Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Engineering
Department of Energy and Process Engineering

Karla Leticia Prieto Camarillo

Sustainability of widespread last-mile drone delivery systems:

A comparative urban case study in the U.S. based on real-world data

Master's thesis in M.Sc. Circular Economy
Supervisor: Anders Hammer Strømman
Co-supervisor: Jan Klenner and Helene Muri
June 2023

NTNU
Norwegian University of
Science and Technology

Karla Leticia Prieto Camarillo

Sustainability of widespread last-mile drone delivery systems:

A comparative urban case study in the U.S. based on real-world data

Master's thesis in M.Sc. Circular Economy
Supervisor: Anders Hammer Strømman
Co-supervisor: Jan Klenner and Helene Muri
June 2023

Norwegian University of Science and Technology
Faculty of Engineering
Department of Energy and Process Engineering

NTNU
Norwegian University of
Science and Technology

Abstract

The transportation sector is responsible for a substantial share of the worldwide emissions. For this reason, efforts in climate change mitigation should include new vehicle technologies that have the potential of lowering carbon emissions along supply chains. Drones or Unmanned Aerial Vehicles (UAVs) have been identified as an emerging technology within last-mile package delivery operations, given that they are lightweight and energy efficient battery-powered vehicles. Major logistic service providers have already deployed early-stage drone delivery programs in recent years. Therefore, as UAVs become increasingly relevant in last-mile operations, there is a need to evaluate the potential environmental impact of future drone delivery systems.

This master thesis evaluates the environmental performance of a drone delivery system, comparing it to a ground delivery system. Environmental performance was measured in terms of energy consumption and carbon emissions during the transportation phase.

To achieve this objective, an urban case study set in five metropolitan cities in the U.S. was modeled based on a pioneering dataset, containing historical last-mile delivery data from Amazon. In the context of this thesis, 9,157 routes with over 1.3 million delivery coordinates were employed to model the system. A baseline ground delivery system was established based on an electric and a diesel delivery vehicle (G1, G2). Further, four drone delivery cases based on a state-of-the-art fixed-wing VTOL configuration were proposed (D1, D2, D3, D4). Additionally, the effects of transitioning to a less carbon-intensive energy mix were modeled.

The results indicate that drones possess the benefit of being energy efficient, but face limitations in terms of distance traveled and package carrying capacity. In certain cases involving package multi-delivery and a closer distribution center, drones could contend with diesel vehicles. However, electric vehicles outperformed drones in terms of energy consumption and CO_2 emissions.

The findings highlight the importance of implementing multi-delivery as routing measure to enhance the performance of drone delivery in reference to the baseline case (D1). Battery constraints were found to be a significant limitation for multi-delivery; however, it is expected that future drone technology advancements will enable drones to operate over longer distances.

Lastly, the transition to a less carbon-intensive electricity mix would significantly favor electric vehicles and drones, which are battery-powered vehicles, over diesel fleets in terms of operational emissions. However, this would also shift the relevance to other life cycle phases such as vehicle production, infrastructure or end-of-life phases.

Sammendrag

Transportsektoren står for en betydelig andel av de globale utslippene. Derfor bør innsatsen for å redusere klimaendringene også omfatte nye kjøretøyteknologier som har potensial til å redusere karbonutslippene langs forsyningskjedene. Droner eller ubemannede luftfartøyer (UAV-er) har blitt identifisert som en ny teknologi innen pakkelevering, siden de er lette og energieffektive batteridrevne farkoster. Store leverandører av logistikktjenester har allerede tatt i bruk droneleveringsprogrammer i en tidlig fase de siste årene. I takt med at droner blir stadig mer relevante for pakkelevering, er det derfor behov for å evaluere den potensielle miljøpåvirkningen fra fremtidige droneleveringssystemer.

Denne masteroppgaven evaluerer miljøytelsen til et droneleveringssystem og sammenligner det med et bakkeleveringssystem. Miljøytelsen ble målt i form av energiforbruk og karbonutslipp under transportfasen.

