge the invaluable resources provided by NIT Calicut, which facilitated the execution of experiments and data analysis. The project was also enhanced by the cooperation and willingness of participants who generously shared their experiences and feedback.

In conclusion, this project has been a journey of learning and growth, and we are deeply thankful to everyone who contributed to its success. Their collective efforts have not only advanced our understanding of the subject but also made a meaningful contribution to the field of last-mile delivery optimization.

MOHD FAISAL AFTAB
SAIREDDY SHREYAS
AMGOTH THARUN
GANESH NAIK

ABSTRACT

This project addresses the critical challenges in last-mile delivery by proposing a comprehensive optimization framework for customer location clustering and drone-based routing. With the rapid growth of e-commerce and the emergence of q-commerce, the last mile delivery has become an important aspect of the supply chain as the need for it to become fast paced and cost effective has increased. Our research aims to streamline this process by using advanced algorithms and drone technology.

The project consists of two main components: customer location clustering and drone-based routing. The customer location clustering algorithm optimizes delivery routes by grouping geographically proximate customers, minimizing delivery time and costs. Concurrently, the drone-based routing system enhances efficiency by incorporating unmanned aerial vehicles into the delivery network, further reducing transit times and environmental impact.

Additionally, for the final step in our project strategically chosen locations within the NITC campus were analysed, where we established a hub to analyse the delivery of products within selected locations by hybrid truck and drone. This study provides practical insights into the implementation of our optimization framework in a real-world setting, showcasing its adaptability and effectiveness within a controlled environment.

Through several simulations using various algorithms, the proposed framework demonstrates significant improvements in delivery speed, cost-effectiveness, and environmental sustainability. The findings contribute to the ongoing discourse on optimizing last-mile delivery logistics, offering practical insights for industry stakeholders and policymakers. This project lays the foundation for future advancements in the field, fostering innovation and sustainability in the rapidly evolving landscape of last-mile delivery.

CONTENTS

LIST OF ABBREVIATIONS

DAPDP	Drone-Assisted Pickup and Delivery Problem
FSTSP	Flying Sidekick Traveling Salesman Problem
FTRPD	Fixed-Time Route Planning with Drones
HTGVNS	Hybrid Tabu Search and Greedy Variable Neighbourhood Search
mFSTSP	Multi-depot Flexible Service Task Scheduling Problem
MILP	Mixed Integer Linear Programming
MTRPD	Moving-Truck Route Planning with Drones
PDSTSP	Pollution-Routing Dynamic Stochastic Traveling Salesman Problem
PDSVRP	Pollution-Routing Dynamic Stochastic Vehicle Routing Problem
SLR	Systematic Literature Review
STRPD	Speed-Adaptive Route Planning with Drones
TDTL	Time-Dependent Traveling Salesman Problem with LIFO loading
UAV	Unmanned Aerial Vehicle

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION

In today's connected world, delivering goods to customers' doorsteps is the foundation of modern business. From e-commerce giants to local businesses, efficient transportation of products from distribution centres to consumers is critical to success. However, in the vast landscape of logistics, the last mile of delivery remains one of the most complex and difficult areas to optimize.

The "last mile" refers to the final stage of the delivery process, which includes transporting goods. hub to the customer's door. Despite the relatively short physical distance, the last mile presents significant logistical obstacles such as route optimization, traffic congestion, delivery schedule and customer accessibility. Addressing these challenges requires innovative solutions that simplify the delivery process while simultaneously reducing costs, improving efficiency and minimizing environmental impact.

The importance of last mile delivery is underscored by its significant share of total shipping and supply chain costs. Last mile delivery accounts for 53% of shipping costs and 41% of total supply chain costs, according to a Capgemini report. Despite the fact that more than 70% of consumers are price-sensitive and do not want to pay for fast delivery, expectations about the reliability and speed of delivery of goods are growing. This has led companies to compete fiercely to develop cost-effective and faster last-mile delivery operations.

Aware of the limitations of traditional methods, there is a growing interest in finding alternative solutions. last mile delivery. With relatively low operating costs and navigating uncongested flight routes, drones are emerging as a viable and attractive option for efficient last-mile delivery. In addition, drones offer an ecological mode of delivery, especially when charged with a low-carbon power source. The smaller packages associated with the rapid growth of Business-to-Consumer (B2C) transactions make drones particularly suitable for the delivery of most customer orders.

The application of drones in delivery logistics has increased due to initiatives from large companies such as Amazon, Google (X-Company) and DHL. In response to the growing demand for fast, efficient and cost-effective last-mile delivery, our project aims to address the road challenges faced by drones. Optimizing drone routing, we aim to improve last-mile delivery operations, providing a potential solution to one of the most important challenges in modern supply chain management.

Optimizing last mile delivery could not be more important. As consumer expectations evolve and e-commerce changes, companies are under increasing pressure to deliver goods faster, cheaper and easier. According to industry reports, the last mile can account for up to 53% of total logistics costs, making it an important area of savings opportunities.

Additionally, inefficient last-mile delivery processes increase traffic congestion, air pollution and carbon emissions, underscoring the urgent need for more sustainable delivery solutions. With this in mind, our project aims to address the critical challenges facing the last mile. delivery through a multifaceted approach.

Our project focuses on addressing these issues by offering an integrated optimization framework for last mile delivery. This framework consists of two main components: customer location clustering and hybrid routing. Using advanced algorithms and evolving drone technology, we aim to transform the world of last mile delivery and pave the way for more efficient, cost effective and sustainable logistics operations.

Firstly optimization framework involves grouping customers by location. This requires grouping customer locations based on factors such as geographic proximity and delivery frequency to optimize delivery routes. By intelligently grouping customers, we aim to minimize delivery times, reduce transport costs and improve overall efficiency.

Focusing on the optimization of drone routes in the context of a hybrid truck and drone transport system results in a compelling integration of traditional and innovative logistics solutions. In this hybrid model, trucks play a central role as mobile hubs that transport drones to strategically selected locations. This approach takes advantage of the strengths of both modes of transport and takes into account the limitations of each.

Trucks offer the ability to transport larger quantities of packages over long distances, while drones are great for navigating congested urban areas to provide efficient last-mile deliveries. This hybrid approach exploits the strengths of both drones and trucks and maximizes the efficiency and flexibility of last-mile deliveries.

In addition, the project recognizes the importance of addressing regulatory challenges related to airspace use and safety, and ensuring the seamless integration of drones into the hybrid delivery network. With this innovative approach, we aim to redefine last-mile logistics by providing a cost-effective and timely solution that combines the strengths of traditional and state-of-the-art delivery methods.

In addition to technological advances, our project considers the environmental and social impacts of optimizing last-mile deliveries. By reducing the number of delivery vehicles on the road and minimizing our fuel consumption optimization framework, it is possible to significantly reduce carbon dioxide emissions and environmental impact. In addition, by enabling faster and more efficient delivery options, we aim to improve customer accessibility and convenience, especially in underserved areas.

In summary, our project aims to address critical challenges in last mile delivery by offering innovative optimization. framework that combines customer location groups and hybrid truck routing. Through rigorous experimentation, analysis and real-world testing, we aim to demonstrate the feasibility and effectiveness of our framework, laying the foundation for future advances in last-mile delivery logistics. Finally, our project is an important step forward. to optimize last mile delivery operations. Using the latest technology and a deep understanding of logistics challenges, we want to transform the delivery process and make it faster, cheaper, and more environmentally friendly. With our optimization framework, we believe that meet customer expectations and efficiency and sustainability in final deliveries.

1.2 PROBLEM STATEMENT

In this problematic scenario, we are faced with a logistical challenge involving a total of N delivery points, M of which are exclusively for truck deliveries. Each truck is equipped with D drones, which complicates our optimization goal. The general goal is to strategically divide N transmission points into clusters so that each cluster can accommodate D or fewer points. It is important to understand that these clusters may contain points suitable for both truck and drone delivery, reflecting the integrated nature of our delivery system.

With the complexity of this hybrid delivery model where trucks play a central role, our next task is to name a central point of contact in each cluster. This hub serves as a hub for both drone pickup and drop off locations. The optimization challenge is to determine the most efficient route for the trucks and ensure that they pass through the selected connection points. This requires careful consideration of factors such as distance, transportation modes, and number of packages to maximize truck routes.

After optimizing truck routes, we focus on the complexity of drone delivery in each cluster. It is necessary to optimize drone routes from selected connection points, considering various factors such as delivery urgency, package size and real-time conditions. At the same time, determining the optimal number of drones per truck becomes a critical part of the overall efficiency of the system. Finding the right balance of drone-to-truck distribution ensures streamlined and seamless last-mile delivery that promotes both cost-efficiency and on-time service.

The complexity of our logistics challenge is not just about strategic service distribution and optimization. delivery points truck routes, as well as fine-tuning drone deliveries in each cluster. Balancing these elements and determining the optimal number of drones per truck is a multifaceted problem that requires a comprehensive and integrated approach to achieve the desired efficiency of last mile deliveries.

1.3 OUTLINE OF THE REPORT

The report starts off with chapter 1, the introduction to the topic of Last mile delivery and our Project's contribution in making it cost efficient, energy efficient and environment friendly in section 1.1. It explains the motivation behind the project as well as the need making last mile delivery more efficient. The introduction section also explains the basic idea behind the methodology and the expected outcomes of the project. In section 1.2, the problem statement has been explained in detail. It lays the base for the mathematical modelling of our Model. Section 1.3 is the outline of the report.

The next chapter, Review of Literature focuses on how SLR was performed and how the narrowing down of research journals was done. Section 2.1 gives the significance of models and the art of modelling to narrow down the journals. In Section 2.2, a detailed literature review was conducted and the objectives, achievements and the future scope of research are noted for all the shortlisted journals. Summary of the literature review is also given in the Chapter 2.

Chapter 3 gives a detailed description of the methodology of our project. Section 3.1 gives a detailed description of the mathematical modelling. It explains both, the hybrid model-1 and hybrid model-2. The working of the mathematical models has been shown visually in the sub-sections which give a good understanding of the working of the models. Section 3.2 gives a brief description about the algorithm which was used for clustering and the reason for using it.

Chapter 4 deals with the simulations performed by deploying the mathematical models using python codes and excel solver. Simulations have been performed on Hybrid Model-1 on random data. Then more realistic data has been obtained from a higher education institute's campus and the Hybrid Model-1 has been deployed over it and results have been obtained. Section 4.3 deals with the results obtained by performing simulations on Hybrid Model-2 on random data.

Chapter 5 concludes our project explaining the major goals achieved by the project and the future work possible on the domain. The base code used in the project has been given in the appendix and the references have been provided at the end.

