UiT The Arctic University of Norway

Department of Industrial Engineering

# Developing a Vans-and-Drones System for Last-Mile Delivery

Kristina Charlotte Pedersen Aasen

Master's Thesis in Industrial Engineering, INE-3900 - May 2021

# Abstract

The e-commerce industry is experiencing rapid growth, and growing customer expectations and demand challenges the industry to find more cost-efficient ways of performing the last-mile deliveries. Drones have in recent years been a hot topic, and with high versatility and several application areas it may be the answer to the challenge. In this project a Vans-and-Drones System for Last-Mile Delivery have been developed considering effective task allocation and route scheduling. A literature review is presented on the topic of drone technology and application areas, especially emphasizing utilization of drones in logistic operations and routing problems. A mathematical model for the Vehicle Routing Problem with Drones is derived based on the classical Capacitated Vehicle Routing Problem, and the formulation is modeled in Jupyter Notebook with Python programming language and solved with CPLEX solver.

A case study is carried out to examine the effects of integrating drones into the delivery system for a vaccine distribution scenario in a sparsely populated area, Ofoten region, considering vehicle employment cost, delivery time and emission impact. Results show that the proposed vans-and-drones system outperforms a truck-only delivery system for this purpose.

# Acknowledgement

I would like to express my deepest gratitude and appreciation to my supervisor Assoc. Prof. Hao Yu for his invaluable support and guidance during the execution of this project, and for providing such an interesting topic for my master's thesis.

Table of Contents


1 Introduction ........ 1

1.1 Background ........ 1

1.2 Problem Statement ........ 1

1.3 Research objective ........ 2

1.4 Limitations ........ 2

2 Literature Review ........ 4

2.1 Unmanned Aerial Vehicles ........ 4

2.1.1 Fuselage ........ 4

2.1.2 Application Areas ........ 5

2.2 Routing Problems with Drones ........ 6

2.2.1 The TSP-D ........ 6

2.2.2 The FSTSP ........ 7

2.2.3 The VRPD ........ 9

2.3 Environmental Impact ........ 11

3 Methodology ........ 12

3.1 CVRP with Drones ........ 12

3.2 Mathematical formulation ........ 12

3.3 Modeling Phase ........ 14

3.3.1 Generation of Customer Locations and Distance Matrices ........ 14

3.3.2 Visualization of Generated Customer Locations ........ 18

3.3.3 Coding the Mathematical Formulation ........ 21

3.3.4 Illustration of Results ........ 23

4 Verification and Validation Phase ........ 27

4.1 Comparison of VRPD model and Google OR-Tools CVRP ........ 27

4.2 Case Study ........ 28

4.2.1 Case Study Part 1 – Validation ........................................................ 30

4.2.2 Case Study Part 2 ........................................................................... 31

5 Discussion ............................................................................................ 38

6 Conclusion............................................................................................ 40

7 Future Work .......................................................................................... 41

References................................................................................................. 42

# List of Tables

Table 1 - Pros and cons on different fuselages ........ 5
Table 2 - Comparison Study Google OR-Tools Results ........ 28
Table 3 - Customer Locations ........ 29
Table 4 – Comparison Results Vans-and-Drones and Truck-Only Systems ........ 30
Table 5 - Sensitivity Analysis Results ........ 31
Table 6 – Results from Case Study Part 2 ........ 33
Table 7 - Travel Time Analysis ........ 34
Table 8 - Truck Employment Cost Analysis ........ 34
Table 9 - Emission Analysis ........ 35

# List of Figures

Figure 1 - Jupyter Notebook Generation of Customer Locations and Demand...................... 16
Figure 2 - Jupyter Notebook Generation of Distance Matrix Truck ..................................... 17
Figure 3 - Jupyter Notebook Generation of Distance Matrix Drone..................................... 18
Figure 4 – Jupyter Notebook Customer Node Plot .............................................................. 19
Figure 5 - Jupyter Notebook Customer Node Google Maps Plot ......................................... 20
Figure 6 - Jupyter Notebook Coding the Mathematical Formulation 1 ................................ 21
Figure 7 - Jupyter Notebook Coding the Mathematical Formulation 2 ................................ 22
Figure 8 - Jupyter Notebook CPLEX Solver ....................................................................... 23
Figure 9 - Jupyter Notebook Plotting Results....................................................................... 24
Figure 10 - Jupyter Notebook Code for Plotting Results in Google Maps............................ 25
Figure 11 – Jupyter Notebook Results plotted in Google Maps ........................................... 26
Figure 12 - Customer Locations for Setting 1 and 2 ............................................................ 32
Figure 13 - Customer Locations for Setting 3....................................................................... 32
Figure 14 - Test Instance (2)................................................................................................. 36
Figure 15 - Test Instance (3)................................................................................................. 36
Figure 16 - Test Instance (4)................................................................................................. 36
Figure 17 - Test Instance (5)................................................................................................. 36
Figure 18 - Test Instance (6)................................................................................................. 36
Figure 19 - Test Instance (8)................................................................................................. 37
Figure 20 - Test Instance (9)................................................................................................. 37

# 1 Introduction

## 1.1 Background

In order to survive in today's competitive market, logistics companies are engaging to provide a faster and more cost-efficient last-mile delivery service to customers [1]. At the same time, with technological advancements, the effective use of drones has been extensively focused in a large variety of industries, i.e., facility maintenance, entertainment, disaster management, and transportation and logistics [2]. Thus, the development of a vans-and-drones system for improving last-mile delivery services has been widely focused by both academicians and practitioners.

The e-commerce industry has been experiencing rapid growth during recent years, this together with urbanization is putting the delivery logistics systems to the test [3]. Customers get more and more demanding and tend to choose the retailer option with the shortest delivery time. Companies often offer same day or next day delivery and the parcel delivery operation, especially with last mile, is easily the most expensive part of the supply chain. The e-commerce industry therefore seeks to find a more cost-efficient option while maintaining customer satisfaction [4]. The huge increase in parcel deliveries is putting strain on the infrastructure, especially in urban regions leading to traffic congestion and negative environment and health impacts[5]. UAVs are a promising solution to this challenge, being airborne resulting in congestion-free routing, and with relatively low operational costs and environmentally friendly when powered by electricity from low-carbon energy sources.

With today's drone technology there are still some setbacks, for instance with the rotary-winged drone that is the most tested for parcel deliveries in urban areas. The drones still have limited capacity related to range due to high power consumption, and often need hubs or recharging stations nearby. Another issue is low carrying capacity, or payloads. Researchers are now addressing these issues to improve the technology and its performance, as well as improving drone-assisted parcel delivery logistics systems in routing and scheduling [4].

## 1.2 Problem Statement

This project aims at improving the last-mile parcel delivery operation and increase the competitiveness of logistics services to satisfy growing customer demand and expectations. The last-mile delivery operation is very expensive and time consuming, the problem to be explored in this project is whether vans-and-drones last-mile delivery systems can outperform traditional truck-only last-mile delivery systems when considering delivery cost and emission [4].

A vans-and-drones optimization model is to be developed and tested to see if the effectiveness of the last-mile delivery system can be improved by integrating drones in the delivery process. In addition, the optimization model shall be tested in a real-world case study. The world is currently battling a pandemic. Vaccines have been created to defeat the virus, but the next challenge is how to effectively distribute the vaccines to the public. A case-study that will be executed is based on this vaccine distribution challenge and will test if drones can play a key role in the effective distribution of the vaccines in Ofoten region, considering delivery time-consumption, cost, and emission.

## 1.3 Research objective

The objectives of this project is;

- The development of an optimization model for task allocation and route scheduling of a vans-and-drones last-mile delivery system, which shall be verified and validated.
- Comparison data from the Vans-and-Drones Last-Mile Delivery System versus truck-only last-mile delivery systems shall be identified and derived.
- Most importantly, a small-scale case-study shall be executed where the optimization models effectiveness is tested in the scenario of vaccine distribution in a sparsely populated area, Ofoten region, with consideration of delivery time, cost and emission level compared to truck-only systems.

**Scope:**

1. Formulate a mathematical model of a Vans-and-Drones System for Last-Mile Delivery to optimize the task allocation and routing.
2. Code and solve the mathematical model with an industrial optimization solver.
3. Verify and validate the Vans-and-Drones Delivery System.
4. Perform a case study to test the model in distribution of vaccines in Ofoten region.

## 1.4 Limitations

Even though drones can be a useful tool in several industries, the logistic industry is the area of interest for this project. There are several factors to take into consideration when operating UAVs, especially for commercial use. Drone operations in civil areas are subject to strict rules and regulations and the UAV devices needs to be operated/controlled by authorized personnel under continuous monitoring [6]. However, for this project the focus area will be on the Operation Research aspects of the last-mile delivery operations with drones, considering logistics and operational problems related to routing, allocation, and network design.

Another area that will not be emphasized is the technical specifications and details for the drone devices.

# 2 Literature Review

## 2.1 Unmanned Aerial Vehicles

Unmanned Aerial Vehicles, UAVs, commonly known as *drones*, are aircrafts with no onboard human pilot, controlled by autonomous navigation or remote-control navigation [7] . UAVs are not a new invention, in fact they have been around for over a decade. The first reported drone dates back as far as 1916, created by the Americans Lawrence and Sperry. However, UAVs did not have a prominent role in either of the world wars, due to the immaturity of the technology. It was not until the 1950's that the UAV technology really started to develop. [7]

For several years UAV technology have been deployed in military operations for reconnaissance and combat purposes. However, in recent years it has become more accessible to the public.

### 2.1.1 Fuselage

Drones comes in different shapes and sizes, listed in Table 1, based on Macrina, et al. [4], are the main categories with general advantages and disadvantages. The sizes range from 1 mm (wingspan) and 0,005 g to 61 m and 15,000 kg. The most common drone types are the fixed- and rotary-winged drones. In 2016 an American start-up called Zipline International used fixed-winged drones to delivery medical supplies to remote areas in Rwanda [4]. The fixed-winged drones are well suited for this kind of delivery operations, since they have relatively low energy consumption and can travel long distances with high payloads, drop of delivery packages, and return to depot while maintaining high traveling speed throughout the operation.

Rotary-winged drones are probably the most known drones for the public, especially the quadcopter with four rotors. The fuselage gives it great maneuverability and the possibility to hover in the air, making it ideal for operations where high flexibility is required.

In an attempt to combine the advantages for both the fixed- and rotary-winged drone, different companies, like Amazon, are designing hybrids that can travel fast with high range and payloads while having good maneuverability and adopting the rotary-winged drones hovering features [8].

| | | |
|---|---|---|
| **Fixed-wing** | Similar to traditional airplane |
| | Pros: Longest range, heavy payloads, low maintenance cost |
| | Cons: Low maneuverability |
| **Rotary-wing** | Helicopter, Quadcopter, Hexacopter, Octocopter |
| | Pros: High maneuverability |
| | Cons: High energy consumption |
| **Flapping-wing** | Imitates birds/insects flying |
| | Pros: High maneuverability (when small sized) |
| | Cons: Low range, high maintenance cost, high energy consumption |
| **Hybrid** | Combine fixed-, rotary- and flapping-wing features |
| | Often have longer range and high payload |

*Table 1 - Pros and cons on different fuselages*

### 2.1.2 Application Areas

UAVs are very versatile tools that have several application areas. Macrina, et al. [4] performed an extensive literature review on the drone-aided routing problem, and adopted the work of Singhal, et al. [9], where they grouped the application areas into three different classifications; civilian, environment and defense.

#### 2.1.2.1 Civilian

Drones are widely used by hobbyists for recreational purposes, like photographing. In the industry drones have been proven a useful tool, e.g., in construction, mining, agriculture, disaster management, and most importantly for this literature study; logistics and delivery.

The online-shopping industry have been experiencing rapid growth in recent years, which results in increased competition between companies, as well as a growing population of online retailers. Customers become more demanding and tend to purchase from the online shops with the fastest delivery time. The last-mile delivery operation is for many retailers the most expensive part of the supply chain. Therefore, they explore delivery options to reduce this cost. Drone-aided routing may be the solution to this challenge.[4]

#### 2.1.2.2 Environment

The drone technology makes it more accessible to monitor environmental situations like air quality, crops, and ecosystems, and inspect mountains and observe the environmental effects from the climate changes. Different drones can operate from the air, on the water surface and under water. Underwater drones are used for studying ocean animals or under water regions, such as polar regions.[4]

#### 2.1.2.3 Defense

For defense operations UAVs are used for espionage, missile launches, boarder surveillance, bomb dropping, supplying medical supplies to warzones and other combat purposes. [4]

## 2.2 Routing Problems with Drones

There are several configuration models for delivery routing problems with drones, for this literature study the Traveling Salesman Problem with Drone, TSP-D, the Flying Sidekick Traveling Salesman Problem, FSTPS, and the Vehicle Routing Problem with Drones, VRPD, is considered and reviewed. The forthcoming paragraphs will explain these delivery models, accompanied with recent literature on the subjects.