For å oppnå dette målet ble det gjennomført en urban casestudie i fem storbyer i USA, basert på et banebrytende datasett som inneholder historiske data om levering av varer fra Amazon. I denne avhandlingen ble 9 157 ruter med over 1,3 millioner leveringskoordinater brukt til å modellere systemet. Det ble etablert et grunnleggende bakkeleveringssystem basert på et elektrisk og et dieseldrevet leveringskjøretøy (G1, G2). I tillegg ble det foreslått fire tilfeller av dronelevering basert på en moderne VTOL-konfigurasjon med fastvinge (D1, D2, D3, D4). I tillegg ble effekten av en overgang til en mindre karbonintensiv energimiks modellert.

Resultatene viser at droner har fordelen av å være energieffektive, men at de har begrensninger når det gjelder tilbakelagt distanse og kapasitet til å frakte pakker. I visse tilfeller med flere pakkeleveringer og et nærmere distribusjonssenter kan droner konkurrere med dieselkjøretøy. Elektriske kjøretøy presterte imidlertid bedre enn droner når det gjelder energiforbruk og CO2-utslipp.

Funnene understreker viktigheten av å implementere multilevering som rutetiltak for å forbedre ytelsen til dronelevering i forhold til referansealternativet (D1). Batteribegrensninger viste seg å være en betydelig begrensning for multilevering, men det forventes at fremtidige teknologiske fremskritt vil gjøre det mulig for droner å operere over lengre avstander.

Til slutt vil overgangen til en mindre karbonintensiv elektrisitetsmiks i betydelig grad favorisere elektriske kjøretøy og droner, som er batteridrevne kjøretøy, fremfor dieselbiler når det gjelder driftsutslipp. Dette vil imidlertid også ha betydning for andre faser av livssyklusen, for eksempel produksjon av kjøretøy, infrastruktur og utrangeringsfasen.

Preface

This master thesis concludes my international Erasmus Mundus Joint Master's Degree in Circular Economy in the Industrial Ecology Faculty at the Norwegian University of Science and Technology.

This thesis continues the work of the research project of the fall semester (Prieto Camarillo, 2022), also written at the Department of Energy and Process Engineering.

I would like to take a moment to express my gratitude to the following individuals: my supervisor Anders Hammer Strømman for his exceptional research ideas, innovative thinking, and unwavering commitment to excellence as a researcher.

I would also like to extend my heartfelt appreciation to my co-supervisors Jan Klenner and Helene Muri for their reliable support, understanding, motivation, and remarkable input, without whom I would not have been able to accomplish this thesis in such a short amount of time.

Additionally, I would like to thank Ruslan Zhuravchak, for his invaluable patience and motivation in explaining the functioning of the virtual machine at the IndEcol department.

Furthermore, I want to acknowledge Javier for all the support and encouragement, which gave me confidence in my programming abilities. I would also like to express my appreciation to Johann for his moral support amidst times of change.

Lastly, I want to extend my gratitude to all my friends and family in Norway, Austria, Mexico, and Germany, whose support I always felt.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ACS: American Community Survey
API: Application Programming Interface
AWS: Amazon Web Services
CSV: Comma-Separated Values
DC: Distribution center
EV: Electric vehicle
GHG: Greenhouse gas
GIS: Geographic Information System
GREET: Greenhouse Gases, Regulated Emissions, and Energy Use in Transportation
ICEV: Internal combustion engine vehicle
IPUMS NHGIS: IPUMS National Historical GIS
KDE: Kernel density estimation
LCA: Life Cycle Analysis
L/D ratio: Lift-to-drag ratio
LSP: Logistic service provider
OSRM: Open Source Routing Machine
RADR: Road autonomous delivery robot
RAM: Random-access memory
RQ: Research Question
SADR: Sidewalk autonomous delivery robot
TTW: Tank-to-Wheel
UAV: Unmanned Aerial Vehicles
vCPU: Virtual central processing units
VM: Virtual machine
VTOL: Vertical take-off and landing
WLTP: Worldwide Harmonized Light Vehicles Test Procedure
WTT: Well-to-Tank
WTW: Well-to-Wheel

1 Introduction

1.1 Background and relevance

According to the Intergovernmental Panel on Climate Change (2022), 15% of the global carbon emissions originated within the transportation sector in 2019. Although this sector is responsible for a considerable share of the worldwide emissions, it is simultaneously regarded as a challenging sector to decarbonize.