CHAPTER 2
REVIEW OF LITERATURE

2.1 SIGNIFICANCE OF MODELS

A comprehensive systematic literature review (SLR) was conducted to delve deeper into the topic of our "Hybrid Truck and Drone Route Optimization" project. This involved a careful and structured process of systematic search, review, and synthesis of existing peer-reviewed literature. The goal was to provide an unbiased and comprehensive overview of the current state of knowledge on this topic. This systematic approach ensured that our review was methodical, rigorous and followed pre-defined criteria, enabling a comprehensive study of route optimization for hybrid truck and drone delivery systems.

Our chosen topic has significant relevance and relevance to the current last mile. delivery system in terms of improving efficiency. The available literature on this topic is extensive and covers many complex variations of the problem. Various aspects such as drone flight time, payload, battery capacity, number of customers, and the roles and quantities of both drones and trucks were thoroughly investigated. SLR sought to gain insights from this wealth of data, helping to better understand the challenges and potential solutions associated with hybrid transmission systems.

To ensure the timeliness of the findings, SLR focused on journals published between 2019 and 2023. This timeline was chosen to capture the latest advances and trends in the field. Carefully selected keywords, as shown in the accompanying diagram, guided our search strategy, allowing us to find relevant journals and gain valuable insights. This systematic literature review is the basis of our project and provides a solid overview of the existing scientific world and guides our next steps and method.

2.1.1 Art of Modelling

Our Scopus search utilizing keywords related to "The Hybrid Truck and Drone Route Optimization" initially yielded a substantial pool of 370 papers. To refine our focus, we proceeded to filter these papers based on their titles, resulting in a narrowed selection of 112 papers. Further refinement was achieved by assessing the abstracts of these papers, leading to a more selective pool of 27 papers. A meticulous examination of the content of these shortlisted papers allowed for a final filtration, resulting in a focused set of 11 papers. From the 11 papers we have reviewed 6 crucial papers in detail and focused on the shortcomings of these papers. This iterative process of narrowing down the literature through title, abstract, and content filters ensured that the papers selected for our systematic literature review are not only relevant to our topic but also contribute substantially to the current understanding of hybrid truck and drone route optimization. The final set of 6 papers will serve as a valuable foundation for our research, guiding our methodologies and insights in addressing the challenges associated with last-mile delivery systems.

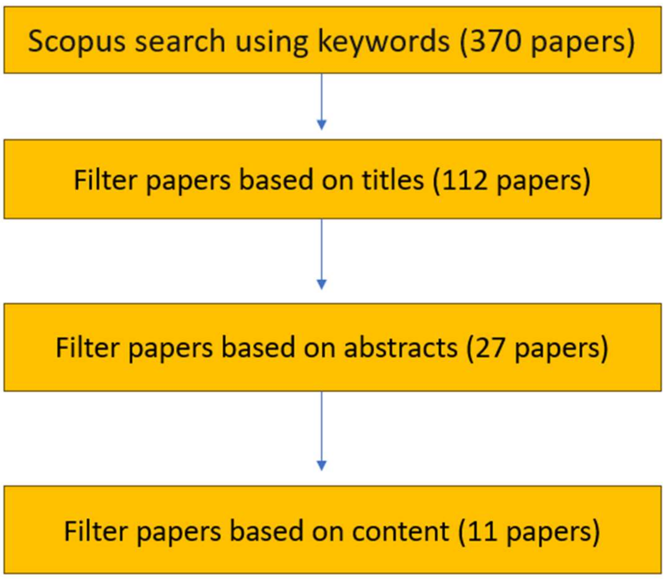

Fig 2.1 Filtering of research journals

Below are the Keywords we have chosen for performing the SLR.

Fig 2.2 Keywords used for filtering the Journals

Below are the percentage of research papers found through a keyword.

Fig 2.3 Percentage of Journals found through Keywords

The intersection of development degree (density) and relevance degree (centrality) at "route optimization scheduling package delivery" in our literature review graph suggests that this is a highly researched and crucial aspect within the broader domain of hybrid truck and drone route optimization. This convergence indicates that the topic has captured the attention of researchers and is considered central to advancements in the field. This aligns with our interests and objectives. Thus, pursuing this topic seems promising.

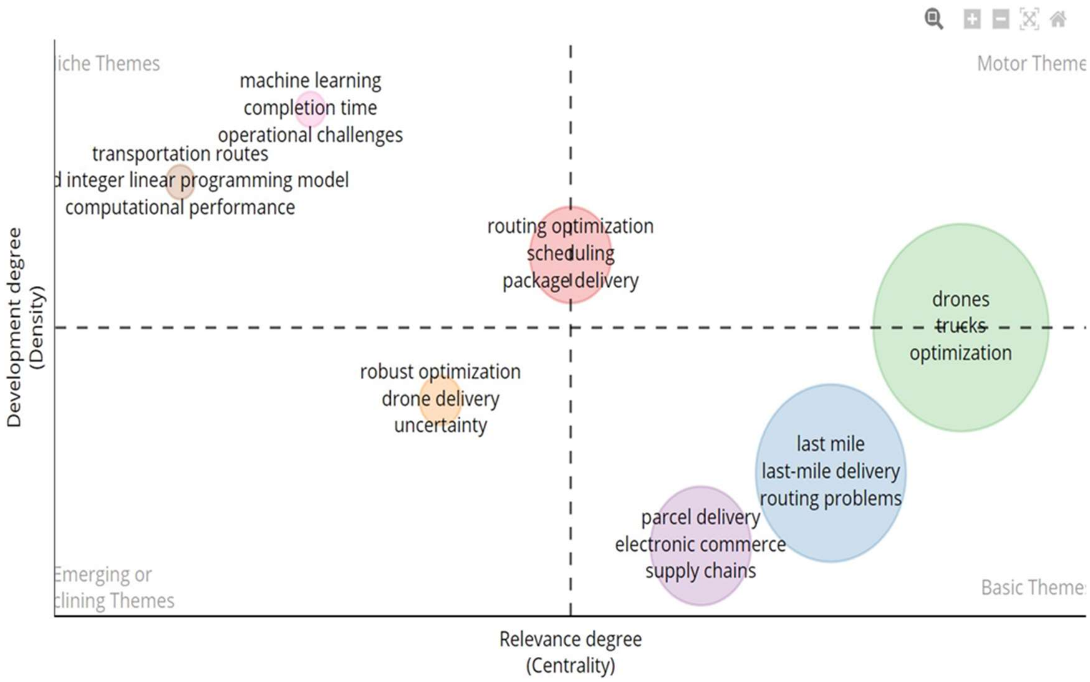

Fig 2.4 Development Degree vs Relevant Degree graph

pinpointed authors whose work holds considerable relevance and impact in this field. The bibliometric analysis not only facilitated the identification of influential scholars but also provided insights into the collaborative networks that contribute to the advancement of knowledge in hybrid delivery systems. These findings will guide our engagement with the existing literature and help establish connections with scholars who have made substantial contributions to the field, enhancing the depth and breadth of our research endeavors.

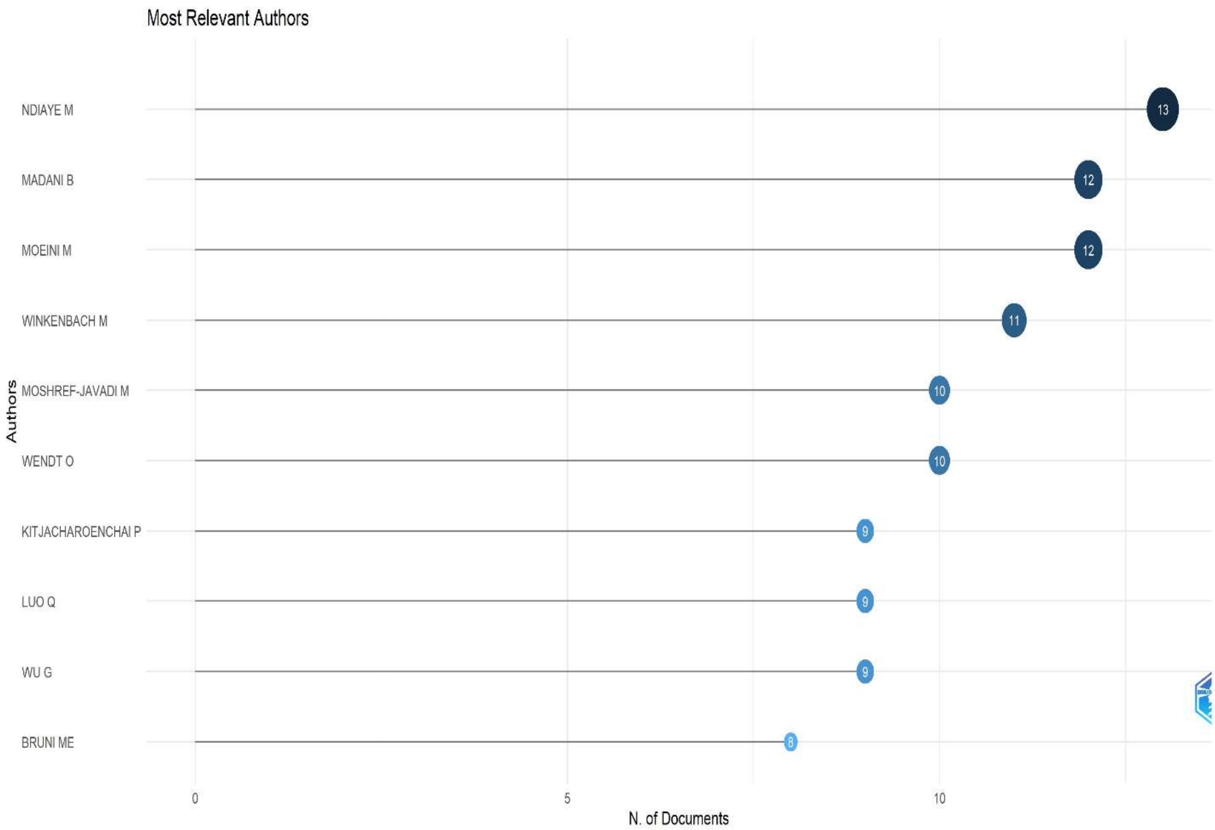

Fig 2.5 Most relevant authors

2.2 DETAILED LITERATURE REVIEW

Author	Contribution	Future Scope
Salamaa et al (2020)	The primary objective of paper is to optimize the total cost of delivery and total time of completion of delivery. It also provides an algorithm which can find the optimize solution in minimum time. It also tries to know the optimal number of drones in a truck for minimum cost of delivery.	The main gap in the paper is that it only considers one truck- multi drone instead of multi-truck and drone. The future scope is to improve the heuristic algorithm for getting fast solution and finding solution for multi-truck multi drone problem

Author	**Contribution**	**Future Scope**
Blanco et al, (2022)	Improving upon the Truck-Drone Traveling Salesman Problem (TDTL), simplifying the Multiple Flying Sidekick Traveling Salesman Problem (mFSTSP) Introducing an agent-based approach on a grid representation. Compare their model's performance against existing instances, achieving favourable results in larger cases.	Explore parallelization for enhanced computational speed, leveraging effective agent communication. Expand the algorithm's scope to address larger instances and accommodate multiple ground vehicles. Incorporate real-life constraints like vehicle variations, payload differences, and battery depletion models using parallel programming.
Mulumba et al, (2023)	Introduces a Mixed Integer Programming formulation, providing new optimal solutions and improved bounds for Fixed-Start Traveling Salesman Problem(FSTSP). Presents the Hybrid Tabu General Variable Neighborhood Search (HTGVNS) algorithm for larger instances, demonstrating its effectiveness through numerical analysis. Highlights the efficiency of a truck-drone delivery system, particularly with faster drones.	Allowance for the formulation to consider multiple UAVs per truck. Determination of the optimal number of UAVs to deploy per truck. Consider the dynamic case where requests for pickup and delivery are made in real-time.