### 2.2.1 The TSP-D

The Traveling Salesman Problem with Drones, TSP-D originate from the classical Traveling Salesman Problem, TSP which is an NP-hard problem in combinatorial optimization. The objective of the TSP is to find the shortest possible path concerning minimum cost or distance travelled while satisfying different requirements and constraints. The TSP-D includes drones in this operation. Hernández, et al. [10] studied the usage of UAVs in parcel delivery, in collaboration with the traditional trucks. With basis in the TSP, they proposed a mathematical model to solve the problem with the goal to increase efficiency in the distribution network. They emphasized several advantages with the implementation of UAVs in the distribution network, in relation to traveling speed, accessibility, flexibility and the fact that the drones can operate autonomously without the need for human support.

Ha, et al. [11] presented an extension of the work from Murray and Chu [12] on the FSTSP. They propose a new variant of the TSP-D, which they chose to call min-cost TSP-D. Unlike Murrey and Chu's work, where they focus on developing a model to reduce the delivery completion time, Ha, et al.[11] propose a model where the objective is to minimize the total operational cost of the delivery process.

Moshref-Javadi, et al. [13] notice a limited research base on the topic synchronized multi-echelon routing. They studied multi-modal delivery systems with truck and drones and identified two major disadvantages with the usage of UAVs in the parcel delivery process, geographical reach, or flight time, and carrying capacity. The first is due to limited battery life, where the drones require recharging or replacement of the batteries after a certain period of time or flight distance. The latter considers limited carrying capacity of the drone in relation to parcel size and weight. This needs to be considered when deciding the customer allocation between the truck and drones.

Due to the limitations with today's technology a purely UAV based parcel delivery system could be challenging [13]. On the other hand, during this literature review we see more and more researchers investigate delivery systems where the traditional trucks work in collaboration with UAVs, combining their strengths to create a more efficient system. With the trucks having high travel ranges and ability to carry bigger payloads with parcel of both high volume and weight, and the drone's flexibility in travel routes and speed.

Moshref-Javadi, et al. [13] proposed a multi-modal last-mile delivery system based on the traditional Traveling Repairman Problem, TRP, called the Simultaneous Repairman Problem with Drones, STRPD. The problem considers one truck and one or more drones operating in synchronization with each other, with the objective of minimizing the customer waiting time. Due to the synchronization between the truck and the drones, the drones are able to dispatch from the truck, serve a customer, and return to the truck on a subsequent truck stop, or a rendezvous location. This function extends the service range for the drone by making the truck serve as a portable hub.

The STRPD was formulated as a Mixed-Integer Linear Programming problem, MILP. A Truck and Drone Routing Algorithm, TDRA, was also designed to efficiently solve the STRPD in real-world instances. Compared with the MILP model, the TDRA was able to provide optimal or near optimal solutions for small-scale problems.

### 2.2.2 The FSTSP

The Flying Sidekick Traveling Salesman Problem, FSTPS was introduced by Murray and Chu [12] in 2015. In the FSTSP a set of customers is served either by a single truck or a single drone, with the objective of minimizing the completion time of the operation. Each customer must be served exactly once, and the drone can be transported by the truck. In recent years, several researchers have studied and extended their work, proposing new solving methodologies and heuristics to reduce the completion time for the algorithms and to be able to solve problems with larger instances.

Crişan and Nechita [14] proposed and tested a new heuristic so solve the FSTSP, called the New Greedy Heuristic, NGH. This heuristic is stated to be less complex than the one proposed by Murray and Chu [12]. For the proposed method they included a new cost function to consider the drones limited flying time, with the objective of minimizing the total transportation cost for the truck and drone delivery problem. They used multiple instances for the testing of the heuristics, both real-world and theoretical instances to fully analyze the behavior of the NGH method. The results came out promising and provided time savings for the total system.

Jeong, et al. [15] focus on the importance of considering drone energy consumptions related to parcel weight, and restrictions in flying areas for UAVs when operating with truck and drone delivery systems. They notice that these factors tend to not be included in the literature, so they emphasize it in their studies aiming to receive more accurate and realistic data. The authors proposed a model called FSTSP-ECNZ, a hybrid last-mile delivery model that considers one truck and one drone. They developed a new solution algorithm with MILP formulation and tested it up against known metaheuristics like nearest neighbor, particle swarm optimization, generic algorithm, and simulated annealing. The model considers the drones energy consumption related to the parcel weight carried by the drone, in addition to proposing detour routes for the drone to avoid time-dependent restricted flight zones. To derive these routing schedules, they developed a heuristic named the Two-Phase Construction and Search Algorithm, TPCSA, which combines construction heuristics and search heuristics. The truck and drone routing problems tend to be quite complex, but the authors states that their TPCSA method is able to give promising solutions within a reasonable time frame.

Dell'Amico, et al. [16] proposed a new formulation for the solving methodology of the truck and drone delivery problem, also with focus on the FSTSP. They presented a two- and a three-indexed formulation in a new objective function, as well as inequalities for application in the branch-and-cut method, with the objective of finding optimal solutions to the problems within short time. They considered two scenarios of the delivery problems and adapted the solving methodologies for both; one in which the drone is allowed to wait at customer location to save battery power, and one where the drone is only allowed to wait while in the air. Their methods gave promising results, and the authors claimed it to outperform other optimization methods, especially the two-index formulation, by finding optimal or near optimal solution to a high percentage of the benchmark instances tested within a reasonable amount of time. The endurance of the drone proved to be the feature that impacted the algorithms the most, with lower endurance allowing the algorithms to perform better.

Raj and Murray [17] introduced the Multiple Flying Sidekick Traveling Salesman Problem with Variable Drone Speed, mFSTSP-VDS. In this problem a truck work in conjunction with a fleet of drones in the last-mile delivery system, where the objective is to minimize the makespan. Unlike the mFSTPS, where the drones are assumed to travel at a constant speed, the mFSTPS-VDS considers the drone speed as a decision variable, where the drones can travel at varying velocities. To solve the problem a three-phase algorithm was developed, which were able to obtain the optimal performance by dynamically varying the drone velocity. The challenge was to find the best trade-off between the drones maximum traveling speed and the maximum travel range relative to the drone endurance, to minimize the makespan. The result from the study suggests that the performance of the system is improved, in terms of overall delivery time, when utilizing drones where their maximum speed exceeds the speed at which the range is maximized. Also, the drones tend to take on a higher number of deliveries relative to the truck when the velocity varies, which often results in reduces makespan.

### 2.2.3 The VRPD

The Vehicle Routing Problem with Drones, VRPD is an extension of the classical Vehicle Routing Problem, VRP that considers drones in the delivery operation. Kitjacharoenchai and Lee [3] proposed a variant of the VRPD, with the objective of minimizing the total transportation time. The model is based on the FSTSP and their own, previously published work of the mTSPD, but unlike these models the VRDP considers capacity constraints of the vehicles, both truck and drone, and the order of launching and landing operations. They suggested a MIP algorithm to find optimal solutions to the routing problem with small instances. The results from the study reported that the usage of drones in the delivery problem reduces the transportation time in the last-mile delivery process.

Wang and Sheu [18] also considered the VRPD but reformulated the problem as a path-based model and an arc-based MIP model was proposed with a branch-and-price algorithm. They compared their VRPD solution to the VRP general solution and found that an average cost saving of 20% was obtained, as well as an average 5 min decrease in waiting time for each customer. Their results also supported the perception that the VRP is improved by including drones in the delivery process.

Lin, et al. [19] considered parcel delivery systems with one truck and multiple drones in collaboration. They wanted to create a system that could be suitable for various city road networks since they noticed a lack of research in this area. To solve the routing problem in their distribution system they propose a mixed-integer programming model based on an urban road network. For the solving methodology they presented two algorithms, a hybrid genetic algorithm, h-GA, and a hybrid particle swarm optimization algorithm, h-PSO. Testing of the algorithms showed promising results and their solving methodology was claimed to outperform existing basic solving algorithms.

Similarly, Salama and Srinivas [1] focused on the last mile delivery problem with one truck and multiple drones. However, they proposed a system with non-overlapping clustering, where the cluster focal point can either be on a customer location, JOCR-R, or anywhere in the delivery region, JOCR-U, unlike common literature on the same field. By not restricting the clustering focal points to be placed on a customer location, total operational cost and delivery completion times could be reduced. They introduced a mathematical programming model that jointly optimizes the clustering delivery problem, by using machine learning-based heuristic and knowledge-based constraints. Their objective was to find the best trade-off solution between reduction in total cost and delivery completion time. They tested their method up against known sequential heuristic methods and their model proved to be a better solution method in all cases.

Schermer, et al. [20] studied the multi-drone VRPD by formulating it as a MILP. They introduced sets of valid inequalities to improve the performance of the general MILP solvers. Since this approach is only possible for problems with small instances, they proposed a new metaheuristic approach where they included the Drone Assignment and Scheduling Problem, DASP, for handling problems with larger instances. By considering an existing routing of trucks, the DASP model utilizes two MILP formulations aiming to minimize the makespan by searching for optimal assignment and scheduling for a set of drones. Results from the study suggested that the makespan could be reduced by applying drones.

Similar to Kitjacharoenchai and Lee (2019), Sacramento, et al. [21] also studied an extension of the VRPD. They presented a new mathematical formulation of the VRPD by extending the MIP formulation utilized in the FSTPS, considering both several truck and drones, and time and capacity limitations. For the solving methodology for the multi-truck problem, they propose a metaheuristic with Adaptive Large Neighborhood Search, ALNS, with the objective of minimizing total operational cost. The obtained results suggested that the VRP works better when including drones.

## 2.3 Environmental Impact

Elsayed and Mohamed [6] studied how the transitioning to drones in the first- and last-mile delivery will affect the environment regarding emission, considering airspace regulations and policies. Traditionally, the delivery of parcel is executed by a truck or similar vehicle powered by fossil fuels, which is known have a poor impact on the environment due to high CO2 and GHG emissions. Electric powered ground delivery vehicles reduce this emission but does not reduce the traffic congestion in urban areas. An important factor to consider is that strict aerial flight regulations may disrupt the UAVs traveling path, forcing the UAV to reroute the delivery path to stay clear of the restricted areas, or even making the customer location inaccessible, causing up to 400% increase in CO2e emission compared to lean regulations. Results from the study suggested that UAVs in the parcel delivery system would be as much as a 1000-fold more CO2 efficient compared to traditional trucks, and around 30% more efficient compared to electric ground vehicles.

# 3 Methodology

## 3.1 CVRP with Drones

The capacitated VRPD model presented in this report is based on the classical VRP, which is an extension of the classical TSP by Dantzig and Ramser [22]. The mathematical formulation proposed in this report is derived based on the information gathered from the literature review, and the works presented by Dantzig and Ramser [22], Nanda Kumar and Panneerselvam [23], Kulkarni and Bhave [24] and Laporte [25].

The VRP is an NP-hard problem, hence it is difficult to get an optimal solution within reasonable runtime for problems with larger instances when using exact solution methods [23]. However, for the application of the VRPD model in this project the instances are kept low, under 50 – 100 which is considered to be the size limit for exact VRP solving methods [23], so an exact solving method is utilized.