In recent years, environmental impact has emerged as a growing concern for businesses worldwide. This heightened concern stems from a combination of factors, including legal and regulatory requirements, the increasingly perceptible effects of climate change and a shift in relevance for some consumer markets. For this reason, companies specifically in the logistics sector are undertaking efforts to redesign their distribution activities, not solely with the aim of reducing costs, but also to minimize environmental impact (Frota Neto et al., 2008). These efforts are included within the expanding field of *Sustainable Logistics and Supply Chain Management*, which recognizes the major impact of logistics and supply chain operations on the environment. This field strives to minimize environmental impacts through emission assessment, reverse logistics and the decarbonization of supply chains (Grant et al., 2015).

A highly complex component within the supply chain is the *last-mile* (Laseinde & Mpofu, 2017). Last-mile operations consist on the delivery of goods to the final customer, specifically along the last leg of transportation (Macioszek, 2018). However, last-mile delivery is considered to be one of the most expensive stages along the supply chain and one of the most challenging to optimize (Awwad et al., 2018). At the same time, service level requirements keep rising as customers come to expect faster deliveries (Capgemini Research Institute, 2019).

Last-mile delivery is also being influenced by the rapid surge of the e-commerce sector in the past years. In 2022, the worldwide e-commerce retail sales were estimated to surpass 5.7 trillion USD with a growing prognosis for the following years (Statista, 2022). For reference, in the U.S. it was estimated that for the year 2017, an average person received around 21 e-commerce packages per year, while in markets such as China this number went up to around 70 packages per year (Briest et al., 2019). Following this pattern, last-mile customer delivery exhibits a clear tendency towards smaller packages and more frequent customer deliveries (European Environment Agency, 2020).

As e-commerce delivery becomes ubiquitous and packages tend to be smaller in size, the potential of Unmanned Aerial Vehicles (UAVs) or drones to address last-mile logistic challenges is greater than ever (Yowtak et al., 2020). To illustrate this, Jeff Wilke, former Amazon Consumer CEO, estimated that around 75-90% of Amazon's deliveries could technically be carried out via drone due to their small weight and size (D'Onfro, 2019).

Commercial drones are an emerging technology that is argued to be more energy efficient compared to larger delivery vehicles such as trucks (Borghetti et al., 2022; Goodchild & Toy, 2018). A further commonly stated benefit of drones, is that in an urban context they would not contribute to traffic congestion (Oršič et al., 2022). In addition, drone delivery is expected to increase in the next years. It is estimated that around 660,000 drone customer deliveries were carried out worldwide from 2019 until 2021, and that this number considerably grew in 2022 (Carter et al., 2022).

For this reason, companies are exploring how drone delivery could be integrated into their last-mile logistics operations. Major Logistic Service Providers (LSPs) such as DHL, FedEx, UPS, the e-commerce company Amazon, or Google-owned Alphabet have established prominent drone delivery programs. Some of them have already carried out pilot deliveries during the past 7 years. Furthermore, the Federal Aviation Administration has authorized different types of *Part 135* certifications for these companies, in order to be able to carry out drone delivery beyond the visual sight line in the U.S. (FAA, 2022).

In terms of the latest advancements in delivery programs, Amazon unveiled its latest drone design, the *MK30*, which is expected to be employed for deliveries under 5 pounds in 2024 (Amazon, 2022). Recently, in December 2022, Amazon carried out with their first small-scale suburban customer deliveries as a part of the Prime Air program in Lockeford, California and College Station, Texas (Lenihan, 2022). Additionally, Walmart has established a network of 36 stores in the U.S. with drone delivery hubs that deliver items from a 20,000-item catalogue to customers within 1 mile of the hubs within less than 30 minutes (Garland, 2023).

In other parts of the world, drone delivery systems have reached a more advanced stage. Meituan, a Chinese digital commerce platform, has been carrying out food delivery with drones in the city of Shenzhen for around 18 months, demonstrating that drones can also be used in the challenging context of urban areas. The company executed over a 100,000 deliveries in 2022 with kiosk-like rooftop stations distributed within the city (Z. Yang, 2023).