Author	Contribution	Future Scope
Javadi et al, (2021)	Evaluate customer-centric delivery models (FTRPD, STRPD, and MTRPD) and their performance on various problem instances. Investigate the impact of key parameters like the number of UAVs, UAVs to truck speed ratio, and flight time limit on model effectiveness. Determine the model that outperforms others and assess the reduction in customer waiting times compared to a truck-only approach.	Increase synchronization in FTRPD by allowing UAVs to launch from the moving truck for potential performance improvements. Analyze and quantify the performance gain achieved by the enhanced synchronization level in the model. Explore the inclusion of additional real-world factors in these models, such as customer time windows, no-fly zones, and regulatory constraints, for more comprehensive and practical applications.
Nguyena et al, (2021)	Minimize total operational cost in the PDSTSP. Add constraints for parcel weight distribution among trucks and total vehicle working time. Study the PDSVRP, a derivative of the Vehicle Routing Problem, considering multiple trucks and drones.	Explore advanced algorithms and heuristics for improved problem-solving. Incorporate real-world constraints and dynamic adaptations into the model. Investigate multi-objective optimization, sustainability, and emerging technology integration for enhanced solutions.

Author	Contribution	Future Scope
Freitas et al, (2023)	The paper introduces DAPDP, extends MILP formulation, and proposes an efficient solution method. Performance analysis demonstrates cost savings from drones and trucks collaboration, including fuel and labour cost reductions.	Future research can explore extensions of DAPDP, such as allowing multiple UAVs per truck and optimizing their deployment. Investigating real-time dynamic scenarios for pickup and delivery requests presents another promising avenue for study.
Wang et al (2023)	The contributions of the research paper lie in the application of a rescheduling-based genetic algorithm for optimizing the delivery route planning problem for a fleet of heterogeneous drones. Paper aims to address the challenges of heterogeneous drones, which have different flight ranges, speeds, and payload capacities.	Incorporating dynamic constraints like battery levels, maintenance schedules, and varying flight regulations to make the algorithm more practical in real-world scenarios.
Poikonen et al (2020)	The contributions of the research paper lie in the application of a groundbreaking algorithm tailored to the multi-visit drone routing problem, revolutionizing the efficiency of route planning.	Exploring potential customer visitation orders and utilizing a different assignment mechanism within operations, such as integrating an integer programming formulation, hold promise in enhancing solution quality

Author	Contribution	Future Scope
Moadab et al (2021)	This research paper presents a model which integrates the existing public transportation network as a dynamic energy source for model tackles the multifaceted challenges of last-mile deliveries comprehensively. It not only optimizes drone routes, customer-location assignments, and package allocation but also offers a novel approach to leverage public transport for charging purposes	Scope for future research includes conducting pilot studies and collaboration with industry partners will bridge the gap between theory and application. Extending the model's adaptability to changing scenarios, such as fluctuating transportation schedules and evolving customer demands,

The literature review highlights several important contributions and future dimensions in the optimization of delivery logistics through the integration of trucks and drone systems. Various methods and algorithms have been proposed to address the challenges, such as minimizing total delivery cost and lead time, determining the optimal number of drones per truck, and optimizing route planning for multi-vehicles. One notable drawback is the focus primarily on single-truck and multi-drone configurations, ignoring the potential of multi-truck and drone configurations. Future research aims to close this gap by improving heuristic algorithms to find a faster solution and solve the multi-truck and multi-drone problem.

Several contributions stand out, including the development of new algorithms such as Hybrid Tabu General Variable Neighborhood Search (HTGVNS) for larger cases, which demonstrate efficiency in optimizing delivery systems, especially for faster drones. In addition, the introduction of customer-oriented delivery models such as FTRPD, STRPD and MTRPD offer promising opportunities to reduce customer waiting times compared to traditional truck deliveries.

Other contributions include the application of reorder-based genetic algorithms to heterogeneous drone fleets and the development of innovative algorithms for the multi-visited drone routing problem. These improvements aim to improve the efficiency of route planning and meet the challenges of drones with different functions. The literature also emphasizes the importance of dynamic adaptation in cluster formations, integration of new technologies such as artificial intelligence and IoT devices to improve route planning and address urban delivery challenges. These include congested traffic, restricted airspace and restricted access, which require specialized routing algorithms and delivery strategies adapted to urban environments.

Overall, the literature review highlights the significant potential of hybrid cargo aircraft delivery systems to address last mile delivery challenges. However, further research is needed to validate these findings in real-world scenarios and explore multi-stakeholder collaborative networks to ensure optimized resource allocation and route planning.

CHAPTER 3
METHODOLOGY

This chapter describes the approach used in this study, which uses a truck-drone hybrid model to optimize a drone delivery system. A thorough explanation model description, clustering technique and process is given in the parts that follows.

3.1 MODEL DESCRIPTION

The model proposed to solve the above problem is a variation of truck-drone hybrid model. In this model the delivery points are first divided into small clusters and a focal point is assigned to each cluster. Through these focal points the truck is passed with drones and delivery packages. Using this focal point as a launching place, the drone are sent to do the delivery of the packages to their respective locations. While doing this, the model can optimize the route travelled by truck and by doing clustering, the drone route is also optimised. The working of the model is described as below: -

- Clustering of the delivery points based on the number of drones and range of drones using KMeans algorithm
- Assigning focal points to each cluster based on whether a truck only point is there or not.
- Routing of the truck through all the focal points and then routing drone from focal point to delivery points

The model is further divided into two types: -

1.) Hybrid model with only drone-delivery points: - In this model there are no truck delivery points so all the focal points are centroid of the cluster. The routing of the truck is done through centroid of cluster.

2.) Hybrid model with both truck and drone delivery points; - In this model there are truck points. If a cluster contains truck-only point then that point is treated as focal point instead of centroid. So, the truck is routed through both centroid and truck-only delivery points.

Below is the mathematical model and the parameters are being defined according to [4]: - Let us consider a set where $l, l' \in L$ which is the set of depot and customer (or delivery) locations, $L = \{l_0, l_1, l_2, l_3, \ldots l_N\}$, where l_0 denotes the depot location. $k, k' \in \mathcal{K}$ set of possible clusters, $\mathrm{K} = \{k_o, k_1, k_2, k_3, \ldots k_{\widehat{K}}\}$, where k_o denotes the cluster with depot as its focal point Decision Variables and $\widehat{K}$ is the total number of clusters.

Parameters

C^D	travel cost of drone (in Rs/Km)
C^T	travel cost of truck (in Rs/Km)
(A_l, B_l)	coordinates of delivery location $l \in L$
F_l	maximum flight range of a drone (in Km) serving delivery location $l \in L$
g	number of drones mounted on the truck

Decision variables

(a_k, b_k)	coordinates of truck stop or cluster focal point $k \in K$
d^E_{lk}	Euclidean distance (in Km) between delivery location $l \in \mathcal{L}$ and cluster focal point $k \in \mathrm{K}$
$d^E_{kk'}$	Euclidean distance (in Km) between cluster focal points $k \in \mathrm{K}$ and $k' \in \mathrm{K}$
x_{lk}	1 if a delivery location $l \in \mathcal{L}$ is assigned to cluster $k \in \mathrm{K}$, 0 otherwise
q_{lk}	1 if a delivery location $l \in \mathcal{L}$ is assigned to cluster $k \in \mathrm{K}$ and served by a drone, 0 otherwise
$y_{kk'}$	1 if truck travels from cluster focal point $k \in \mathrm{K}$ to another focal point $k' \in \mathrm{K}$, 0 otherwise
u_l	Order in which cluster focal point $l \in L$ is visited by the truck
C	Total cost of the model

3.1.1 Hybrid Model 1 (No Truck- Only Delivery Points)

Then the mathematical equation of the above model can be given as following: -

$$\text{Minimize } C = C^D \sum_{l\in\mathcal{L}} \sum_{k\in\mathcal{K}} x_{lk} \times (2d_{lk}^E) + C^T \sum_{k\in\mathcal{K}} \sum_{k'\in\mathcal{K}} y_{kk'} d_{kk'}^E$$

$$\sum_{k\in\mathcal{K}} x_{lk} = 1 \ \forall l \in \mathcal{L} \tag{1}$$

$$d_{lk}^E = \sqrt{(A_l - a_k)^2 + (B_l - b_k)^2} \ \forall l \in \mathcal{L}, k \in \mathcal{K} \tag{2}$$

$$x_{lk} d_{lk}^E \leqslant F_l q_{lk} \ \forall l \in \mathcal{L}, k \in \mathcal{K} \tag{3}$$

$$\sum_{l\in\mathcal{L}} q_{lk} \leqslant g \ \forall k \in \mathcal{K} \tag{4}$$

$$\sum_{k'\in\mathcal{K}, k\neq k'} y_{kk'} = 1 \ \forall k \in \mathcal{K} \tag{5}$$

$$\sum_{k\in\mathcal{K}, k\neq k'} y_{kk'} = 1 \ \forall k' \in \mathcal{K} \tag{6}$$

$$u_k - u_{k'} + (\widehat{K} - 1) y_{kk'} + (\widehat{K} - 3) y_{k'k} \leqslant (\widehat{K} - 2) \ \forall k, k' \in \mathcal{K} \setminus \{k_o\}, k \neq k' \tag{7}$$

$$(a_k, b_k) = \left(A_{l_o}, B_{l_o}\right) \forall k \in \mathcal{K} \ni k = \{k_o\} \tag{8}$$

$$d_{lk}^E = \sqrt{(a_k - a_{k'})^2 + (b_k - b_{k'})^2} \ \ \forall l \in \mathcal{L}, k \in \mathcal{K} \tag{9}$$

$$x_{lk} \in \{0,1\} \ \forall l \in \mathcal{L}, k \in \mathcal{K} \tag{10}$$

$$y_{kk'} \in \{0,1\} \ \forall k, k' \in \mathcal{K} \tag{11}$$

The (1) constraint ensures that every delivery location $l \in \mathcal{L}$ is assigned to one and only one cluster focal point $k \in \mathcal{K}$. (2) determines the drone travel distance between a delivery location $l \in \mathcal{L}$ and its focal point $k \in \mathcal{K}$. (3) ensures that this flight distance to be within the maximum drone flight range for location $l \in \mathcal{L}$. (4) ensures that the number of locations in each cluster cannot exceed the total drones on the truck. (5) and (6) ensures each cluster focal point $k \in \mathcal{K}$ has exactly one inbound and one outbound visit by the truck. (7) is classic constraint for subtour elimination for the routing of the truck. (8) assigns the cluster focal point $\{k_o\} \in \mathcal{K}$ to the depot with coordinates $\left(A_{l_o}, B_{l_o}\right)$ to ensure a truck visit to the depot, in addition to allowing delivery locations to be assigned to the depot like any other cluster focal point. (9) computes the Euclidian travel distance for a truck between two cluster focal points k and k'. (10) and (11) are the binary restrictions on the decision variables are specified using the above constraints.