## 3.2 Mathematical formulation

The VRPD is defined on a complete graph $G = ( V, E )$, where $V$ represents a set of $n$ customers defined as $V = \{ 0, 1, 2, \ldots, n \}$, with a depot located at $V$. $E$ is a set of arcs connecting each node pair, $e_{ij} = (i, j)$. $K$ is a homogeneous fleet of trucks defined in a set $K = \{ 1, 2, \ldots, k \}$, where each truck has maximum carrying capacity $Q_k$, and the set $KD = \{ 1, 2, \ldots, kd \}$ defines a homogenous fleet of drones with maximum carrying capacity $Q_{kd}$. Both the truck and drone fleets are initially located at the depot. Each customer node has a demand, $q_i \geq 0$, that needs to be satisfied by either a truck or a drone, except for the depot node where the demand is $q_0 = 0$. Let $c^k_{ij}$ and $c^{kd}_{ij}$ denote the cost of travel over arc $e_{ij}$ for truck and drone, respectively. For this model, the cost parameters $c^k_{ij}$ and $c^{kd}_{ij}$ are represented by the distance traveled over arcs for each vehicle, for the trucks the distance is actual driving distance between node $i$ and $j$, whilst for the drone the distance is flight distance in a straight line. Let $x^k_{ij}$ and $x^{kd}_{ij}$ be the principal decision variables for truck and drone respectively, where $x^k_{ij} = 1$ if truck $k$ travels along arc $e_{ij}$, and 0 otherwise. Same goes for $x^{kd}_{ij}$ in the drone set. Based on the generalized subtour elimination constraint (GSECs), where r(s) corresponds to the minimum number of vehicles required to serve set S, which is subset of V. [26]

**Objective function:**

$$(1)\quad min \sum_{k\in K} \sum_{(i,j)\in E} c_{ij}^{k} x_{ij}^{k} + \sum_{kd\in KD} \sum_{(i,j)\in E} c_{ij}^{kd} x_{ij}^{kd}$$

**Subject to:**

$$(2)\quad \sum_{k\in K} \sum_{i\in V, i\neq j} x_{ij}^{k} + \sum_{kd\in KD} \sum_{i\in V, i\neq j} x_{ij}^{kd} = 1 \qquad ,\forall j \in V \setminus \{0\}$$

$$(3)\quad \sum_{j\in V\setminus\{0\}} x_{0j}^{k} = 1 \qquad ,\forall k \in K$$

$$(4)\quad \sum_{j\in V\setminus\{0\}} x_{0j}^{kd} = 1 \qquad ,\forall kd \in KD$$

$$(5)\quad \sum_{i\in V\setminus\{0\}} x_{i0}^{k} = 1 \qquad ,\forall k \in K$$

$$(6)\quad \sum_{i\in V\setminus\{0\}} x_{i0}^{kd} = 1 \qquad ,\forall kd \in KD$$

$$(7)\quad \sum_{i\in V, i\neq j} x_{ij}^{k} + \sum_{i\in V, i\neq j} x_{ij}^{kd} - \sum_{i\in V} x_{ji}^{k} + \sum_{i\in V} x_{ji}^{kd} = 0 \quad ,\forall j \in V, \forall k \in K, \forall kd \in KD$$

$$(8)\quad \sum_{i\in V} \sum_{j\in V\setminus\{0\}, i\neq j} q_j x_{ij}^{k} \leq Q_k \qquad ,\forall k \in K$$

$$(9)\quad \sum_{i\in V} \sum_{j\in V\setminus\{0\}, i\neq j} q_j x_{ij}^{kd} \leq Q_{kd} \qquad ,\forall kd \in KD$$

$$(10)\quad \sum_{k\in K} \sum_{(i,j)\in S, i\neq j} x_{ij}^{k} \leq |S| - r(s) \qquad ,S \subseteq V \setminus \{0\}$$

$$(11)\quad \sum_{kd\in KD} \sum_{(i,j)\in S, i\neq j} x_{ij}^{kd} \leq |S| - r(s) \qquad ,S \subseteq V \setminus \{0\}$$

$$(12)\quad x_{ij}^{k} \in \{0,1\} \qquad ,\forall (i,j) \in E, \forall k \in K$$

$$(13)\quad x_{ij}^{kd} \in \{0,1\} \qquad ,\forall (i,j) \in E, \forall kd \in KD$$

The mathematical formulation for the proposed VRPD model with the parameters, sets and decision variables are defined above, where objective function (1) minimizes the total cost of travel for completing the tour for both the trucks and the drones. Constraint set (2) ensures that each customer node is visited only once by only one vehicle, either truck or drone. Constraint set (3) and (4) guarantees that each vehicle from both the truck and drone fleets depart from the depot. Similarly, constraint set (5) and (6) guarantees that each vehicle returns to the depot. Constraint set (3), (4), (5) and (6) jointly ensures a complete tour for each vehicle. Constraint set (7) restricts that for each customer the same number of vehicles enters and leaves the node. Constraint set (8) and (9) limits the delivery capacity of each vehicle to be less than or equal to the maximum vehicle capacity for truck and drone, respectively. Constraint set (10) and (11) ensures the removal of subtours for both truck and drone routes, respectively. Lastly, set (12) and (13) define the binary decision variables for truck and drone respectively, which is 1 if arc $e_{ij}$ is in the vehicle route, and 0 otherwise.

## 3.3 Modeling Phase

The coding process was executed in Jupyter Notebook, which is an open source computational notebook where you can compile different types of data [27]. The code was written in Python language and generated through the linear programming modeler PuLP [28]. The solver used for this model is the CPLEX solver from IBM ILOG, which uses the branch-and-cut exact solving algorithm [29]. An advantage of using the Jupyter web application is that it allows you to visualize the model in the same document while working on it. Using this technique makes it easier to detect errors in the model right away. MatPlot was used to visualize the results in a 2D plotted graph, and functions from Gmaps was used to plot results on Google Maps with customer locations, depot location and vehicle routes. The work by Kim [30] is used as inspiration when performing the modeling phase in Jupyter Notebook.

### 3.3.1 Generation of Customer Locations and Distance Matrices

Jupyter allows you to import different packages and libraries into the notebook that are compatible with Python. Numerical Python, or NumPy, is a Python library used for working with arrays. For this model, the customer locations and associated demands are generated by using the “NumPy. Random” function, and together with Pandas, which is also a Python package, data frames are generated with the customer location data, presented with latitudes and longitudes, and the demands for each customer. For the depot, the demand is set to 0.

The cost for the vehicles is their travel distance which is different for the truck and the drone since the drone can fly in a straight line between locations, while the truck is limited to using roads. To obtain distance matrices for the vehicles two different approaches is used:

- **Drone travel distance**
  The distance matrix for the drones is generated by calculating the Euclidean distance between the customer nodes in 2D plane. For the case study, described later in the report, the distances are calculated using the Haversine formula, which is assumed to be more accurate in a real-world scenario since it considers the spherical form of the earth. Over small distances the difference between the Euclidean and the Haversine distance is presumably negligible, but is considered, nevertheless.

- **Truck travel distance**
  Google Direction API allows you to retrieve actual driving directions between geographical locations that are connected to a road network [31]. This function was utilized to generate the distance matrix for the trucks, which was assembled in a data frame.

For the following sections of the modelling phase an example problem is solved. This problem is only inserted for illustration purposes, and the results are not considered any further. The complete code for the model accompanies this report in a separate .ipynb file called “Vans-and-Drones System for Last-Mile Delivery Jupyter Notebook”.

```python
import numpy as np
import pandas as pd
import cplex
import gmaps
import googlemaps
import pulp
import itertools

API_KEY = ''
gmaps.configure(api_key=API_KEY)
googlemaps = googlemaps.Client(key=API_KEY)

customers = 9
trucks = 2
truck_capacity = 20
drones = 1
drone_capacity = 2
fixed_cost_truck = 50

depot_lat = 68.43852
depot_long = 17.42726
np.random.seed(0)
df = pd.DataFrame({"lat":np.random.logistic(depot_lat, 0.002, customers),
                   "long":np.random.logistic(depot_long, 0.004, customers),
                   "demand":np.random.randint(1, 10, customers)})
df.iloc[0,0] = depot_lat
df.iloc[0,1] = depot_long
df.iloc[0,2] = 0

print(df)
```

```
         lat       long  demand
0  68.438520  17.427260       0
1  68.440361  17.432601       3
2  68.439354  17.427723       4
3  68.438880  17.428356       9
4  68.437904  17.437344       2
5  68.439722  17.416976       4
6  68.438018  17.417863       4
7  68.442738  17.411737       4
8  68.445076  17.433677       8
```

*Figure 1 - Jupyter Notebook Generation of Customer Locations and Demand*

Illustrated above in Figure 1 is the generation of customer locations with latitude and longitude coordinates and the demand for each customer. Location 0 is the depot and has therefore no demand. The necessary packages, libraries and solvers are imported, the API key is inserted in the “API_KEY” slot and the preferred customer count is defined as well as the number of drones and their respective carrying capacity. Furthermore, a fixed cost for employing the trucks can be inserted, which is optional. The depot latitude and longitude coordinates are then defined and works as a basis for the random generation of customer locations with, in this case, logistic distribution within a defined range. For this case, the depot is located in Narvik city center. The distribution around the depot location is optional as well. For the generation of demands upper and lower bounds are defined and a seed is used together with the “numpy. random()” function to get equal results for each run of the kernel. From these inputs a data frame with customer location coordinates and demands will be produced as output.

```
pd.options.mode.chained_assignment = None
# Retrieve Distance Matrix Trucks
def _distance_matrix_truck(_df):
    # Create empty matrix
    _travel_distance_truck = np.zeros((len(_df),len(_df)))
    _df['lat-long'] = '0'
    for i in range(len(_df)):
        _df['lat-long'].iloc[i] = str(_df.lat[i]) + ',' + str(_df.long[i])

    for i in range(len(_df)):
        for j in range(len(_df)):

            # Retrieve distances between customer nodes
            _gmaps_results = googlemaps.directions(_df['lat-long'].iloc[i],
                                                   _df['lat-long'].iloc[j],
                                                   mode = 'driving')

            # Append distances to matrix
            _travel_distance_truck[i][j] = _gmaps_results[0]['legs'][0]['distance']['value']

    return _travel_distance_truck

distance_truck = _distance_matrix_truck(df)

print(distance_truck)

```

```
[[   0.  354.  736.  115. 1000.  521.  789. 1340. 1110.]
 [ 604.    0. 1342.  541.  645. 1262. 1395. 1908.  756.]
 [ 231.  298.    0.  168.  792.  753. 1021. 1572. 1053.]
 [  63.  239.  800.    0.  885.  585.  853. 1404.  995.]
 [ 985.  542. 1268.  922.    0. 1507. 1321. 2425. 1273.]
 [ 560.  850. 1232.  611. 1495.    0. 1285. 1133. 1122.]
 [ 803. 1093.  853.  854. 1322. 1325.    0. 2144. 1849.]
 [1379. 1669. 2051. 1430. 2388.  853. 2104.    0. 1768.]
 [1044. 1399. 1781.  981. 1236. 1122. 1834. 1768.    0.]]
```

*Figure 2 - Jupyter Notebook Generation of Distance Matrix Truck*

A distance matrix is generated from the data in the customer coordinate list using the “googlemaps.directions()” function as shown in Figure 2. The distance between all pairs of customer nodes is retrieved and appended into a defined empty matrix “_travel_distance_truck”. The mode of transportation is set to “driving”, hence the distances listed represent actual driving distances between the different locations, in this case in meters.

```
pd.options.mode.chained_assignment = None
from haversine import haversine, Unit
#Retrieve Distance Matrix Drones
def _distance_matrix_drone(_df):

    _travel_distance_drone = np.zeros((len(_df),len(_df)))
    _df['lat-long'] = '0'
    for i in range(len(_df)):
        _df['lat-long'].iloc[i] = (df.lat[i],df.long[i])

    for i in range(len(_df)):
        for j in range(len(_df)):
            # Calculate distances between customer nodes
            _haversine_result = haversine(_df['lat-long'].iloc[i],
                                          _df['lat-long'].iloc[j],
                                          unit='m')

            # Append distances to matrix
            _travel_distance_drone[i][j] = int(_haversine_result)

    return _travel_distance_drone

distance_drone = _distance_matrix_drone(df)

print(distance_drone)
```

```
[[   0.  299.   94.   60.  417.  440.  388.  788.  774.]
 [ 299.    0.  228.  239.  334.  642.  656.  892.  526.]
 [  94.  228.    0.   58.  424.  441.  429.  753.  681.]
 [  60.  239.   58.    0.  382.  474.  439.  803.  722.]
 [ 417.  334.  424.  382.    0.  856.  796. 1176.  811.]
 [ 440.  642.  441.  474.  856.    0.  192.  397.  905.]
 [ 388.  656.  429.  439.  796.  192.    0.  581. 1016.]
 [ 788.  892.  753.  803. 1176.  397.  581.    0.  933.]
 [ 774.  526.  681.  722.  811.  905. 1016.  933.    0.]]
```

*Figure 3 - Jupyter Notebook Generation of Distance Matrix Drone*

The drone distance matrix is generated through a similar setup as for the truck distance matrix, except that the calculation of the distances is now done by using the haversine formula function as shown in Figure 3. Distances between each pair of customer nodes is calculated and appended into an empty matrix. The initial expectation is that the distances for the drone should be shorter than for the truck since the drone does not have to follow a road network, this appears to be true.