In general, the adoption of green vehicles is needed to achieve a reduction of the environmental impacts of last-mile logistics (Patella et al., 2020). However, while emerging technologies exhibit a great potential, they could also have negative externalities on the environment (Yang et al., 2023). Hence, the necessity to evaluate the environmental implications of possible technology transitions arises.

Given that the widespread use of drones in last-mile delivery is still in a relatively early phase, there is a lack of extensive research and consensus regarding the environmental impacts of drone delivery systems (Mitchell et al., 2023). As the current understanding remains limited, it is necessary to further evaluate the possible environmental effects of scaling up drone delivery programs.

1.2 State of knowledge

This subsection provides a condensed overview of the state of knowledge of last-mile delivery in the U.S., the state of current drone delivery and the research regarding the sustainability of last-mile drone delivery. This serves primarily to understand the context of the modeling assumptions made in the study and to contextualize the existing research gaps in literature that were addressed in this master thesis.

1.2.1 Last-mile delivery in the U.S.

Several main enterprises dominate in the U.S. parcel shipping market. According to the Pitney Bowes Parcel Shipping Index (2022), the most relevant entities in the market are UPS, Amazon and DHL, FedEx and the US Postal Service. As can be observed in Figure 1, excluding the postal service, the most significant actors in the parcel market in 2021 were UPS and Amazon, occupying about 24% and 22% of the market parcel volume share respectively. The next relevant actor is FedEx with 19% of the market share by package volume.

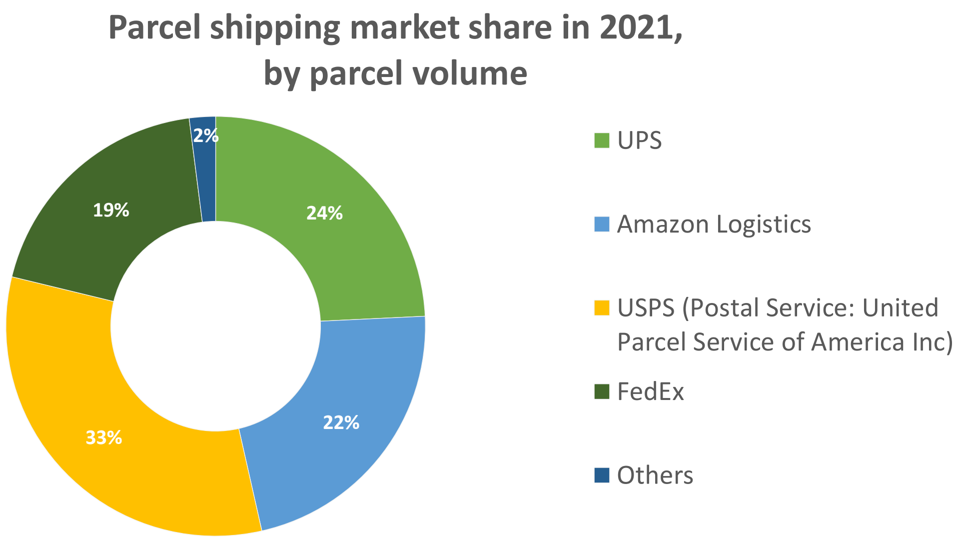


Figure 1. Parcel shipping market share in 2021, by parcel volume (data from Pitney Bowes, 2022)

Last-mile delivery is mainly carried out by light delivery trucks (Awwad et al., 2018). In terms of electrification, 50% percent of fleet operators in the U.S. have taken initial steps to electrify their transportation fleets (Chauhan et al., 2022a). However, it is assumed that the majority of commercial fleets still employ internal combustion vehicles (ICEVs).

As the electrification of commercial fleets is expected to be slower than with passenger vehicles, charging infrastructure is an additional concerning factor (Chauhan et al., 2022b).

Nonetheless there are discussed benefits to fleet electrification, which can include carbon savings and a potential reduction in the total cost of ownership of the fleets (Heid et al., 2017).