The images below show how model works: -

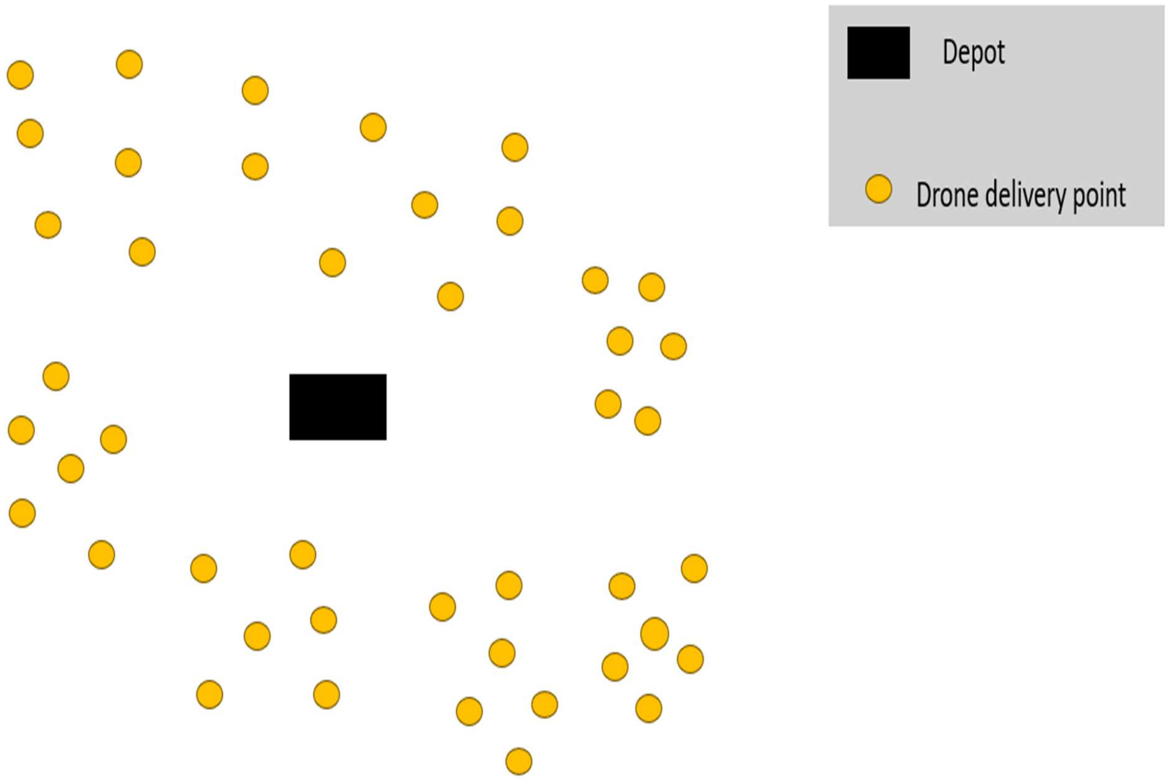

Fig 3.1 Scattered delivery points

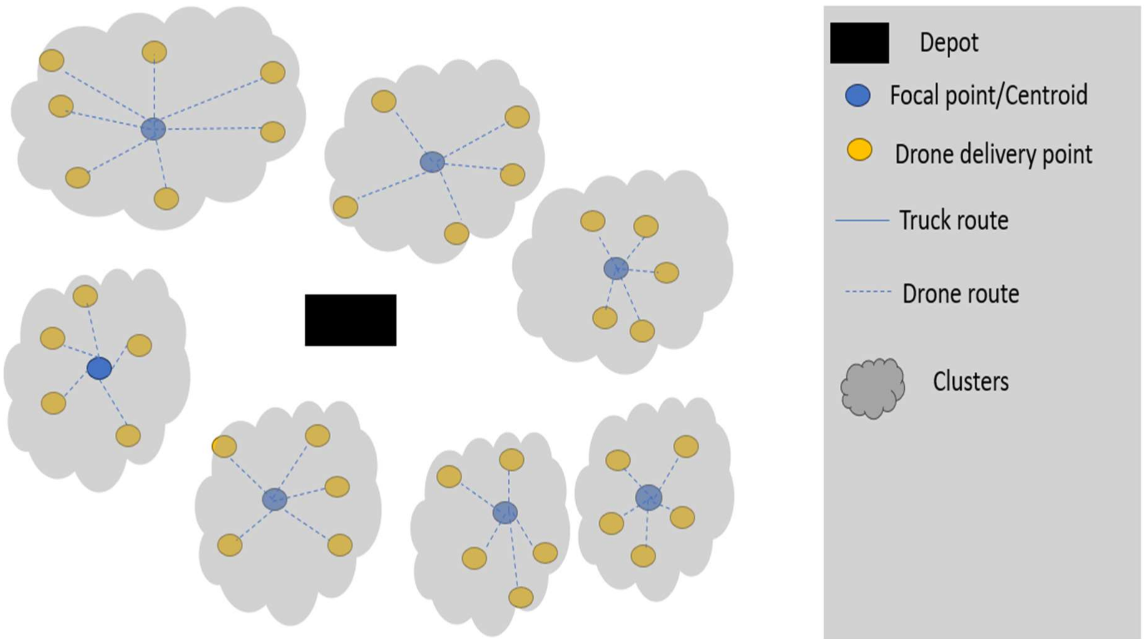

Fig 3.2 Clustering of the points

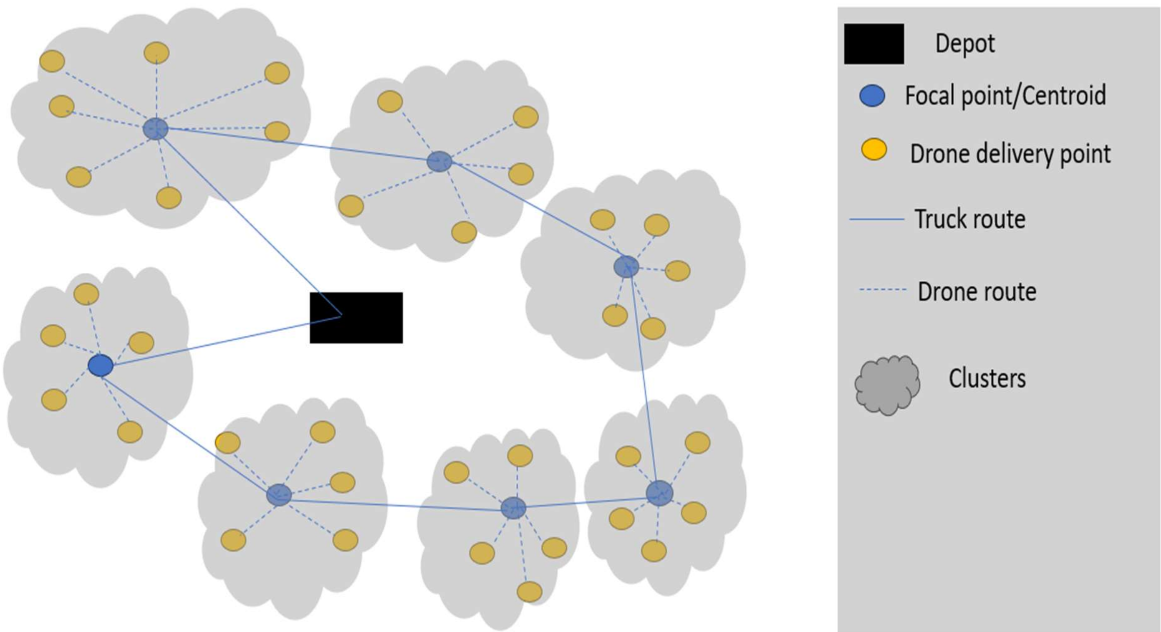

Fig 3.3 Final routing

3.1.2 Hybrid Model 2 (With Truck-Only Delivery Points)

The mathematical model is given below: -

$$\text{Minimize C} = C^D \sum_{l' \in \mathcal{S}} \sum_{l \in \mathcal{S}} x_{l^l} \times (2D_{l^V}) + C^T \sum_{l' \in \mathcal{S}} \sum_{l \in \mathcal{P}} y_{l'l} D^R_{l_l}$$

$\sum_{l \in \mathcal{L}} x_{ll} \leqslant \widehat{K}$ (1)

$x_{l'} \leqslant x_{ll} \; \forall l', l \in \mathcal{L}$ (2)

$\sum_{l \in \mathcal{L}} x_{l'l} = 1 \; \forall l' \in \mathcal{L}$ (3)

$x_{ll} = 1 \; \forall l \in \mathcal{L} \ni l = \{l_0\}$ (4)

$\sum_{l' \in \mathcal{L}} x_{l'l} \leqslant g + 1 \; \forall l \in \mathcal{L}$ (5)

$g \leqslant G$ (6)

$x_{l'l} D^E_{l'l} \leqslant F_{l'} \; \forall l', l \in \mathcal{L}$ (7)

$\sum_{l' \in \mathcal{L}, l' \neq l} y_{l^l} = x_{ll} \; \forall l \in \mathcal{L}$ (8)

$\sum_{l \in \mathcal{L}, l \neq l'} y_{l'l} = x_{l^{l'}} \; \forall l' \in \mathcal{L}$ (9)

$u_l - u_{l'} + (\widehat{K} - 1) y_{ll'} + (\widehat{K} - 3) y_{l'l} \leqslant (\widehat{K} - 2) \; \forall l, l' \in \mathcal{L} \setminus \{l_o\}, l' \neq l$ (10)

$x_{l'l}, y_{l'l} \in \{0,1\} \; \forall l', l \in \mathcal{L}$ (11)

The objective function minimizes the total cost of operating the truck and set of drones. Constraint (1) restricts the total truck stops to be capped by the maximum allowable clusters. Constraint (2) allows a delivery location to be assigned to another location only if the latter serves as a focal point. Further, constraint (3) ensures that every delivery location is assigned to exactly one cluster focal point location. Constraint (4) forces the depot to be a focal point so that it can dispatch drones to nearby locations. constraint (5) ensures that the number of drone-supplied locations in each cluster cannot exceed the total drones carried by the truck. The truck has a capacity restriction of G drones on its roof, and this condition is guaranteed using constraint (6). Constraint (7) stipulates that a location served by a drone is assigned to a cluster focal point only if the distance between them is within the flight range. Constraints (8) and (9) specify the truck route by confining its stops to cluster focal points and limiting the number of truck visits to each focal point to one. In addition, constraint (10) eliminates sub-tours to ensure a single trip of the truck to visit all focal points before returning to the depot. Finally, the binary restrictions are specified by constraint (11)

The images below show how the model works.