### 3.3.2 Visualization of Generated Customer Locations

To visualize the customer locations the nodes are plotted in a plane coordinate system by utilizing functions from the MatPlot library. Figure 4 illustrates the code for retrieving the graph and appending the customer and depot nodes. For each customer node the required demand is included. In the graph the x-axis represents the latitudes, and the y-axis represents the longitudes, customizations can be made to this cell to obtain preferred layout for the output.

```python
import matplotlib.pyplot as plt
plt.figure(figsize=(8,8))
ax=plt.axes()
ax.set_facecolor('aliceblue')
for i in range(customers):
    if i == 0:
        plt.scatter(df.lat[i], df.long[i], marker='s', c='black', s=200)
        plt.text(df.lat[i]-0.00095, df.long[i], "Depot", fontsize=14)
    else:
        plt.scatter(df.lat[i], df.long[i], c='red', s=50)
        plt.text((df.lat[i]+0.0001), df.long[i], ('$q_%d=%d$'%((i,df.demand[i]))),
                 fontsize=14)
```

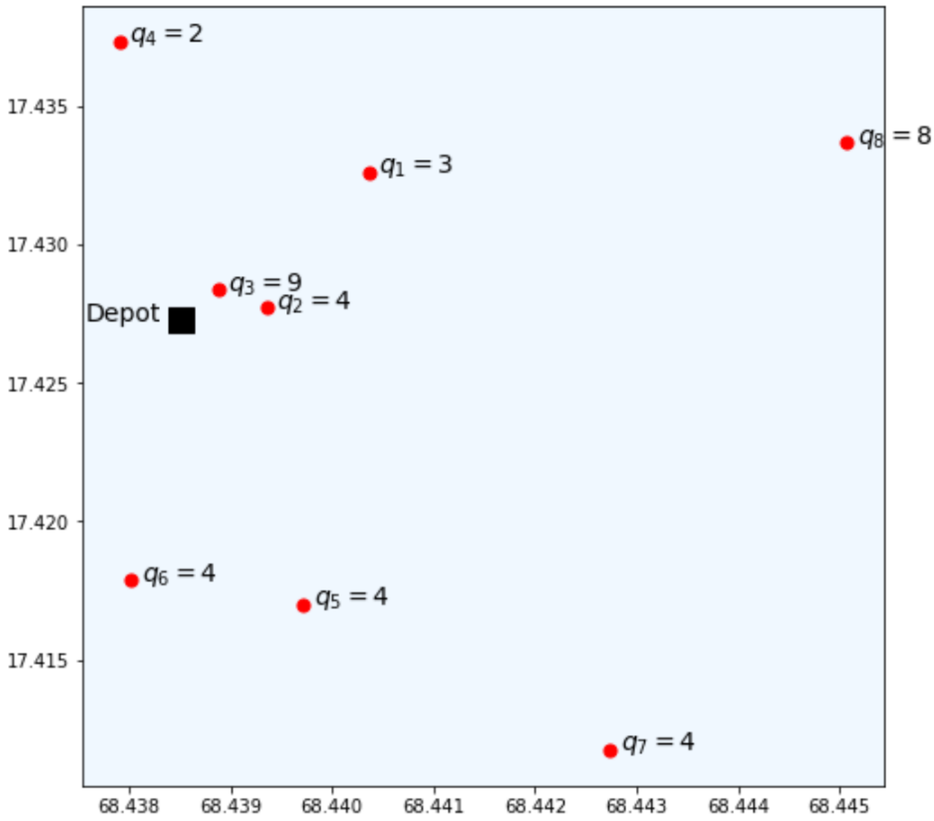


*Figure 4 – Jupyter Notebook Customer Node Plot*

To make the system even more understandable the same customer nodes are plotted in Google Maps by using the "gmaps.figure()" function as shown in Figure 5. Customer nodes are represented as red markers that points out their exact location, and the depot is presented as a black circle. One aspect to be aware of when creating customer locations randomly is that some nodes may be places in a location that the vehicles are unable to access, like for instance in the ocean or on top of a mountain. For the drones, this occurrence will not be an issue, but the trucks are dependent on having road access to the customer location in order to serve it. This will also impact the distance matrix for the truck resulting in inaccurate distances and directions between locations since the driving distance will only be measured up to the point of where the road ends closest to the customer location that is currently out of reach. To prevent this issue the customer locations should be checked in advance to ensure that they are within reach when using randomly generated nodes, this cell is suitable for that purpose.

```
def _customer_plot(_df):

    _customer_locations = []
    for i in range(1,len(_df)):
        _customer_locations.append((_df['lat'].iloc[i],_df['long'].iloc[i]))

    _fig = gmaps.figure()
    _pins = gmaps.marker_layer(_customer_locations)
    _fig.add_layer(_pins)
    _depot_location=[]
    for i in range(0,1):
        _depot_location.append((_df['lat'].iloc[0],_df['long'].iloc[0]))
    _depot = gmaps.symbol_layer(_depot_location,fill_color='black', scale=6)
    _fig.add_layer(_depot)

    return _fig
customer_plt = _customer_plot(df)
customer_plt
```


*Figure 5 - Jupyter Notebook Customer Node Google Maps Plot*

### 3.3.3 Coding the Mathematical Formulation

The formulation presented in section 3.2 is coded into the Jupyter Notebook using PuLP, which creates a Linear Programming-file, or LP-file, that the solver can read. Figure 6 illustrates the first part of the code, where the decision variables for both truck and drone is defines as well as the objective function. Constraint set 1 in the model corresponds to constraint set 2 in the mathematical formulation which ensures that each customer is served once by only one vehicle, and constraint set 2 in the model corresponds to constraint set (3), (4), (5) and (6) in the mathematical formulation which guarantees that each vehicle departs from and returns to the depot.

```
from pulp import *
# Vans-and-Drones Routing Model
for trucks in range(1,trucks+1):

    prob = LpProblem("VRPD", LpMinimize)

    # define decision variables for truck and drone
    xt = [[[LpVariable("xt%s_%s,%s"%(i,j,k), cat="Binary") if i != j else None
            for k in range(trucks)]for j in range(customers)] for i in range(customers)]
    xd = [[[LpVariable("xd%s_%s,%s"%(i,j,kd), cat="Binary") if i != j else None
            for kd in range(drones)]for j in range(customers)] for i in range(customers)]

    # Objective function - Minimize total travel cost
    prob += lpSum((distance_truck[i][j] * xt[i][j][k]) if i != j else 0
                        for k in range(trucks)
                        for j in range(customers)
                        for i in range (customers))
            + lpSum((fixed_cost_truck * xt[0][j][k])
                        for k in range(trucks)
                        for j in range(1,customers))
            + lpSum((distance_drone[i][j] * xd[i][j][kd] if i!=j else 0
                        for kd in range(drones)
                        for j in range(customers)
                        for i in range(customers)))

    # constraint 1 - Customer is only served once by only one vehicle
    for j in range(1, customers):
        prob += lpSum(xt[i][j][k] if i != j else 0
                      for i in range(customers)
                      for k in range(trucks)) + lpSum(xd[i][j][kd] if i != j else 0
                                                      for i in range(customers)
                                                      for kd in range(drones)) == 1

    # Constraint 2 - Vehicle must depart from and return to depot
    for k in range(trucks):
        prob += lpSum(xt[0][j][k] for j in range(1,customers)) == 1
        prob += lpSum(xt[i][0][k] for i in range(1,customers)) == 1

    for kd in range(drones):
        prob += lpSum(xd[0][j][kd] for j in range(1,customers)) == 1
        prob += lpSum(xd[i][0][kd] for i in range(1,customers)) == 1
```

*Figure 6 - Jupyter Notebook Coding the Mathematical Formulation 1*

The remaining code for the mathematical formulation is illustrated in Figure 7, where constraint set 3 corresponds to constraint set (7) in the mathematical formulation which restricts the arcs in and out of the customer node to be used only once by one vehicle. Constraint set 4 ensures that the truck and drones carrying capacity is not exceeded during their route, this corresponds to constraint set (8) and (9) in the mathematical formulation. Lastly, constraint set 5 is defined which prevents subtours in the vehicle routes, like constraint set (10) and (11) in the mathematical formulation. The mathematical formula is now translated into a complete LP-file which for this case is called “VRPDModel”.  Line #76 then calls for the CPLEX solver and the problem status is printed, as well as required truck and drone count, and the objective value from the solution which is the total distance traveled for all vehicles in the system. An additional feature found is the model is the ability to determine the number of vehicles required to perform all the deliveries if the carrying capacity for each vehicle is large enough, and as long as the objective function is not affected negatively by the reduction. If desirable, line #77 and #78 can be used to print the node pairs in the active arcs.

```
43      # Constraint 3 - Only one vehicle enters and leaves each customer location
44      for k in range(trucks):
45          for kd in range(drones):
46              for j in range(customers):
47                  prob += (lpSum(xt[i][j][k] if i != j else 0
48                                 for i in range(customers))
49                            + lpSum(xd[i][j][kd] if i!=j else 0
50                                   for i in range(customers)))
51                            - (lpSum(xt[j][i][k] for i in range(customers))
52                            + lpSum(xd[j][i][kd] for i in range(customers))) == 0
53
54      # Constraint 4 - Demand must not exceed vehicle capacity
55      for k in range(trucks):
56          prob += lpSum(df.demand[j] * xt[i][j][k] if i != j else 0
57                        for i in range(customers) for j in range (1,customers)) <= truck_capacity
58      for kd in range(drones):
59          prob += lpSum(df.demand[j] * xd[i][j][kd] if i != j else 0
60                        for i in range(customers) for j in range (1,customers)) <= drone_capacity
61
62      # Constraint 5 - Subtour elimination
63      subtours = []
64      for i in range(2,customers):
65          subtours += itertools.combinations(range(1,customers), i)
66
67      for s in subtours:
68          prob += lpSum(xt[i][j][k] if i !=j else 0 for i, j in itertools.permutations(s,2)
69                        for k in range(trucks)) <= len(s) - 1
70
71      for s in subtours:
72          prob += lpSum(xd[i][j][kd] if i !=j else 0 for i, j in itertools.permutations(s,2)
73                        for kd in range(drones)) <= len(s) - 1
74
75      if prob.solve(CPLEX_PY()) == 1:
76          prob.writeLP("VRPDModel.lp")
77      #for v in prob.variables():
78          #print(v.name, "=", v.varValue)
79          print('Status:', LpStatus[prob.status])
80          print('Trucks:', trucks)
81          print('Drones:', drones)
82          print('Total distance traveled:', pulp.value(prob.objective))
83          break
```

*Figure 7 - Jupyter Notebook Coding the Mathematical Formulation 2*

The presented problem example contains a small amount of customer nodes located closely together and a small vehicle count with sufficient combined carrying capacity. The solver was then able to find an optimal solution, using branch-and-cut algorithm, with 0% gap between best bound and best integer solution within a short period of time as presented in Figure 8. The solution required all available vehicles, and the objective value was found with a total travel distance of 7095 meters.

```
Version identifier: 20.1.0.0 | 2020-11-11 | 9bedb6d68
CPXPARAM_Read_DataCheck                          1
Tried aggregator 1 time.
MIP Presolve eliminated 371 rows and 80 columns.
MIP Presolve modified 60 coefficients.
Reduced MIP has 156 rows, 136 columns, and 3232 nonzeros.
Reduced MIP has 136 binaries, 0 generals, 0 SOSs, and 0 indicators.
Presolve time = 0.02 sec. (8.34 ticks)
Probing fixed 24 vars, tightened 0 bounds.
Probing time = 0.02 sec. (1.89 ticks)
Tried aggregator 1 time.
MIP Presolve eliminated 9 rows and 24 columns.
Reduced MIP has 147 rows, 112 columns, and 3100 nonzeros.
Reduced MIP has 112 binaries, 0 generals, 0 SOSs, and 0 indicators.
Presolve time = 0.02 sec. (1.61 ticks)
Probing time = 0.02 sec. (1.88 ticks)
Tried aggregator 1 time.
Detecting symmetries...
Reduced MIP has 147 rows, 112 columns, and 3100 nonzeros.
Reduced MIP has 112 binaries, 0 generals, 0 SOSs, and 0 indicators.
Presolve time = 0.02 sec. (1.69 ticks)
Probing time = 0.00 sec. (1.87 ticks)
Clique table members: 971.
MIP emphasis: balance optimality and feasibility.
MIP search method: dynamic search.
Parallel mode: deterministic, using up to 8 threads.
Root relaxation solution time = 0.00 sec. (0.56 ticks)

        Nodes                                         Cuts/
   Node  Left     Objective  IInf  Best Integer    Best Bound    ItCnt     Gap

*     0+    0                        9297.0000      834.0000            91.03%
      0     0     6989.0000    18    9297.0000     6989.0000       39    24.83%
      0     0     7031.3889    19    9297.0000       Cuts: 5       48    24.37%
*     0+    0                        7948.0000     7031.3889            11.53%
*     0     0      integral     0    7098.0000       Cuts: 5       55     0.00%
*     0+    0                        7098.0000     7098.0000             0.00%
      0     0        cutoff          7098.0000     7098.0000       55     0.00%
Elapsed time = 0.19 sec. (27.98 ticks, tree = 0.01 MB, solutions = 3)

Zero-half cuts applied:  2
Lift and project cuts applied:  1
Gomory fractional cuts applied:  2

Root node processing (before b&c):
  Real time             =    0.20 sec. (28.00 ticks)
Parallel b&c, 8 threads:
  Real time             =    0.00 sec. (0.00 ticks)
  Sync time (average)   =    0.00 sec.
  Wait time (average)   =    0.00 sec.
                          ------------
Total (root+branch&cut) =    0.20 sec. (28.00 ticks)
Cplex status= 101
Status: Optimal
Trucks: 2
Drones: 1
Total distance traveled: 7098.0
```