The relevant players in the electric delivery van market include startups such as Rivian, Canoo, Workhorse and Arrival, but also established automotive manufacturers such as Ford with the E-Transit or GM with BrightDrop (Boudette, 2023). Nonetheless, delays in the production of electric vans, have also influenced the limited pace of electrification of delivery fleets (Boudette, 2023).

1.2.2 Context of drone package delivery

Drones have multiple possible designs, which are referred to as configurations. For this master thesis, it is relevant to distinguish between multi-rotor, fixed-wing and hybrid configurations.

Firstly, multi-rotor or copter configurations utilize rotors to lift the drone from the ground. Copter configurations are the most common with four rotors (quadcopter), six rotors (hexacopter) or eight rotors (octocopter). The second category, the fixed-wing drone, utilizes wings of varying dimensions to fly, usually needing a large landing and takeoff space (JOUAV, 2022).

Especially relevant for this study, is the hybrid configuration, which combines the fixed-wing with the rotor technology. This hybrid configuration is also known as a fixed-wing vertical take-off and landing (VTOL) drone. Hybrid configurations use the rotors for vertical take-off and landing. However a significant benefit comes with the usage of the wings for cruise mode, which makes a fixed-wing VTOL drone more energy efficient than a copter drone (Figliozzi, 2017; Stolaroff et al., 2018), due to the lift provided by the wing. One downside is that fixed-wing VTOL drones tend to be larger in size than a multi-copter drone (Stolaroff et al., 2018).

Overview of the most relevant drone delivery programs

Logistic Service Provider (LSP)	Drone Program	Most recent drone configuration	Partnership	Use case
Amazon	Prime Air Delivery	Fixed-wing VTOL	-	E-commerce delivery
Alphabet Wing	Wing's Hummingbird	Fixed-wing VTOL	-	Food delivery and e-commerce delivery
DHL	DHL Parcelcopter	Fixed-wing VTOL	Wingcopter	Parcel delivery
FedEx	FedEx Express & Wing Collaboration	Fixed-wing VTOL	Alphabet Wing	Commercial residential drone delivery service & e-commerce
UPS	UPS Flight Forward	Fixed-wing VTOL	Wingcopter	Parcel delivery

Table 1. Drone projects of the major logistic service providers or drone manufacturers (Prieto Camarillo, 2022)

As can be seen in Table 1, the most relevant actors in drone package delivery are Amazon, Alphabet Wing, DHL, FedEx and UPS. Wingcopter and Alphabet are both drone manufacturers, while the rest are logistic service providers. A highly relevant player in the market is Wingcopter, given that it has undergone partnerships with other large LSPs such as FedEx and UPS. Most LSPs have established delivery programs, however these are in the early development phase and are yet to be scaled, mainly due to regulatory, safety or technical feasibility reasons.

In package delivery programs, logistic service providers used to mostly employ copter drones at an earlier stage. This can be for instance observed in the evolution of DHL Parcelcopter program which started with copter configurations. However, a recent trend has been recognized in drone package delivery services that points towards the predominant use of fixed-wing VTOL configurations.

1.2.3 Sustainability of last-mile drone package delivery

Last-mile drone delivery publications have witnessed a surge in the past five years, with more than 50% of the works being published since 2018 (Bányai, 2022). However, in a study by Eskandaripour & Boldsaikhan (2023), it was found that most of the publications relate to routing problems and battery usage or swapping; whereas only a small fraction of the papers includes aspects of environmental protection. This is a known critique, since the economic perspective tends to be more present than the environmental perspective in last-mile drone delivery research (Kirschstein, 2021).

As identified by Eskandaripour & Boldsaikhan (2023) environmental protection is one of the main challenges or aspects to consider with last-mile drone delivery. The authors identified main measures, which include the reduction of Green House Gas emissions (GHG), the improvement energy efficiency and the utilization of renewable energy sources.

Given the aim of this study, this literature review focuses primarily on selected works that consider the environmental aspects of last-mile drone delivery. Table 2 shows an overview of the most relevant works, including their compared delivery systems, employed distance calculation methods, drone model employed, drone energy consumption model, life cycle phases considered and context of the study.