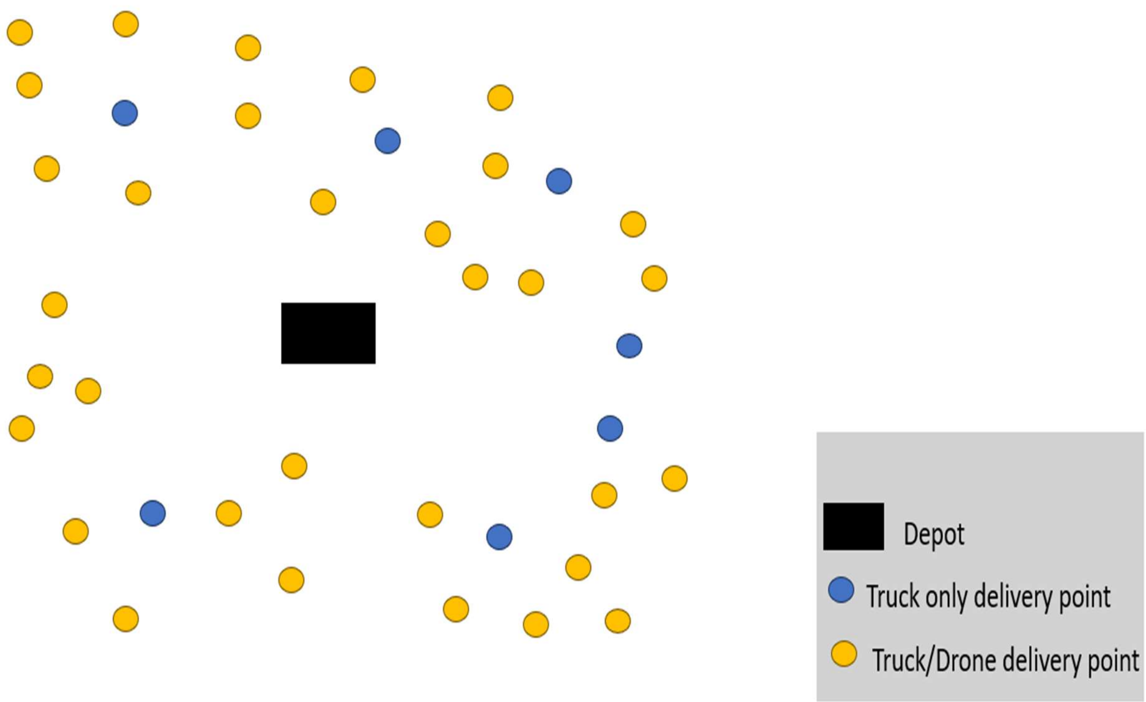

Fig 3.4 The delivery points and the depot of a last mile delivery order

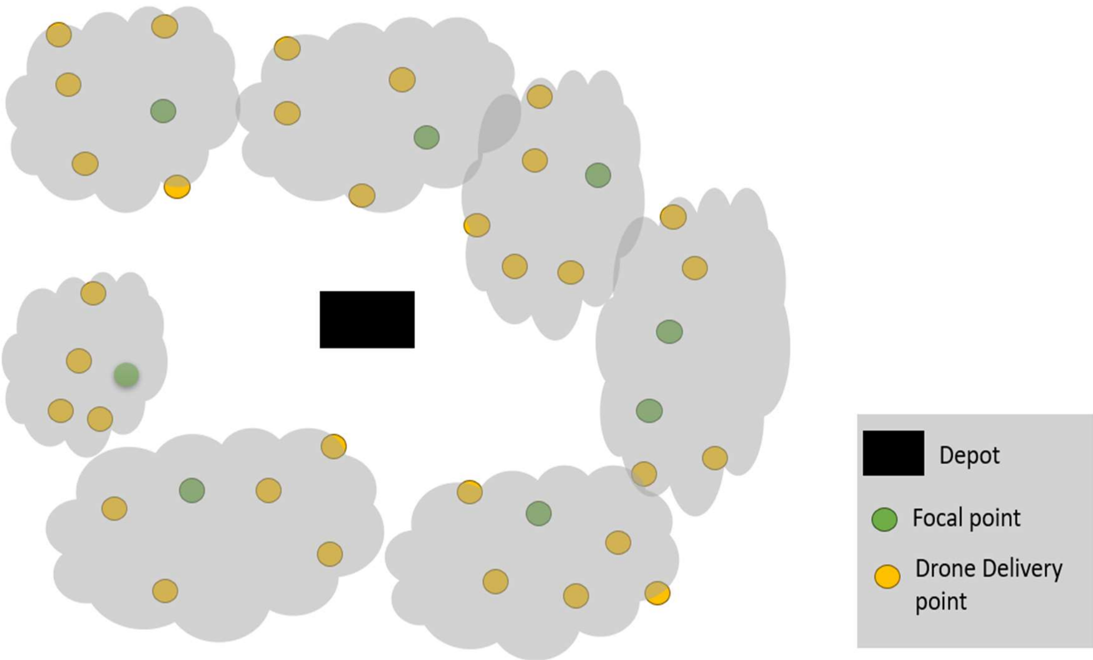

Fig 3.5 Clustering of the delivery points

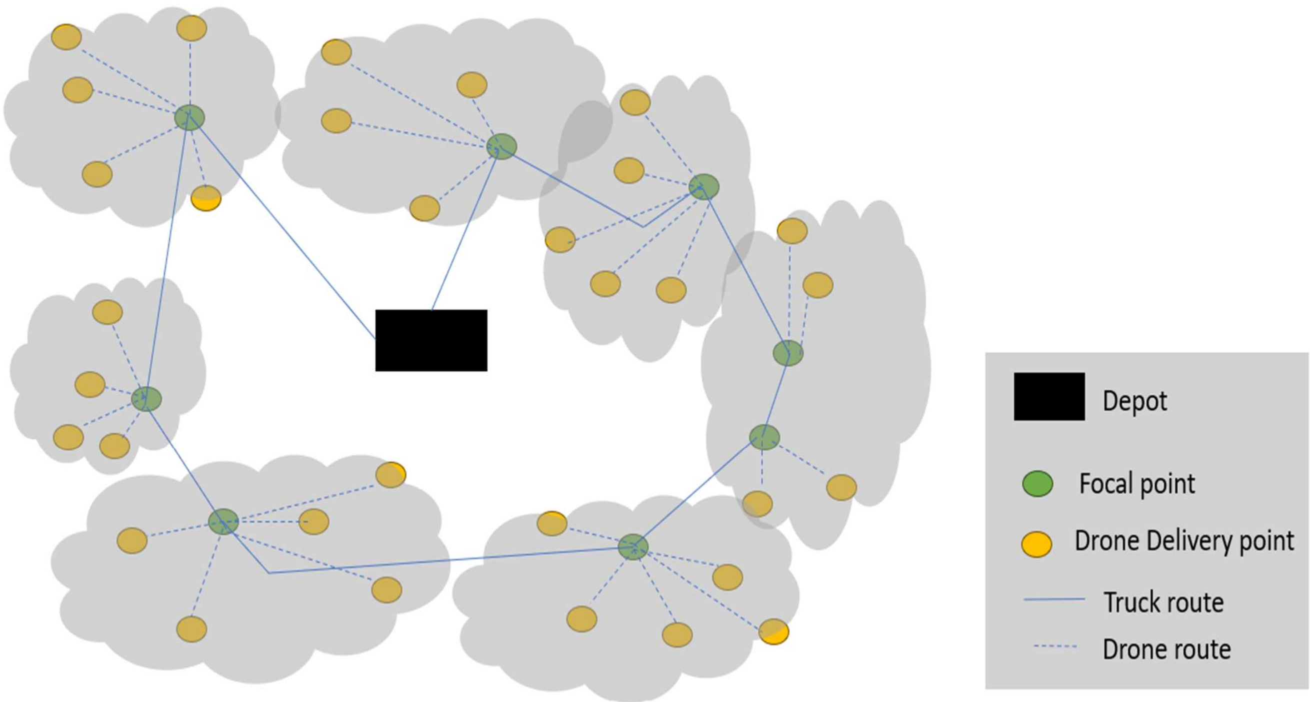

Fig 3.6 Final route of tuck and drone

3.2 CLUSTERING ALGORITHM

The model is highly depended on cluster formation for final routing to be optimised. To get the best cluster, in this study the KMeans algorithm is used. The advantage of using KMeans algorithm is the way it forms clusters. As the clustering is done according to distances in the Kmeans algorithm, the cluster formed will always be an optimised cluster with minimum distances between points which are close to each other. This give the model a nearly optimised solution for drone routing from focal point as the cluster formation has already made the distance travelled by drone the minimum.

K-Means clustering is an unsupervised learning algorithm that aims to group observations in a given dataset into clusters. Unlike supervised learning, there is no labelled data for this clustering. The algorithm divides objects into clusters that share similarities and are dissimilar to objects belonging to other clusters. The term 'K' represents the desired number of clusters. To perform K-Means clustering, we first specify the value of K, and then the algorithm assigns each observation to exactly one of the K clusters. The goal is to minimize the sum of distances between data points and their respective cluster centroids.

The working of KMeans algorithm is as below: -

- **Initialization**: Randomly select K initial cluster centroids (often based on data points).
- **Assignment**: Assign each data point to the nearest centroid (usually using Euclidean distance).
- **Update Centroids**: Recalculate the centroids based on the mean of the data points assigned to each cluster.
- **Repeat Assignment and Update**: Iteratively reassign data points and update centroids until convergence (when centroids no longer change significantly).

In this study, the variant of Kmeans algorithm is used which allow us to put constrained on the number of points and the distance between points in cluster. This helps in taking account of all the constrained that is available in the model. The code of the clustering algorithm along with the whole model is given in appendix 1.