*Figure 8 - Jupyter Notebook CPLEX Solver*

### 3.3.4 Illustration of Results

The results from the solver are presented in a graph from the code shown in Figure 9. Similar to the code in Figure 4, this code creates a graph with the customer nodes, depot node and demands. In addition, the active arcs are plotted, in this case the two truck routes are illustrated with the blue and red routes, while the drone route is a dotted purple line. All customer nodes are visited, and each vehicle starts and ends their route at the depot, without having any subtours. The direction of the routes is denoted by arrowheads. This cell is also suitable for detecting any obvious errors in the system during the modeling phase if the model happens to be incorrect.

```
plt.figure(figsize=(8,8))
ax=plt.axes()
ax.set_facecolor('aliceblue')
for i in range(customers):
    if i == 0:
        plt.scatter(df.lat[i], df.long[i], c='black', marker='s', s=200)
        plt.text(df.lat[i]-0.001, df.long[i], "Depot", fontsize=16)
    else:
        plt.scatter(df.lat[i], df.long[i], c='red', s=100)
        plt.text((df.lat[i]+0.0001), df.long[i], ('$q_%d=%d$'%((i,df.demand[i]))), fontsize=14)
color_list = ["red","blue","green","yellow"]
for k in range(trucks):
    for i in range(customers):
        for j in range(customers):
            if i != j and pulp.value(xt[i][j][k]) == 1:
                plt.arrow(df.lat[i], df.long[i], (df.lat[j] - df.lat[i]), (df.long[j] - df.long[i]),
                          width=0.00001, head_width=0.0002, color=color_list[k], length_includes_head=True)

color_list_d = ["purple","white"]
for kd in range(drones):
    for i in range(customers):
        for j in range(customers):
            if i != j and pulp.value(xd[i][j][kd]) == 1:
                plt.arrow(df.lat[i], df.long[i], (df.lat[j] - df.lat[i]), (df.long[j] - df.long[i]),
                          width=0.00001,length_includes_head=True, head_width=0.0002, color=color_list_d[kd],
                          linestyle='dotted')
```

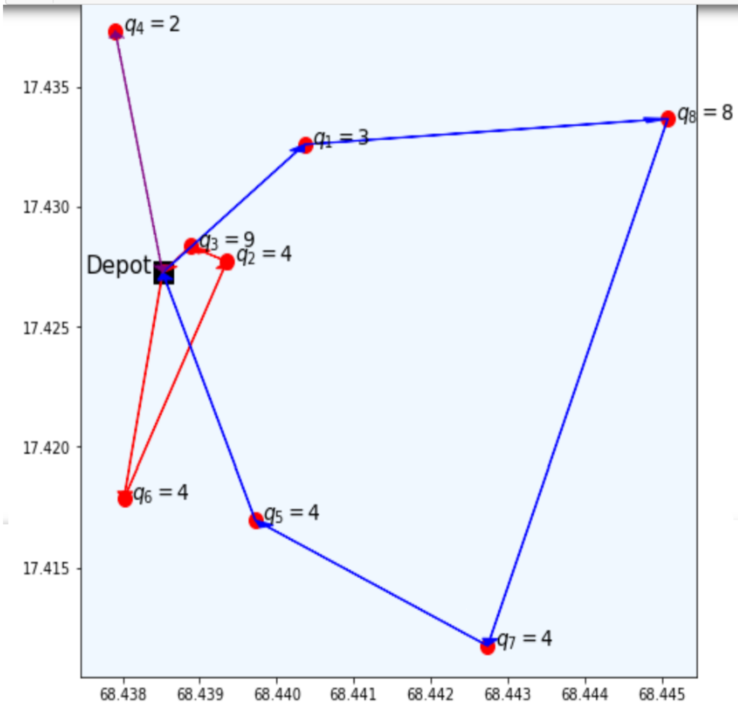


*Figure 9 - Jupyter Notebook Plotting Results*

The last part of the model is to illustrate the results in Google Maps, now the vehicle routes are included, the code for this is shown in Figure 10. A google maps figure is created and the customer and depot locations is plotted. Layers are added to plot the vehicle routes, which are illustrated with different colors for each active vehicle.

```
fig = gmaps.figure()
layer_truck = []
color_truck = ["red","blue","green","black"]

for k in range(trucks):
    for i in range(customers):
        for j in range(customers):
            if i != j and pulp.value(xt[i][j][k]) == 1:
                layer_truck.append(gmaps.directions.Directions(
                    (df.lat[i],df.long[i]),
                    (df.lat[j],df.long[j]),
                    mode='car',stroke_color=color_truck[k],stroke_opacity=1.0, stroke_weight=5.0))

for i in range(len(layer_truck)):
    fig.add_layer(layer_truck[i])

layer_drone = []
color_drone = ["purple","yellow"]
for kd in range(drones):
    for i in range(customers):
        for j in range(customers):
            if i != j and pulp.value(xd[i][j][kd]) == 1:
                drawing=(gmaps.Line(start=(df.lat[i], df.long[i]), end=(df.lat[j], df.long[j]),
                             stroke_color=color_drone[kd],stroke_opacity=1.0, stroke_weight=5.0))
                mark=(gmaps.Marker((df.lat[i],df.long[i])))

                layer_drone_draw = gmaps.drawing_layer(features=[mark, drawing])
                layer_drone.append(layer_drone_draw)
for i in range(len(layer_drone)):
    fig.add_layer(layer_drone[i])

depot_location=[]
for i in range(0,1):
    depot_location.append((df['lat'].iloc[0],df['long'].iloc[0]))
    depot = gmaps.symbol_layer(depot_location,fill_color='black', stroke_color='black', scale=6)
    fig.add_layer(depot)


fig
```

*Figure 10 - Jupyter Notebook Code for Plotting Results in Google Maps*

Figure 11 shows the results plotted in a map, where the truck routes are represented in blue and red, and the drone route is purple. Each route is connected and starts and ends at the depot, which is located at the black circle. It is clearly shown that the trucks follow the road network and visits all customers on its route. The drone is not restricted to roads and travel in a straight line between the depot and the customer location it is serving. As intended the model succeeds to find optimal routes based on the given inputs while meeting all the requirements subjected by the formulation constraints.

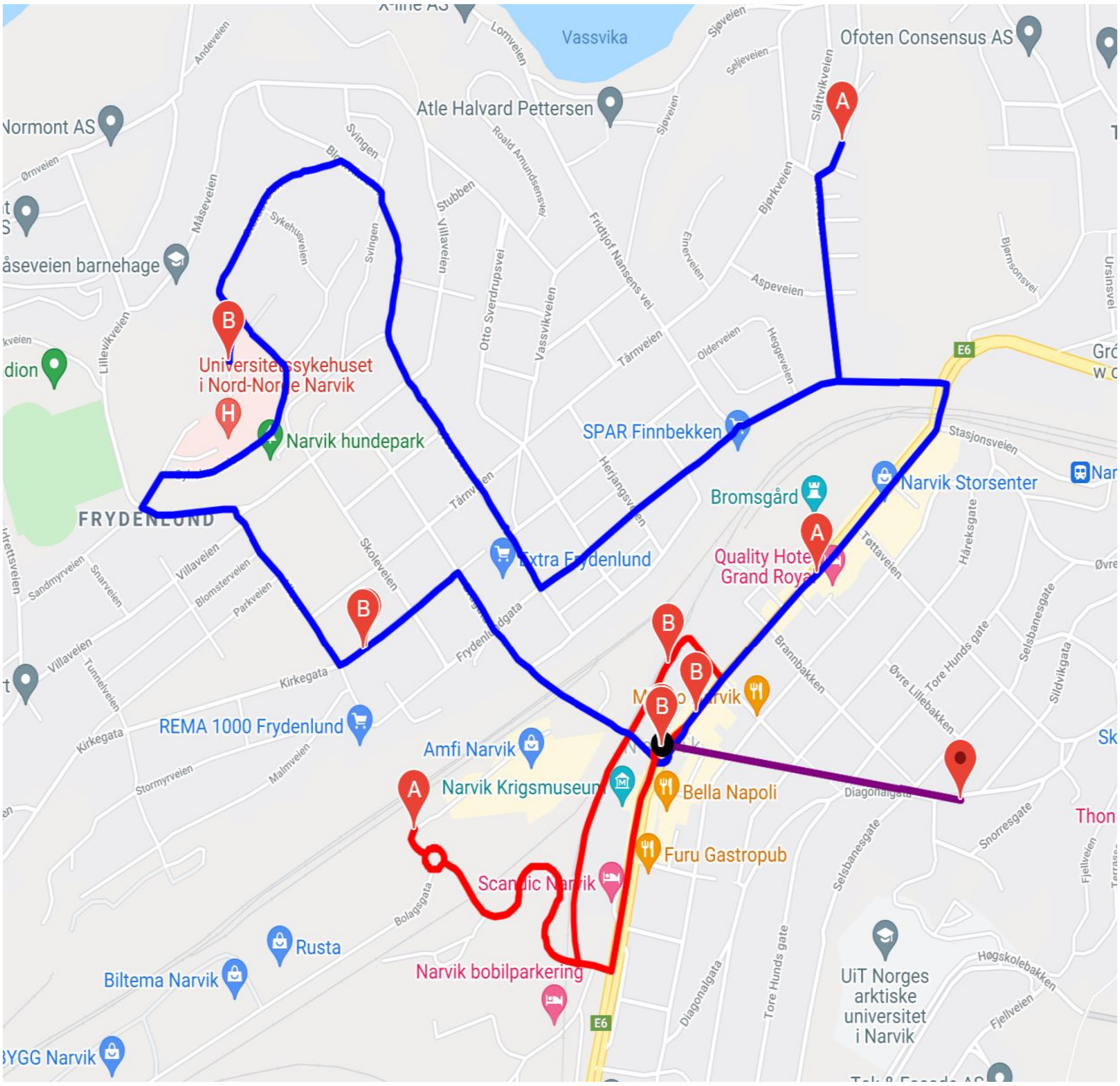


*Figure 11 – Jupyter Notebook Results plotted in Google Maps*

# 4 Verification and Validation Phase

In this section the proposed VRPD model is tested through a verification and validation phase. To verify the VRPD model a small comparison study is executed where the model is set up against Google OR-Tools CVRP model, which is assumed to be an accurate representation of such a system [32]. The instances used for the study are listed in Table 2. Four different settings are tested for both models and compared based on runtime and optimal objective values. Google uses its own solver GOLP, while our model uses CPLEX solver. The customer locations are randomly chosen as well as the customer demands.

For the validation phase a small-scale case study is carried out, where the vans-and-drones model is compared with a truck-only configuration. The two models share the same base layout, so for the CVRP tests the drone parameters are simply ignored. The case study is in two parts with different test instances, the set of customer locations are predefined, and the objective is to effectively deliver vaccines in Ofoten region. In this part emission is also tested as well as total travel cost.

## 4.1 Comparison of VRPD model and Google OR-Tools CVRP

In this section the results from the verification phase are presented, where the proposed VRPD model is tested against Google OR-Tools VRPD model. The purpose for this test is to compare the computational effectiveness and efficiency of the models, therefore the parameters are randomly generated and equal for both systems, and the carrying capacity is the same for both truck and drone. Two different customer count configurations are tested, first with 8 then with 10 customers. Since the drones are given the same carrying capacity as the trucks it is expected that the results from the VRPD model should be equal to the results from the CVRP model. As shown in Table 2 each customer count configuration has two test instances each. For 8 customers, denoted as “N8”, the first test instance is “K3-Q10 / K3-Kd1-Qk10-Qkd10”, where “K3-Q10” belong to the CVRP configuration and represents three trucks, “K3”, with maximum carrying capacity of 10 units, “Q10”. The second part of the instance notation belong to the VRPD configuration where “K3” represents three trucks, “Kd1” represents one drone, “Qk10” denotes a truck carrying capacity of 10 units and lastly “Qkd10” denotes the drone carrying capacity which is also 10 units. The results show that the deviation of objective values between the to models are 0% for all instances, which indicates that the VRPD model is able to solve the routing problems accurately, with somewhat longer runtimes but still assumed to be acceptable. It may be that Google OR-Tools uses more efficient heuristic methods which lead to a better performance than the exact solution in CPLEX. However, this is not investigated further since the runtimes for the VRPD is considered acceptable.

| | | **GOLP** | | **CPLEX** | | **Deviation of objective values (%)** |
|---|---|---|---|---|---|---|
| | | **CVRP** | | **VRPD** | | |
| | **Instance** | **Objective value** | **Runtime sec** | **Objective value** | **Runtime sec** | |
| **N8** | K3-Qk10 / K2-Kd1-Qk10-Qkd10 | 4404 | 1.2 | 4404 | 2 | **0** |
| | K4-Qk8 / K3-Kd1-Qk8-Qkd8 | 4108 | 2 | 4108 | 7.2 | **0** |
| **N10** | K3-Qk15 / K2-Kd1-Qk15-Qkd15 | 3880 | 2 | 3880 | 10.1 | **0** |
| | K4-Qk10 / K3-Kd1-Qk10-Qkd10 | 4268 | 0.5 | 4268 | 12.5 | **0** |

*Table 2 - Comparison Study Google OR-Tools Results*

## 4.2 Case Study

This section contains a small-scale case study in two parts carried out as a numerical experiment. In the first part the proposed Vans-and-Drones routing model is applied to a real-world scenario in order to prove the validity of the model. The customer locations are predefined, the cost parameter uses actual driving distances for the trucks, obtained through Google Directions API with avoidance of ferries in the truck routes, and the drones routes are obtained using the haversine function. The results are compared to a truck-only CVRP model.

In the second part of the case study the purpose is to test the effectiveness of the proposed Vans-and-Drones routing model against a truck-only system in a vaccine distribution scenario for a sparsely populated delivery area. Delivery by drone would presumable be ideal for vaccine distribution, since the vaccine bottles are small in size and several dozes can be fitted into one unit of parcel for the drone to carry. This could prevent excessive use of traditional diesel trucks in the system and reduce the emission and costs of employment associated with this type of transportation in the delivery operation.

A total of 13 geographical locations are serving as the set of customer nodes used in the case study including the depot, as shown in Table 3. The delivery area is located in the northern region of Norway, and the customer locations are chosen based on the highest populated areas in Ofoten region [33]. Two additional locations are included which are not located in this region but nearby. These are the customer locations number 11 and 12 and are added in order to analyze how a wider spread of the customer nodes in the network would affect the results, and they are only considered in part two of the case study. Each location is denoted with latitude and longitude coordinates and Narvik serves as the depot for all test instances and is marked as location number 0.

| Number | Location | Latitude | Longitude |
|---|---|---|---|
| **0** | Narvik (D) | 68.438575 | 17.42726 |
| **1** | Kjøpsvik | 68.09695 | 16.37415 |
| **2** | Liland | 68.48083 | 16.8861 |
| **3** | Lødingen | 68.41415 | 15.99439 |
| **4** | Håkvik | 68.40514 | 17.3085 |
| **5** | Bjerkvik | 68.54929 | 17.5575 |
| **6** | Beisfjord | 68.37617 | 17.59601 |
| **7** | Ballangen | 68.34293 | 16.83141 |
| **8** | Bogen | 68.52244 | 17.00573 |
| **9** | Kjeldebotn | 68.40556 | 16.66178 |
| **10** | Fjelldal | 68.55605 | 16.52621 |
| **11** | Abisko | 68.34954 | 18.83124 |
| **12** | Setermoen | 68.86100 | 18.34857 |

*Table 3 - Customer Locations*

For this case study the features for the drones, related to travel speed and range, are based on the drone “Robin XL” manufactured by a Canadian company called Drone Delivery Canada [34]. Hence, the drone travel speed is set to 105 km/h and the range is 60 km for all drones in the delivery system. The longest travel distance from the depot that the drone will experience in this case study is approximately 60 km one way, so for the drone to be able to return to the depot after delivering the parcel is it assumed that there is recharging facilities for the drone at the farthest located customer nodes. For the trucks, the case study considers diesel vans where the range is assumed to be unlimited, since it is assumed that there are available gas stations along the roads. The truck travel speed is set to 80 km/h since this is the general speed limit in Norway outside densely populated areas [35].

The emission parameters considered in the case study is 0.079 kg $CO_2$-eq for each km traveled by a drone, based on the results obtained in the works by Koiwanit [5] and Goodchild and Toy [36]. The emission level for the truck is based on data from Statistics Norway, where the average emission level for a van is set to 0.1754 kg $CO_2$-eq per km traveled [37].

### 4.2.1 Case Study Part 1 – Validation

In the first part of the case study the proposed VRPD model is applied to a real-world scenario with the purpose of delivering vaccines to different locations in Ofoten region. The VRPD model is tested with 8 and 10 customers, each with three similar instances. For each instance, a CVRP and a VRPD configuration is listed, where the total vehicle counts are the same but for the VRPD one truck is replaced with a drone. For the drones, the carrying capacity is set to one unit, and for the trucks the capacity varies in order to meet the total demand of the system while trying to assign roughly the same number of customers to each truck. Table 4 presents the results from the numerical experiment, where we can see that that the Vans-and-Drones delivery system manages to outperform the CRPV for all instances. The average improvement is found to be 7.22% by replacing one truck with a drone.

| | Instance | CPLEX | | | | Improvement of objective value (%) |
|---|---|---|---|---|---|---|
| | | CVRP | | VRPD | | |
| | | Objective value | Runtime sec | Objective value | Runtime sec | |
| N8 | K3-Qk15 / K2-Kd1-Qk20-Qkd1 | 509758 | 0.44 | 475501 | 0.30 | **6.72** |
| | K4-Qk10 / K3-Kd1-Qk12-Qkd1 | 579851 | 0.52 | 527928 | 0.41 | **8.95** |
| | K5-Qk8 / K4-Kd1Qk12-Qd1 | 618729 | 2 | 584472 | 1.9 | **5.54** |
| N10 | K3-Qk20 / K2-Kd1-Qk20-Qkd1 | 549751 | 32 | 531484 | 2.5 | **3.32** |
| | K4-Qk15 / K3-Kd1-Qk20-Qkd1 | 619844 | 58 | 570362 | 28 | **7.98** |
| | K5-Qk15 / K4-Kd1-Qk10-Qkd1 | 731579 | 106 | 652738 | 19 | **10.78** |
| | **Average** | | | | | **7.22** |

*Table 4 – Comparison Results Vans-and-Drones and Truck-Only Systems*

A sensitivity analysis is also carried out to see how changes in the vehicle carrying capacity would affect the objective values when the vehicle count remained unchanged. The results from this analysis are listed in Table 5, and they show that when the carrying capacity for each truck gets higher and customer demand remains the same, the objective value for the CVRP system decreases, whilst for the VRPD system it remains the same. The increased vehicle carrying capacity would allow the trucks to serve more customer in its route, resulting in a reduced total distance traveled for the whole system. The customer location layout for case study part 1 is the same as for part 2, see Figure 12, and it can be observed that there are clearly two main legs out from the depot location where the customer locations and demands are somewhat evenly distributed. The test instances for the CVRP hold three trucks and with two main legs protruding from the depot would excessive use of the third vehicle be somewhat redundant, and actually adding cost to the system. While for the vans-and-drones system the two trucks hold in the test instance are already doing this job, and an increase of the carrying capacity for the truck would not change the objective value, unless the carrying capacity is increased to such an extent that one truck could serve the system alone.

| | | CPLEX | | | | Improvement of objective value (%) |
|---|---|---|---|---|---|---|
| | | CVRP | | VRPD | | |
| | Instance | Objective value | Runtime sec | Objective value | Runtime sec | |
| **N10** | K3-Q17 / K2-Kd1-Qk25-Qkd1 | 670879 | 284 | 531484 | 1.15 | **20.78** |
| | K3-Q20 / K2-Kd1-Qk30-Qkd1 | 549751 | 32 | 531484 | 0.86 | **3.32** |
| | K3-Qk24 / K2-Kd1-Qk45-Qkd1 | 517961 | 2 | 531484 | 1.53 | **(-2.61)** |

*Table 5 - Sensitivity Analysis Results*

### 4.2.2 Case Study Part 2

In this section the model is tested even further by increasing the number of drones in the VRPD system in three different settings. For the case study the distances requested for the trucks from Google Direction is set to avoid ferries in the driving route. In setting 1 and 2 the customer locations are the same, as shown in Figure 12, and are equal to the customer location layout used in part 1 of the case study. While in setting 3, customer locations "Beisfjord" and "Håkvik" are replaced with "Abisko" and "Setermoen", as shown in Figure 13. For all instances, each drone is limited to serve only one customer.

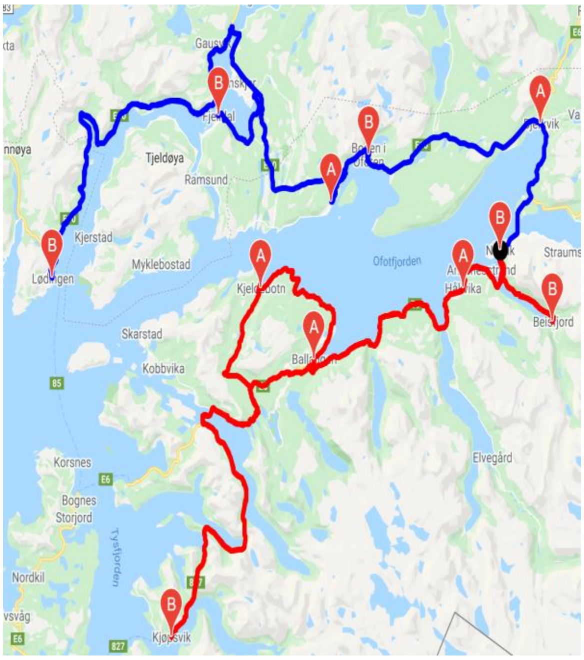


*Figure 12 - Customer Locations for Setting 1 and 2*

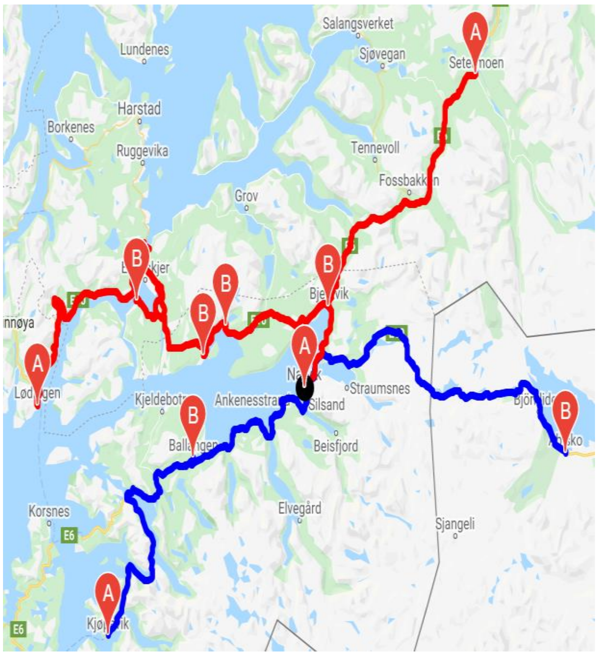


*Figure 13 - Customer Locations for Setting 3*

In setting 1 the CVRP system with 2 trucks gave an optimal objective value of 510873 m, see Table 6, which is an improvement compared to the test results in Table 4 where the objective value for test instance K3-Qk20 with 3 trucks was 549751 m. The method in setting 1 is to let drones serve the customer nodes that are furthest from the depot, which is done continuously through test instances (2), (3), (4) and (5), to see how this would affect the objective value compared to the CVRP system. The results show that neither of the test instances for VRPD in setting 1 improved the objective value compared to the truck-only system.

As mentioned earlier, the layout of the customer node locations in Figure 12 clearly shows two main legs protruding from the depot, with customer nodes located along each one. This could be favorable for trucks since it prevents too many detours from the main truck route and may be the reason why assigning drones to serve the further nodes fail to improve the objective value. In test instance (6) one whole leg of customer nodes is assigned to drones instead of only the furthest nodes. As shown in Table 6 the result from this test instance also shows an increase in the objective value compared to instance (1).