CHAPTER 4
RESULTS AND DISCUSSION

Both models are tested on random data set generated by a program. Hybrid model 1 is also tested on a real 30 location data point in NITC campus.

4.1 HYBRID MODEL 1 RESULTS

Comparative analysis is conducted between Hybrid Model-1 and truck only model, utilizing 50 random locations in a 100 km sq area via Excel solver. In tuck-only model, a truck alone was employed, treating it as a TSP to deliver packages. Conversely, in Hybrid Model 1, a hybrid truck-drone approach was adopted. Here, a clustering algorithm is used to group the 50 delivery points into clusters and determined centroids for each cluster. The truck traversed through these centroids, stopping at each, while drones completed package deliveries within the cluster. Costs were set at Rs 50 per km for the truck and Rs 6 for drones. For truck-only model, the cost totalled Rs 2992.5, whereas for Hybrid Model-1, it amounted to Rs 1568.5 for the truck and Rs 641.2 for drones, totalling Rs 2209.7. Consequently, Hybrid Model-1 yielded a profit of Rs 782.8 over truck-only model.

To assess the impact of varying drone numbers on delivery costs compared to conventional truck delivery, simulations are conducted with 25 delivery points, adjusting drone numbers from 4 to 8 while keeping other parameters constant. The results are tabulated below.

Table 4.1 Effect of number of drones on cost

Numbers of delivery point	Numbers of drones	Cost of using only truck	Cost of hybrid model	Change in cost	% Change in cost
25	4	2046	1712	334	16.32%
25	5	2046	1711	335	16.37%
25	6	2046	1689	357	17.45%
25	7	2046	1637	409	19.99%
25	8	2046	1665	381	18.62%

From the above result it is clear that Hybrid model-1 is more cost effective. The effect of number of drones can be clearly seen. The table shows that there is an optimum number of drones for the specified number of delivery point.

4.2 CAMPUS SIMULATION RESULTS ON HIGHER EDUCATION INSTITUTE

For applying the Hybrid Model 1 to NITC for more practical results, we have picked 30 locations from NITC strategically. The locations picked are

Table 4.2 Locations Chosen

Location	Latitude	Longitude
Main Gate	11.31982	75.932818
NLHC	11.321539	75.933159
ELHC	11.322396	75.933841
DB	11.321999	75.93491
A HOSTEL	11.320788	75.93479
B HOSTEL	11.320595	75.9359
C HOSTEL	11.320384	75.936791
D HOSTEL	11.320177	75.937621
E HOSTEL	11.319707	75.938474
F HOSTEL	11.320914	75.937614
G HOSTEL	11.321387	75.936741
AUDITORIUM	11.322719	75.935858
EAST CAMPUS	11.32293	75.937577
COMPANY MUKKU	11.324827	75.930691
ANSARI HOTEL	11.324996	75.930637
REGIONAL POULTRY	11.321922	75.930655
CHATHAMANGALAM NURSERY	11.320454	75.928336
SBI NITC	11.319673	75.929881
UESI STAFF QUARTERS	11.317912	75.927575
NIELT	11.314542	75.926701
CHILDRENS PARK	11.314459	75.929133
MLH	11.316408	75.930353
LH	11.318177	75.93101
NITC GUEST HOUSE	11.317961	75.933384
NITC QUARTERS	11.316839	75.931929
MBA HOSTEL	11.314874	75.932665
SOMS	11.313905	75.932048
FACULTY QUARTERS	11.315818	75.936233
MBH2	11.317083	75.937646
MBH1	11.317961	75.933384

23 random locations from the 30 locations have been chosen and the Hybrid Model 1 were implemented. We have performed 8 simulations and obtained the following results.

Table 4.3 Simulating Hybrid Model 1 on a higher education institute

Number of iterations	Cost with truck only	Cost with hybrid model 1	Percentage change
1	534.65	279.3	47.76021696
2	451.8	302.5	33.0455954
3	453.55	338.5	25.36655275
4	461.8	251.75	45.48505847
5	450	239.4	46.8
6	442.75	255	42.40542067
7	440.1	265.5	39.67280164
8	469.55	265.5	43.45650091

After performing 8 simulations using the Truck Only Model and Hybrid Model 1, it can be observed that the average cost incurred by using the truck only model is found to be 463.025 whereas the average cost incurred by using the Hybrid model 1 is found to be 274.68125. The savings obtained by implementing the hybrid model 1 is found to be 40.49901835 %. Given below is the graph obtained by comparing the Hybrid Model 1 to the Truck Only model over 8 simulations.

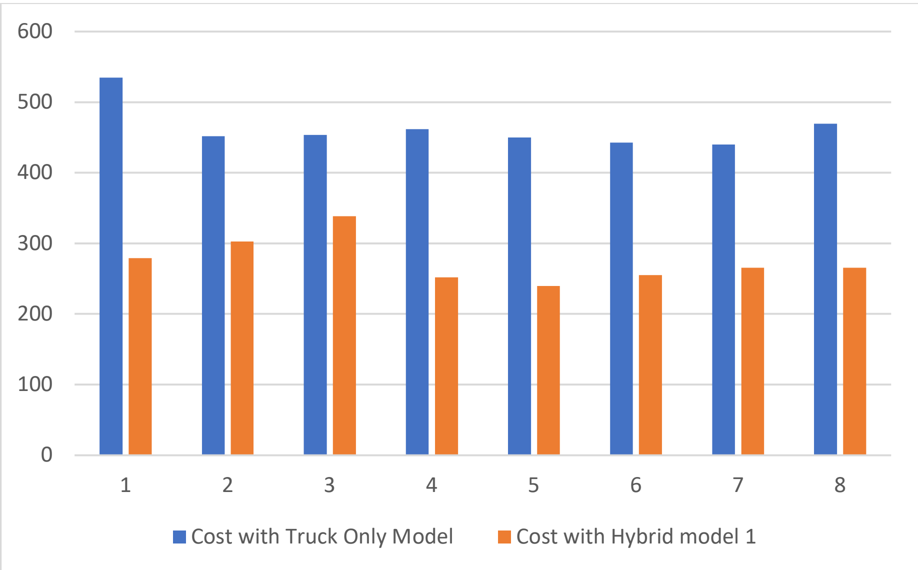

Fig 4.1 Cost comparison graph between Truck only Model and Hybrid Model 1

The number of drones is being varied for a particular delivery situation in NITC and the effect of varying the drones has been studied. Below are the results obtained through the simulations.

Table 4.4 Effect on drone variation

Number of delivery points	**Number of drones**	**Truck-only Model cost**	**Hybrid Model cost**	**Percentage change**
23	4	363.85	294.3	19.11501993
23	5	363.85	255.5	29.77875498
23	6	363.85	259.8	28.59694929
23	7	363.85	279.2	23.26508176
23	8	363.85	186	48.88003298

The graph between the percentage change with variation in number of drones can be observed below. It can be noted that with increase in number of drones the percentage change in cost increases initially and then decreases. This shows that there is an optimum number of drones which can be deployed to obtain the best possible results. Ater a certain number of drones, the clusters contain too many points within it which will make the truck obsolete.

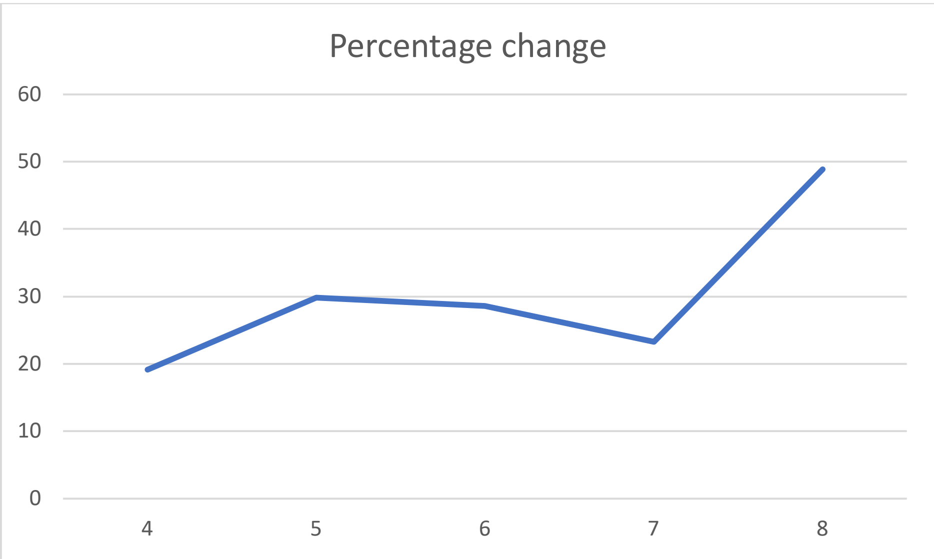

Fig 4.2 Percentage Change with Variation in number of drones

4.3 HYBRID MODEL 2 RESULTS

Comparative analysis is conducted between Hybrid Model-2 and truck only model, utilizing 20 to 30 random locations in a 10 km sq. area via Excel solver. In truck-only model, a truck alone was employed, treating it as a TSP to deliver packages. Conversely, in Hybrid Model 2, a hybrid truck-drone approach was adopted. Here, a clustering algorithm is used to group the delivery points into clusters and determined the focal points for each cluster. The truck traversed through these focal points, stopping at each, while drones completed package deliveries within the cluster. Costs were set at Rs 50 per km for the truck and Rs 6 for drones. For truck-only model, the average cost obtained Rs 624, whereas for Hybrid Model-2, it amounted to Rs 509. Consequently, Hybrid Model-2 yielded an average profit of Rs 115.