Since setting 1 and 2 both failed to improve the objective value, a third setting is created where the customer locations are scattered more evenly around the depot, as shown in Figure 13. The objective value for the CVRP in this layout is 766531m, see instance (7) in Table 6, and instance (8) and (9) gets tested against this value. Introducing drones in this layout results in an improvement of the objective value of 1,32% and 4,71% for one and two included drones, respectively. The reason for this is probably that serving the two new customer nodes would be a considerable detour for the trucks since the system now has four main legs and only 2 trucks. Another aspect to consider is that the road networks connecting the customer nodes in this layout are somewhat geographically straight to begin with, and since the only cost parameter considered in the VRPD model is travel distance, will switching to drones may not have a significant effect on the these results. However, for setting 3 the objective value is indeed improved when employing drones.

| | **Instance** | **CPLEX** | | **Improvement from CVRP Objective Value(%)** |
|---|---|---|---|---|
| | | **CVRP** | **VRPD** | |
| | | **Objective value** | **Objective value** | |
| **Setting 1** | (1) K2 CVRP | 510873 | | |
| | (2) K2-Kd1 | | 531484 | **Neg** |
| | (3) K2-Kd2 | | 520367 | **Neg** |
| | (4) K2-Kd3 | | 555093 | **Neg** |
| | (5) K2-Kd4 | | 577229 | **Neg** |
| **Setting 2** | (K2) | (510873) | | |
| | (6) K1-Kd5 | | 537702 | **Neg** |
| **Setting 3** | (7) K2 CVRP | 766531 | | |
| | (8) K2-Kd1 | | 756410 | **1.32** |
| | (9) K2-Kd2 | | 730455 | **4.71** |

*Table 6 – Results from Case Study Part 2*

The results of the total travel distances for the tested instances are converted into total travel time to see the changes in travel time consumption for both systems. The average speed for the trucks is set to 80 km/h, and the average drone travel speed is set to 105 km/h. The total travel times for the instances are listed in Table 7, and the results show a decrease in travel time for all instances of the VRPD model compared to the CVRP model. Hence, even though the total travel distances in the VRPD system for setting 1 and 2 in Table 6 is not improved compared to the CVRP system, the total delivery time is improved. On average the delivery time is reduced by 6.77% when employing the vans-and-drones system. These results will have an impact on the vaccine delivery operation and making it more efficient by providing overall faster deliveries.

| Instance | Drone Distance Traveled (km) | Truck Distance Traveled (km) | Total delivery time (hours) | Decrease in Total Delivery time (%) |
|---|---|---|---|---|
| **(1) CVRP** | | 510.873 | 6.4 | |
| **(2)** | 115.288 | 416.196 | 6.3 | 1.56 |
| **(3)** | 232.58 | 287.787 | 5.8 | 9.38 |
| **(4)** | 310.536 | 244.557 | 6 | 6.25 |
| **(5)** | 373.57 | 203.659 | 6.1 | 4.69 |
| **(6)** | 263.376 | 274.326 | 5.9 | 7.81 |
| **(7) CVRP** | | 766.531 | 9.6 | |
| **(8)** | 116.66 | 639.75 | 9.1 | 5.21 |
| **(9)** | 236.612 | 493.843 | 8.4 | 12.5 |
| **Average** | | | | **6.77** |

*Table 7 - Travel Time Analysis*

The cost savings related to employment of truck drivers is analyzed based on an average monthly salary of 39 130 NOK [38]. The hourly pay would then be approximately 260 NOK per hour assuming there is 37.5 working hours per week. The total cost for employing the trucks in each test instance is listed in Table 8. The results show an average reduction of 39.4 % for the truck employment costs when drones are also included in the delivery system.

| Instance | Truck Distance Traveled (km) | Total truck delivery time (hours) | Total truck employment cost (NOK) | Reduction in truck employment cost (%) |
|---|---|---|---|---|
| **(1) CVRP** | 510.873 | 6.4 | 1664 | |
| **(2)** | 416.196 | 5.2 | 1352 | 18.75 |
| **(3)** | 287.787 | 3.6 | 936 | 43.75 |
| **(4)** | 244.557 | 3.1 | 806 | 51.56 |
| **(5)** | 203.659 | 2.5 | 620 | 62.74 |
| **(6)** | 274.326 | 3.4 | 884 | 46.88 |
| **(7) CVRP** | 766.531 | 9.6 | 2496 | |
| **(8)** | 639.75 | 8 | 2080 | 16.67 |
| **(9)** | 493.843 | 6.2 | 1612 | 35.42 |
| **Average** | | | | **39.4** |

*Table 8 - Truck Employment Cost Analysis*

In addition to the total distance traveled and total travel time the emission impact is analyzed. For the trucks, the average emission level is set to 0.1754 kg $CO_2$-eq per km traveled, and for the drone it is set to 0.079 kg $CO_2$-eq per km traveled. These are the parameters used for the calculation of the emission for both systems. As shown in Table 9 the emission levels are improved for all the VRPD instances compared to the truck-only system, with an average reduction in $Co_2$ emission of 19.23%. These results are similar to the results obtained in Stolaroff, et al. [39] where they found an average reduction of around 20-30%.

| Instance | Drone Distance Traveled (km) | Truck Distance Traveled (km) | Total $Co_2$ emission (kg $CO_2$-eq) | Decrease in $Co_2$ emission compared to CRVP (%) |
|---|---|---|---|---|
| **(1) CVRP** | | 510.873 | 89.61 | |
| **(2)** | 115.288 | 416.196 | 82.11 | 5.02 |
| **(3)** | 232.58 | 287.787 | 68.85 | 23.17 |
| **(4)** | 310.536 | 244.557 | 67.43 | 24.75 |
| **(5)** | 373.57 | 203.659 | 65.23 | 27.2 |
| **(6)** | 263.376 | 274.326 | 68.92 | 23.09 |
| **(7) CVRP** | | 766.531 | 134.45 | |
| **(8)** | 116.66 | 639.75 | 121.43 | 9.69 |
| **(9)** | 236.612 | 493.843 | 105.31 | 21.67 |
| **Average** | | | | **19.23** |

*Table 9 - Emission Analysis*

The test instances (2), (3), (4) and (5) for setting 1 are illustrated in Figure 14, Figure 15, Figure 16 and Figure 17, and for setting 2 test instance (6) is illustrated in Figure 18.

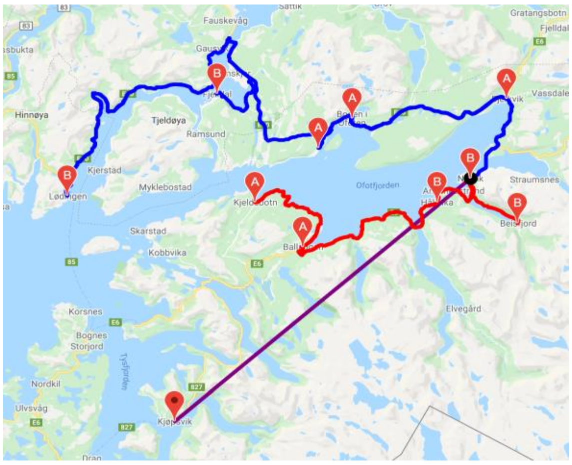


*Figure 14 - Test Instance (2)*

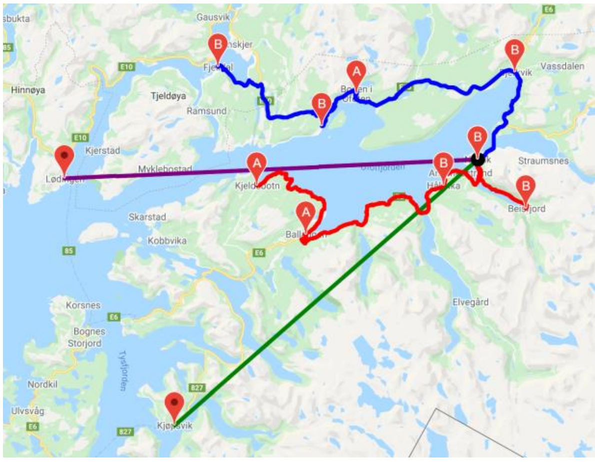


*Figure 15 - Test Instance (3)*

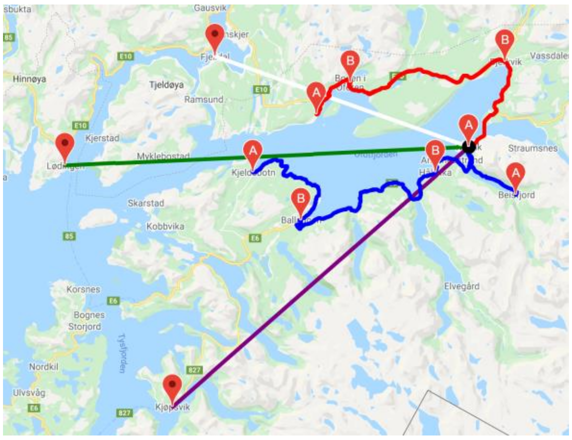


*Figure 16 - Test Instance (4)*

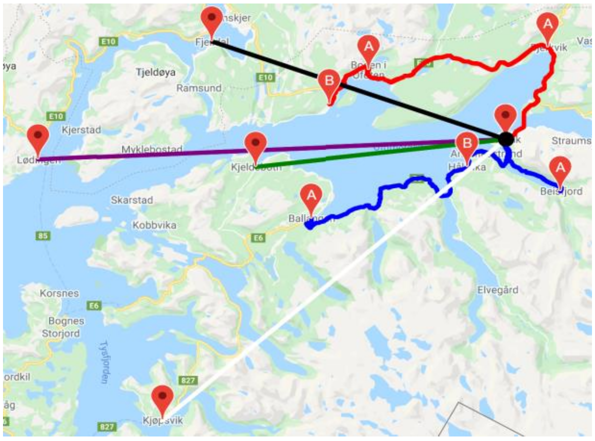


*Figure 17 - Test Instance (5)*

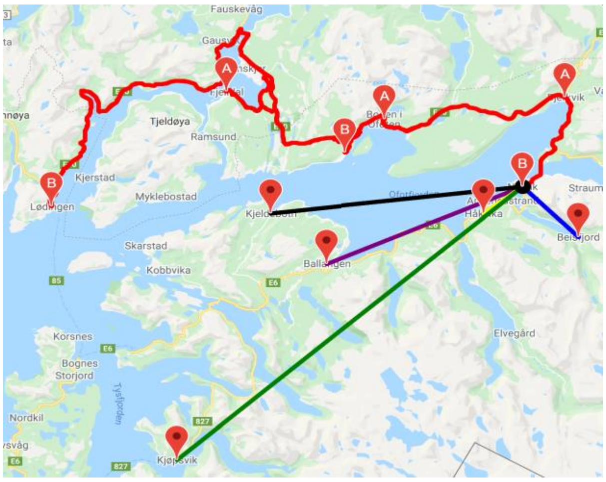


*Figure 18 - Test Instance (6)*

For setting 1 the two truck routes are plotted in red and blue, while the colors for the drone routes varies. In setting 2 only one truck is employed, which is plotted in red. Each customer location is represented with a red marker and the depot location is plotted with a black circle.

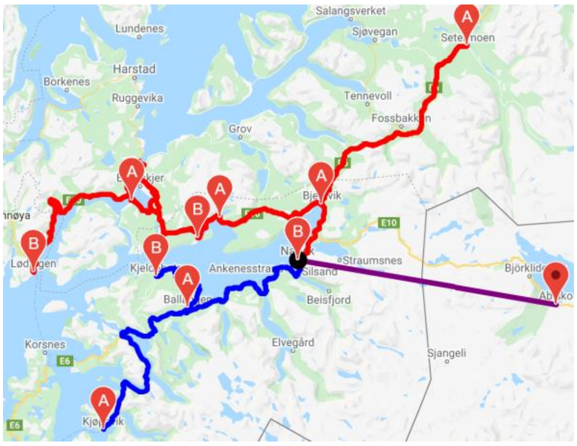


*Figure 19 - Test Instance (8)*

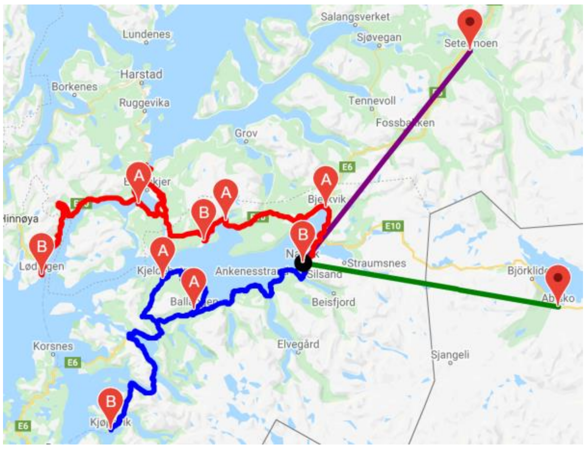


*Figure 20 - Test Instance (9)*

For Figure 13 and Figure 20 the route and marker for customer node “Kjeldebotn” is partially missing or not present, this is due to an issue with the Google Maps API request quota during the plotting causing this layer to be left out. However, the route is included in the calculation of the objective value and is therefore only a shortcoming in the map illustration.

# 5 Discussion

The proposed VRPD model derived in Jupyter Notebook works well for its intended use in a Vans-and-Drones System for Last-Mile Delivery. The model successfully went through a verification and validation phase where no major functionality issues were discovered, and it successfully managed to represent the delivery system for the tested instances. However, the test instances described in this report are small with a low number of customer locations, this was done in order to get the optimal solution using exact solving methods. It would therefore be favorable to test the model further with larger test instances using heuristic solving methods and add a grater number of customer locations and employ more vehicles.

The purpose of the case study was to examine the effectiveness related to delivery time and cost of a delivery system with integrated drones compared to a truck-only system in the scenario of vaccine distribution in a sparsely populated area with a small amount of customers. The analyzed factors are travel distance, travel time, delivery cost of the system and lastly emission impact, where the most current results are the three latter. The results indicated that there would be an overall saving of almost 40% of total employment cost when including drones into the system. In addition, the vans-and-drones system would improve the effectiveness of the delivery time to the customer by almost 7%. If the customer network was located in a more urban environment with more traffic this number would presumably get even higher, since the drones are not dependent one road networks and therefore not affected by traffic. From the literature review it is learned that the main disadvantages with drone delivery is the drones limited carrying capacity and range related to battery consumption. Since the vaccines that gets delivered are small in size and weight, the drone would presumably be able to deliver several vaccine doses in one parcel unit. Hence, the utilization of drones seems suitable for this purpose and together with this the results obtained in the case study also indicates substantial savings in both employment cost and time-consumption. The last important factor to consider is the emission level, and the results indicate an average reduction in $CO_2$-emission for the whole system of almost 20% when including drones. A reduction in emission was obtained for all instances of the VRPD system, which also strengthens the perception that adding drones would be favorable for the delivery process.