Table 4.5 Cost comparison between the 2 models

Number of Points	Cost with truck only	Cost with hybrid model	Percentage change
20	516.7	405.4	21.54054577
23	622	458.21	26.33279743
25	631.292	579.5	8.204127409
27	679.8	553.86	18.52603707
30	671.48	550.3	18.04670281

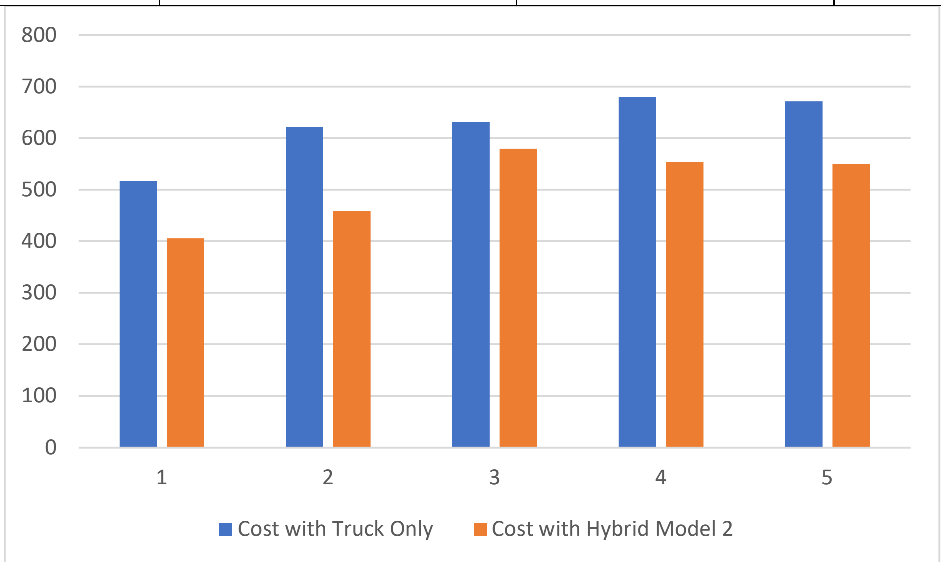

Fig 4.3 Cost comparison graph between Truck-only model and Hybrid model 2

CHAPTER 5
CONCLUSION AND FUTURE SCOPE

5.1 SUMMARY

The last mile delivery is one of the most critical parts of the supply chain management. Cost wise it account for more than 50% of total supply chain cost while it also affects the perception of the customer. With the advancement in e-commerce and q-commerce the demand for fast and cheap delivery is increasing. The use of drone for last mile delivery provides a solution to this problem. The literature review shows that there are many models to accommodate the drone in last mile delivery but its range and its payload capacity become a road block. The use of hybrid model of truck and drone reduce the both problems. To find the best routes there are many algorithms suggested and some are quite innovative and futuristic. But the problem of last mile delivery is still being studied to find the best solution which can cut cost and decrease the delivery time.

5.2 THE CONTRIBUTION OF THIS STUDY

This study presents a sophisticated approach to optimizing delivery routes by integrating the KMeans clustering algorithm with drone routing strategies. Initially, the delivery area is subdivided into smaller clusters using the clustering algorithm, with each cluster assigned a focal point. These focal points serve as strategic hubs from which drones do the final delivery.

The selection of these focal points is done by taking into consideration factors such as minimizing drone travel distances and accommodating the transportation of larger or heavier packages by truck, if necessary. By strategically placing these points, the study ensures overall delivery efficiency. In this procedure, the KMeans algorithm is essential because it forms intelligent clusters that maximize drone travel lengths while preserving the viability of truck-based deliveries. The research illustrates this integrated approach’s cost-effectiveness through testing and analysis.

In conclusion, the integration of clustering algorithms with drone routing presents an innovative solution for optimizing delivery routes. By leveraging this methodology, significant cost savings can be achieved while improving delivery speed and customer satisfaction. This study underscores the potential of innovative approaches to modernise the logistics operations.

5.3 FUTURE SCOPE IN THE WORK

The study has a huge potential of being a solution to the last mile delivery problem. But to be applied in real life scenario the work can be done on following area: -

- Dynamic Cluster Adaption: - The cluster can be made more dynamic in nature by connecting it to real time traffic situation or whether conditions.
- Integration of emerging technology: - Investigate the integration of emerging technologies such as artificial intelligence (AI) and Internet of Things (IoT) devices to enhance route planning and monitoring capabilities.
- Urban delivery challenges: - Address the unique challenges of urban delivery environments, including congested traffic, limited airspace for drones, and restricted access to delivery locations. Future research could focus on developing specialized routing algorithms and delivery strategies tailored to urban settings to overcome these challenges effectively.
- Collaborative Delivery Networks: - Explore the potential for collaboration among multiple delivery providers and stakeholders to optimize delivery routes and share resources more efficiently.