Another observation obtained from the case study is that the task allocation for the drones in the delivery system could have an impact on the total distance traveled for all vehicles when considering the customer network configuration. For some of the test instances the total travel distance would actually increase when including drones, this occurred when the drones was allocated the customer nodes furthest from the depot. For these cases, the customer nodes were placed along two main routes out from the depot, and with two trucks and one drone it resulted in an increased total traveling distance when allocating the further customer nodes to the drone. These results could indicate that it is favorable to make the task allocation in a way so that each vehicle travels in different directions from the depot, or letting the drones serve the customers located close to the depot that would in other ways be a considerable detour for the truck. However, despite the increase in travel distance the cost and time-consumption were improved for these instances as well. This was a small-scale case study, so the task allocations were not tested to grate extents, and it would be advantageous to study this further in future work.

In the case of vaccine distribution during a pandemic, which is the current situation, another advantage with employing drones in the delivery operation is that drones perform contactless deliveries with no human-to-human interaction that can lead to further spreading of the virus. Deliveries performed by trucks would require a worker to hand over the parcel, which would increase the risk of transmitting the virus to the customer. In addition, drones could provide faster deliveries to rural areas with poor road infrastructures since they are not dependent on road networks unlike the trucks. Both of these factors would be interesting to study in future work.

# 6 Conclusion

The main contributions of this project are;

- The development of a Vans-and-Drones System for Last-Mile Delivery based on the traditional CVRP.
- Coding, and verification and validation of the model using Jupyter Notebook with Python programming language and CPLEX solver.
- Most importantly, to provide insight into the advantages of using drones in last-mile delivery of vaccines in a sparsely populated area in Ofoten compared to truck-only delivery systems, considering delivery cost, time-consumption and emission impact.

Based on the results from the verification and validation phase, it can be concluded that the proposed VRPD model represents a vans-and-drones system for last-mile delivery in a satisfactory manner. The model manages to produce reasonable results for task allocation and route scheduling for the routing problem with the given parameters.

The results derived in the case study shows that integrating drones into the last-mile delivery system for vaccine distribution in a sparsely populated area, like Ofoten region, could improve the effectiveness of the delivery system considering total delivery time and employment cost compared to truck-only delivery systems. The pollution level from the delivery operations would also be affected positively when employing drones, resulting in an overall reduction in CO2-emission for the whole system.

# 7 Future Work

It would be advantageous to perform a more detailed study on how the task allocation for the drones is affected by the customer network configuration. Preferably a larger study with several different customer location layouts and larger test instances. Similarly, it would be preferable to test the proposed VRPD model even further by including a higher number of customers and employing more vehicles in the system. Two other factors that would be interesting to study further are the drones advantages of contactless delivery, and how this advantage would impact the situation concerning spreading of the ongoing virus, and how the drone could contribute to more effective delivery of vaccines to rural areas with poor infrastructure.

Other factors should also be considered in the model development phase, weather condition could impact the accessibility of the drones. In arctic areas, like the test area in the case study, the vehicles would be exposed to cold weather with icing and snow during wintertime, and it should be investigated how these conditions would impact the drones performance in the delivery operation. In addition the drone range should be added as a capacity constraint to assure that the drone manages to serve the customer and return to the depot before its battery capacity is exceeded.

# References


[1] M. Salama and S. Srinivas, "Joint optimization of customer location clustering and drone-based routing for last-mile deliveries," *Transportation Research Part C: Emerging Technologies,* Article vol. 114, pp. 620-642, 2020, doi: 10.1016/j.trc.2020.01.019.

[2] S. H. Chung, B. Sah, and J. Lee, "Optimization for drone and drone-truck combined operations: A review of the state of the art and future directions," *Computers & Operations Research*, vol. 123, p. 105004, 2020/11/01/ 2020, doi: https://doi.org/10.1016/j.cor.2020.105004.

[3] P. Kitjacharoenchai and S. Lee, "Vehicle routing problem with drones for last mile delivery," in *Procedia Manufacturing*, 2019, vol. 39, pp. 314-324, doi: 10.1016/j.promfg.2020.01.338. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85082772758&doi=10.1016%2fj.promfg.2020.01.338&partnerID=40&md5=30f6afe73b76d1e8a8e30a9ac383be9a

[4] G. Macrina, L. Di Puglia Pugliese, F. Guerriero, and G. Laporte, "Drone-aided routing: A literature review," *Transportation Research Part C: Emerging Technologies,* Article vol. 120, 2020, Art no. 102762, doi: 10.1016/j.trc.2020.102762.

[5] J. Koiwanit, "Analysis of environmental impacts of drone delivery on an online shopping system," *Advances in Climate Change Research*, vol. 9, no. 3, pp. 201-207, 2018/09/01/ 2018, doi: https://doi.org/10.1016/j.accre.2018.09.001.

[6] M. Elsayed and M. Mohamed, "The impact of airspace regulations on unmanned aerial vehicles in last-mile operation," *Transportation Research Part D: Transport and Environment*, vol. 87, p. 102480, 2020/10/01/ 2020, doi: https://doi.org/10.1016/j.trd.2020.102480.

[7] K. Nonami and F. Kendoul, *Autonomous Flying Robots - Unmanned Aerial Vehicles and Micro Aerial Vehicles*. New York: Springer, 2010.

[8] Amazon. "Amazon Prime Air." Amazon. https://www.amazon.com/Amazon-Prime-Air/b?ie=UTF8&node=8037720011 (accessed 2021).

[9] G. Singhal, B. Bansod, and L. Mathew, *Unmanned Aerial Vehicle Classification, Applications and Challenges: A Review*. 2018.

[10] E. J. U. Hernández, J. A. S. Martínez, and J. A. M. Saucedo, "Optimization of the distribution network using an emerging technology," *Applied Sciences (Switzerland),* Article vol. 10, no. 3, 2020, Art no. 857, doi: 10.3390/app10030857.

[11] Q. M. Ha, Y. Deville, Q. D. Pham, and M. H. Hà, "On the min-cost Traveling Salesman Problem with Drone," *Transportation Research Part C: Emerging Technologies,* Article vol. 86, pp. 597-621, 2018, doi: 10.1016/j.trc.2017.11.015.

[12] C. Murray and A. Chu, "The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery," *Transportation Research Part C: Emerging Technologies,* vol. 54, 05/31 2015, doi: 10.1016/j.trc.2015.03.005.

[13] M. Moshref-Javadi, A. Hemmati, and M. Winkenbach, "A truck and drones model for last-mile delivery: A mathematical model and heuristic approach," *Applied Mathematical Modelling,* Article vol. 80, pp. 290-318, 2020, doi: 10.1016/j.apm.2019.11.020.

[14] G. C. Crişan and E. Nechita, "On a cooperative truck-and-drone delivery system," in *Procedia Computer Science*, 2019, vol. 159, pp. 38-47, doi: 10.1016/j.procs.2019.09.158. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-

85076259202&doi=10.1016%2fj.procs.2019.09.158&partnerID=40&md5=ad6b8100 1fd8e1e9b3828583c424d29d

[15] H. Y. Jeong, B. D. Song, and S. Lee, "Truck-drone hybrid delivery routing: Payload-energy dependency and No-Fly zones," *International Journal of Production Economics,* Article vol. 214, pp. 220-233, 2019, doi: 10.1016/j.ijpe.2019.01.010.

[16] M. Dell'Amico, R. Montemanni, and S. Novellani, "Drone-assisted deliveries: new formulations for the flying sidekick traveling salesman problem," *Optimization Letters,* Article 2019, doi: 10.1007/s11590-019-01492-z.

[17] R. Raj and C. Murray, "The multiple flying sidekicks traveling salesman problem with variable drone speeds," *Transportation Research Part C: Emerging Technologies,* vol. 120, p. 102813, 2020/11/01/ 2020, doi: https://doi.org/10.1016/j.trc.2020.102813.

[18] Z. Wang and J.-B. Sheu, "Vehicle routing problem with drones," *Transportation Research Part B: Methodological,* vol. 122, pp. 350-364, 2019/04/01/ 2019, doi: https://doi.org/10.1016/j.trb.2019.03.005.

[19] M. Lin, J. Y. Lyu, J. J. Gao, and L. Y. Li, "Model and Hybrid Algorithm of Collaborative Distribution System with Multiple Drones and a Truck," *Scientific Programming,* Article vol. 2020, 2020, Art no. 8887057, doi: 10.1155/2020/8887057.

[20] D. Schermer, M. Moeini, and O. Wendt, "A matheuristic for the vehicle routing problem with drones and its variants," *Transportation Research Part C: Emerging Technologies,* vol. 106, pp. 166-204, 2019/09/01/ 2019, doi: https://doi.org/10.1016/j.trc.2019.06.016.

[21] D. Sacramento, D. Pisinger, and S. Ropke, "An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones," *Transportation Research Part C: Emerging Technologies,* Article vol. 102, pp. 289-315, 2019, doi: 10.1016/j.trc.2019.02.018.

[22] G. B. Dantzig and J. H. Ramser, "The Truck Dispatching Problem," *Management Science,* vol. 6, 1959.

[23] S. Nanda Kumar and R. Panneerselvam, "A Survey on the Vehicle Routing Problem and Its Variants," *Intelligent Information Management,* vol. 04, 01/01 2012, doi: 10.4236/iim.2012.43010.

[24] R. V. Kulkarni and P. R. Bhave, "Integer programming formulations of vehicle routing problems," *European Journal of Operational Research,* vol. 20, no. 1, pp. 58-67, 1985/04/01/ 1985, doi: https://doi.org/10.1016/0377-2217(85)90284-X.

[25] G. Laporte, "Fifty Years of Vehicle Routing," *Transportation Science,* vol. 43, pp. 408-416, 11/01 2009, doi: 10.1287/trsc.1090.0301.

[26] P. Toth and D. Vigo, "The Vehicle Routing Problem. Monographs on Discrete Mathematics and Applications.," *Philadelphia: Society for Industrial and Applied Mathematics.,* no. 9, 2002.

[27] P. Jupyter. "Jupyter." https://jupyter.org/ (accessed.

[28] P. D. Team. "Optimization with PuLP." https://coin-or.github.io/pulp/# (accessed.

[29] IBM. "IBM CPLEX Optimizer." https://www.ibm.com/analytics/cplex-optimizer (accessed 2021).

[30] K. Kim. "Capacitated Vehicle Routing Problem (CVRP) with Python+Pulp and Google Maps API." Medium. https://medium.com/jdsc-tech-blog/capacitated-vehicle-routing-problem-cvrp-with-python-pulp-and-google-maps-api-5a42dbb594c0 (accessed 31.03, 2021).

[31] G. Developers. "The Directions API overview." https://developers.google.com/maps/documentation/directions/overview (accessed 05.04, 2021).

[32] G. Developers. "Google OR-Tools - Capacity Constraints." https://developers.google.com/optimization/routing/cvrp?hl=th (accessed 10.05, 2021).
[33] G. Thorsnæs. "Ofoten." https://snl.no/Ofoten (accessed 01.05.2021.
[34] D. D. Canada. "Technology." Drone Delivery Canada. https://dronedeliverycanada.com/technology/ (accessed 05.05, 2021).
[35] S. Vegvesen. "Criteria for speed limits." Vegvesen. https://www.vegvesen.no/fag/fokusomrader/trafikksikkerhet/kriterier-for-fartsgrenser (accessed 08.05, 2021).
[36] A. Goodchild and J. Toy, "Delivery by drone: An evaluation of unmanned aerial vehicle technology in reducing CO2 emissions in the delivery service industry," *Transportation Research Part D: Transport and Environment*, vol. 61, pp. 58-67, 2018/06/01/ 2018, doi: https://doi.org/10.1016/j.trd.2017.02.017.
[37] S. Norway. "Fuel consumption and emissions per kilometer driven for a selection of traffic situations and vehicle groups. 2016. g / km." Statistics Norway. https://www.ssb.no/318322/drivstofforbruk-og-utslipp-per-kjorte-kilometer-for-et-utvalg-av-trafikksituasjoner-og-kjoretoygrupper.2016.g-km (accessed 08.05, 2021).
[38] S. Norway. "11418: Occupationally distributed monthly salary, by statistical target, occupation, statistical variable, agreed / normal working hours per week and year." Statistics Norway. https://www.ssb.no/statbank/table/11418/tableViewLayout1/?loadedQueryId=10046764&timeType=top&timeValue=2 (accessed 08.05, 2021).
[39] J. K. Stolaroff, C. Samaras, E. R. O'Neill, A. Lubers, A. S. Mitchell, and D. Ceperley, "Energy use and life cycle greenhouse gas emissions of drones for commercial package delivery," *Nature Communications*, vol. 9, no. 1, p. 409, 2018/02/13 2018, doi: 10.1038/s41467-017-02411-5.