APPENDIX 1
SIMULATION CODE

The code used for simulation of NITC campus is given below: -

```
import pandas as pd

# Load the CSV file
df = pd.read_csv('coordinate_nitc.csv')

# Randomly select 23 rows
random_rows = df.sample(n=23)

# Display the selected rows
print(random_rows)
random_rows.to_csv('coordinate.csv', index=False)

import pandas as pd

def convert_coordinates_from_csv(file_path):
    # Read the CSV file
    df = pd.read_csv(file_path, header=None,skiprows=1)

    # Convert the DataFrame to a list of tuples
    coordinates = list(df.itertuples(index=False, name=None))

    result = []
    for coord in coordinates:
        result.append({"latitude": coord[1], "longitude": coord[2]})
    return result

# Example usage:
file_path = 'coordinate.csv'  # replace with your actual file path
cord = convert_coordinates_from_csv(file_path)
cord
print(cord)

import requests
import json

# Define the API endpoint
url = "https://dev.virtualearth.net/REST/v1/Routes/DistanceMatrix"

# Define the headers for the API request
```

```
headers = {
    'Content-Type': 'application/json',
}

# Define the parameters for the API request
params = (
    ('key', 'Athz-Voune0cVNFWTtgZ2hNoH9m1D9yf2bBhw_6V9CzngRTcCBGEM7D_P7-
rkFAW'),  # replace with your Bing Maps API key
)

# Define the body of the API request
data = {
  "origins": cord,
  "destination": cord,
  "travelMode": "driving"
}


# Make the API request
response = requests.post(url, headers=headers, params=params,
data=json.dumps(data))
# Get the JSON response content
response_json = response.json()

import numpy as np
import pandas as pd

# Extract the 'results' list from the response
resourceSets = response_json.get('resourceSets', [])
if resourceSets:
    resources = resourceSets[0].get('resources', [])
    if resources:
        results = resources[0].get('results', [])
    else:
        print("No resources found")
else:
    print("No resourceSets found")
print(response.status_code)
print(response_json)

# Prepare an empty matrix
matrix = np.empty((len(data['origins']), len(data['destination'])))

# Fill the matrix with the results
for result in results:
```

```
    origin_index = result['originIndex']
    destination_index = result['destinationIndex']
    travel_distance = result['travelDistance']
    matrix[origin_index][destination_index] = travel_distance

# Convert the matrix to a DataFrame for better visualization
df = pd.DataFrame(matrix)
# Set the float format for pandas DataFrame
pd.options.display.float_format = "{:,.4f}".format

# Add headers (location numbers) to the DataFrame
df.columns = [f"Location {i+1}" for i in range(len(data['destination']))]
df.index = [f"Location {i+1}" for i in range(len(data['origins']))]

# Save the DataFrame to a CSV file
df.to_csv('distance_matrix_nitc_full.csv')

print(df)
# real model of nitc
#real working code
import numpy as np
import pandas as pd
from sklearn.metrics import pairwise_distances
import matplotlib.pyplot as plt
from scipy.spatial import ConvexHull
from scipy.spatial.distance import cdist
from k_means_constrained import KMeansConstrained
import csv
from geopy.distance import geodesic

# Function to calculate haversine distance between two points
def haversine(coord1, coord2):
    lat1, lon1 = coord1
    lat2, lon2 = coord2
    return geodesic((lat1, lon1), (lat2, lon2)).kilometers

def plot_clusters(depot_location, customer_locations,
cluster_assignments):
    # Define a list of colors for each cluster
    cluster_colors = ['r', 'g', 'b', 'c', 'm', 'y', 'k']

    for cluster_id in np.unique(cluster_assignments):
        # Get points belonging to the current cluster
        cluster_points = customer_locations[cluster_assignments ==
cluster_id]
```

```
        # Plot cluster points with a different color
        plt.scatter(cluster_points[:, 0], cluster_points[:, 1],
label=f'Cluster {cluster_id}', color=cluster_colors[cluster_id %
len(cluster_colors)])

        # Check if there are at least three points in the cluster to
compute convex hull
        if len(cluster_points) >= 3:
            # Draw convex hull around the cluster points
            hull = ConvexHull(cluster_points)
            for simplex in hull.simplices:
                plt.plot(cluster_points[simplex, 0],
cluster_points[simplex, 1], color='black', linewidth=1)

    # Additional visualization code for depot and customer locations
    plt.scatter(depot_location[0], depot_location[1], label='Depot',
marker='o', color='blue')
    plt.scatter(customer_locations[:, 0], customer_locations[:, 1],
label='Customers', marker='*', color='black')

    plt.xlabel('X')
    plt.ylabel('Y')
    plt.title('Cluster Visualization')
    plt.legend()
    plt.grid(False)  # Removing grid lines
    plt.show()

import requests

# Function to check if a point is reachable by driving using Bing Maps API
def is_point_reachable_by_car(point):
    print(point)
    # Use Bing Maps API to check if the point is reachable by driving
    # Replace 'YOUR_BING_MAPS_API_KEY' with your actual API key
    API_KEY = 'Athz-Voune0cVNFWTtgZ2hNoH9m1D9yf2bBhw_6V9CzngRTcCBGEM7D_P7-
rkFAW'
    depot_lat, depot_lon = 11.320047447954922, 75.93272836230899  # Depot
location
    endpoint =
f'http://dev.virtualearth.net/REST/v1/Routes/Driving?wp.0={depot_lat},{dep
ot_lon}&wp.1={point[0]},{point[1]}&key={API_KEY}'
    response = requests.get(endpoint)
    data = response.json()
```

```
    # Check if a route is returned
    if 'resourceSets' in data and len(data['resourceSets']) > 0 and
'resources' in data['resourceSets'][0] and
len(data['resourceSets'][0]['resources']) > 0:
        return True
    else:
        return False

def update_centroid_distance_matrix(cluster_points_dict,
centroid_distance_matrix, cluster_centroids):
    for cluster_num, cluster_data in cluster_points_dict.items():
        for i, point in enumerate(cluster_data):
            # Check if the point is reachable by car
            if not is_point_reachable_by_car(point):
                # Find the nearest drivable point within the cluster
                distances = np.linalg.norm(cluster_data - point, axis=1)
                min_distance_index = np.argmin(distances)
                min_distance_point = cluster_data[min_distance_index]

                # Update centroid distance matrix with the distance to the
nearest drivable point
                centroid_distance_matrix[cluster_num, i] =
np.linalg.norm(min_distance_point - cluster_centroids[cluster_num])



# K-Means Clustering Function using KMeansConstrained
def kmeans_constrained_clustering(depot_location, customer_locations,
drone_ranges, max_drones):
    # Combine depot location and customer locations
    all_locations = np.concatenate(([depot_location], customer_locations),
axis=0)
    cluster_number = int((len(customer_locations) + 8) / max_drones)

    # Initialize k-means with n cluster
    max_clusters = min(len(customer_locations), max_drones)
    kmeans = KMeansConstrained(n_clusters=cluster_number,
size_max=max_drones, random_state=0).fit(customer_locations)

    # Select a customer location as the focal point for each cluster
    cluster_assignments = kmeans.labels_
    cluster_centers = kmeans.cluster_centers_

     # Enforce size constraint
    while not np.all(np.bincount(kmeans.labels_) <= max_drones):
```

```
        # Identify points belonging to clusters exceeding size_max
        exceeding_clusters = np.where(np.bincount(kmeans.labels_) >
max_drones)[0]

        # Remove points from exceeding clusters
        for cluster in exceeding_clusters:
            cluster_points = np.where(kmeans.labels_ == cluster)[0]
            customer_locations = np.delete(customer_locations,
cluster_points, axis=0)

        # Refit KMeans
        kmeans = KMeansConstrained(n_clusters=1, size_max=max_drones,
random_state=0).fit(customer_locations)

    # Get cluster assignments for each location
    cluster_assignments = kmeans.predict(customer_locations)

    # Get cluster centroids
    cluster_centroids = kmeans.cluster_centers_

    # Create distance matrix between cluster centroids
    centroid_distance_matrix = pairwise_distances(cluster_centroids,
cluster_centroids, metric='euclidean')

    # Create distance matrix between each point and its cluster centroid
    point_to_centroid_distance_matrix =
pairwise_distances(customer_locations, cluster_centroids,
metric='euclidean')

    # Create DataFrames for better control over precision
    centroid_distance_df = pd.DataFrame(centroid_distance_matrix)
    point_to_centroid_distance_df =
pd.DataFrame(point_to_centroid_distance_matrix)

    # Set precision to two decimal places
    precision = 6
    centroid_distance_df = centroid_distance_df.round(precision)
    point_to_centroid_distance_df =
point_to_centroid_distance_df.round(precision)

    # Save distance matrices to CSV files
    centroid_distance_matrix_csv_file_path =
"centroid_distance_matrix_nitc.csv"
    point_to_centroid_distance_matrix_csv_file_path =
"point_to_centroid_distance_matrix_nitc.csv"
```

```
    centroid_distance_df.to_csv(centroid_distance_matrix_csv_file_path,
index=False, header=True)
    point_to_centroid_distance_df.to_csv(point_to_centroid_distance_matrix
_csv_file_path, index=False, header=True)

    print(f"Centroid Distance Matrix saved to
{centroid_distance_matrix_csv_file_path}")
    print(f"Point-to-Centroid Distance Matrix saved to
{point_to_centroid_distance_matrix_csv_file_path}")

    # Create cluster_points_dict with correct structure
    cluster_points_dict = {}
    for cluster_num in np.unique(cluster_assignments):
        cluster_points = customer_locations[cluster_assignments ==
cluster_num]
        cluster_points_dict[cluster_num] = cluster_points

    update_centroid_distance_matrix(cluster_points_dict,
centroid_distance_matrix, cluster_centroids)
    point_to_centroid_distance_1_df =
point_to_centroid_distance_df.round(precision)
    point_to_centroid_distance_matrix_csv_file_path =
"point_to_centroid_distance_1_matrix.csv"

    return cluster_assignments, cluster_centroids,
centroid_distance_matrix, point_to_centroid_distance_matrix,
cluster_points_dict

# Generate 100 random customer locations
num_customers = 25
#Given customer locations
depot_location = np.array([11.320047447954922, 75.93272836230899])  #
Replace with actual depot coordinates
def convert_coordinates_from_m_csv(file_path):
    # Read the entire CSV file
    df = pd.read_csv(file_path)

    # Select only the 'Column2' and 'Column3' columns
    df = df[['Column2', 'Column3']]

    # Convert the DataFrame to a numpy array
    coordinates = df.to_numpy()

    return coordinates
```

```
# Example usage:
file_path = 'coordinate.csv'  # replace with your actual file path
customer_locations = convert_coordinates_from_m_csv(file_path)
print(customer_locations)
# Generate random drone ranges for each customer
drone_range = 25

# Maximum number of drones carried by the truck
max_drones = 8

# Run the clustering algorithm
cluster_assignments, cluster_centroids, centroid_distance_matrix,
point_to_centroid_distance_matrix, cluster_points_dict =
kmeans_constrained_clustering(depot_location, customer_locations,
drone_range, max_drones)


# Visualize the clusters
plot_clusters(depot_location, customer_locations, cluster_assignments)

# Print distance matrix between cluster centroids
print("Distance Matrix between Cluster Centroids:")
print(centroid_distance_matrix)

# Print distance matrix between each point and its cluster centroid
print("Distance Matrix between Points and Cluster Centroids:")
print(point_to_centroid_distance_matrix)

# Print cluster points for each cluster along with distances from the
centroid
for cluster_num, cluster_points in cluster_points_dict.items():
    cluster_centroid = cluster_centroids[cluster_num]

    # Calculate distances for each point in the cluster from its centroid
    distances_from_centroid = np.linalg.norm(cluster_points -
cluster_centroid, axis=1)

    # Create a list of tuples containing the point coordinates and its
distance from the centroid
    point_distance_list = list(zip(cluster_points,
distances_from_centroid))

    print(f"Cluster {cluster_num + 1} Points and Distances from
Centroid:")
    for point, distance in point_distance_list:
```

```
        print(f"Point: {point}, Distance: {distance}")

    # Optionally, if you want to store these distances for later use
    cluster_points_dict[cluster_num] = {'points': cluster_points,
'distances': distances_from_centroid}

# Create and save a CSV file with distances for each point in a cluster
csv_file_path = "cluster_distances_nitc.csv"

with open(csv_file_path, mode='w', newline='') as csv_file:
    fieldnames = ['ClusterNumber', 'PointX', 'PointY',
'DistanceFromCentroid']
    writer = csv.DictWriter(csv_file, fieldnames=fieldnames)

    # Write header to the CSV file
    writer.writeheader()

    # Write data to the CSV file
    for cluster_num, cluster_data in cluster_points_dict.items():
        cluster_points = cluster_data['points']
        cluster_centroid = cluster_centroids[cluster_num]


        # Calculate distances for each point in the cluster from its
centroid in kilometers
        distances_from_centroid = [haversine(point, cluster_centroid) for
point in cluster_points]

        # Write each point along with its distance and cluster number to
the CSV file
        for point, distance in zip(cluster_points,
distances_from_centroid):
            writer.writerow({'ClusterNumber': cluster_num + 1, 'PointX':
point[0], 'PointY': point[1], 'DistanceFromCentroid': distance})

print(f"Cluster distances saved to {csv_file_path}")
# Define the file path for saving the centroid coordinates
centroid_coordinates_csv_file_path = "centroid_coordinates_nitc.csv"
# Write centroid coordinates to the CSV file
def write_centroid_coordinates_to_csv(centroid_coordinates_csv_file_path,
new_centroid_coordinates):
    with open(centroid_coordinates_csv_file_path, mode='w', newline='') as
csv_file:
        fieldnames = ['ClusterNumber', 'NewCentroidX', 'NewCentroidY']
        writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
```

```
        writer.writeheader()

        # Write new centroid coordinates to the CSV file
        for cluster_num, new_centroid in
enumerate(new_centroid_coordinates):
            writer.writerow({'ClusterNumber': cluster_num + 1,
                             'NewCentroidX': new_centroid[0],
                             'NewCentroidY': new_centroid[1]})

# Save centroid coordinates to CSV with new coordinates
new_centroid_coordinates = cluster_centroids
write_centroid_coordinates_to_csv("centroid_coordinates_nitc.csv",
new_centroid_coordinates)

print(f"Centroid coordinates saved to centroid_coordinates.csv")




import pandas as pd

def convert_coordinates_from_csv(file_path):
    # Read the CSV file
    df = pd.read_csv(file_path, header=None,skiprows=1)

    # Convert the DataFrame to a list of tuples
    coordinates = list(df.itertuples(index=False, name=None))

    result = []
    for coord in coordinates:
        result.append({"latitude": coord[1], "longitude": coord[2]})
    return result

# Example usage:
file_path = 'centroid_coordinates_nitc.csv'  # replace with your actual
file path
cord = convert_coordinates_from_csv(file_path)
cord
print(cord)
import requests
import json

# Define the API endpoint
url = "https://dev.virtualearth.net/REST/v1/Routes/DistanceMatrix"
```

```
# Define the headers for the API request
headers = {
    'Content-Type': 'application/json',
}

# Define the parameters for the API request
params = (
    ('key', 'Athz-Voune0cVNFWTtgZ2hNoH9m1D9yf2bBhw_6V9CzngRTcCBGEM7D_P7-
rkFAW'),  # replace with your Bing Maps API key
)

# Define the body of the API request
data = {
  "origins": cord,
  "destination": cord,
  "travelMode": "driving"
}


# Make the API request
response = requests.post(url, headers=headers, params=params,
data=json.dumps(data))
# Get the JSON response content
response_json = response.json()

import numpy as np
import pandas as pd

# Extract the 'results' list from the response
resourceSets = response_json.get('resourceSets', [])
if resourceSets:
    resources = resourceSets[0].get('resources', [])
    if resources:
        results = resources[0].get('results', [])
    else:
        print("No resources found")
else:
    print("No resourceSets found")
print(response.status_code)
print(response_json)

# Prepare an empty matrix
matrix = np.empty((len(data['origins']), len(data['destination'])))

# Fill the matrix with the results
```

```python
for result in results:
    origin_index = result['originIndex']
    destination_index = result['destinationIndex']
    travel_distance = result['travelDistance']
    matrix[origin_index][destination_index] = travel_distance

# Convert the matrix to a DataFrame for better visualization
df = pd.DataFrame(matrix)
# Set the float format for pandas DataFrame
pd.options.display.float_format = "{:,.4f}".format

# Add headers (location numbers) to the DataFrame
df.columns = [f"Location {i+1}" for i in range(len(data['destination']))]
df.index = [f"Location {i+1}" for i in range(len(data['origins']))]

# Save the DataFrame to a CSV file
df.to_csv('distance_matrix_nitc.csv')

print(df)
```

REFERENCES

[1] Joerss, M., Neuhaus, F., Schroder, J., 2016. How customer demands are reshaping last-mile delivery.

[2] Raj, A., Sah, B., 2019. Analyzing critical success factors for implementation of drones in the logistics sector using grey-DEMATEL based approach. Comput. Ind. Eng. 138

[3] Tuerk, M., 2019. Fixing Amazon’s Drone Delivery Problem.

[4] Mohamed Salamaa , Sharan Srinivas, 2020. joint optimization of customer location clustering and drone-based routing for last-mile deliveries

[5] Timothy Mulumba, Waleed Najy, Ali Diabat 2023. The drone-assisted pickup and delivery problem: An adaptive large neighborhood search metaheuristic

[6] Júlia C. Freitas, Puca Huachi V. Penna, Túlio A.M. Toffolo 2022. Exact and heuristic approaches to Truck–Drone Delivery Problems

[7] Minh Anh Nguyena, Giang Thi-Huong Dang, Minh Hoàng Hàa , Minh-Trien Phamc, 2020. The min-cost parallel drone scheduling vehicle routing proble