With regards of how the environmental perspective is considered in last-mile drone delivery, most of the studies have focused on quantifying energy consumption and environmental impact in terms of carbon emissions, mainly considering the operational delivery phase. There are a few studies that go beyond the operational operations, which will be mentioned at the end of this subsection.

The methods to calculate the distances, which are the basis to obtain carbon emissions and energy consumption values for the operational phase, vary greatly in literature. Li et al. (2023) classified drone routing literature into different categories mainly heuristic methods, which consist of approximations, and vehicle routing problems with optimization.

As for the specific methods used, analytical approximation with mathematical methods, such as continuous approximation, has been widely used (Figliozzi, 2017; Zhang, 2021). Another commonly employed approach consists in simulation-based experiments, using statistical distributions or randomly generated locations. Further elements such as population density or GIS-based modelling have been considered in the modelling. Within the analyzed literature, there was no study found that based distance calculations on historical data.

Author, year	Delivery vehicles	Distance calculation method	Drone model	Drone energy consumption model	Life cycle scope	Case study or context
Figliozzi, 2017	UAV, Diesel, Electric tricycle	Continuous Approximation	Quadcopter (MD4-3000)	D'Andrea (2014)	Production, Operation and Disposal (GHG-only)	U.S.
Goodchild & Toy, 2018	UAV, Diesel	GIS-based modelling, 10 scenarios	-	-	Delivery operations	Los Angeles County, California
Koiwanit, 2018	UAV	-	Quadcopter (Inventory based on DJI models)	-	LCA (cradle-to-grave)	Chiang Mai, Thailand
Stolaroff et. al, 2018	UAV, EV, Diesel (incl. other variations)	-	Quadcopter, Octocopter	-	Production, Delivery operations, Warehouse infrastructure	San Francisco Bay Area
Chiang et. al, 2019	Tandem	Green Vehicle Routing Problem	-	-	Delivery operations	-
Kirschstein, 2020	UAV, EV, Diesel	Simulation experiments, stat. distributions for density delivery	Quadcopter	Own model, wind speed consideration	Delivery operations	Berlin, (urban vs. rural area)
Figliozzi, 2020	UAV, SADR, RADR, EV, ICEV	Continuous approximation	Quadcopter (MD4-3000)	Specifications (21.6 Wh/km)	Delivery operations	Urban grocery delivery
Yowtak et al., 2020	UAV, ICEV, BEV	Ground vehicles travel 1.34 km for every UAV km	Quadcopter (Harris Aerial Carrier HX8 Power XL)	Manufacturer data	LCA (cradle-to-grave)	Michigan, grocery delivery
Zhang, 2020	UAV, Diesel, Tandem	Continuous approximation	Quadcopter	-	Delivery operations	-
Kirschstein, 2021	UAV or EV in mixed-fleet	Simulation experiments, considering population density	Quadcopter	Own model, wind speed consideration	Delivery operations	New York City
Rodrigues et. al, 2022	Diesel, EV, UAV, E-cargo bicycle	Own assumptions	Quadcopter (M100, DJI Matrice)	Field measurement model	Delivery operations	U.S.

Table 2. Overview of selected literature regarding sustainability of drone last-mile delivery

For instance, one of the foundational works by Goodchild & Toy, (2018) compared a drone-based system with a truck-based system in the context of Los Angeles California in terms of CO_2 emissions. GIS-based modeling was done based on 10 proposed scenarios departing from a central depot and in which the delivery density, was affected by varying population density.

One of the main differences within the explored literature are the comparison systems evaluated. For instance, drone delivery has been compared most often with diesel delivery trucks or with battery electric delivery trucks. At the same time, other ground-transportation types have been used, such as sidewalk autonomous delivery robots (SADRs) or with road autonomous delivery robots (RADRs) (Figliozzi, 2020), or e-cargo bicycles (Rodrigues et al., 2022).

Regarding the context of the case studies, different geographical locations have been analyzed. Most of the studies are carried out in a U.S. context, whereas to a lesser extent, some research has also been carried out in the context of Europe in Berlin (Kirschstein, 2020) or Asia in Thailand (Koiwanit, 2018).

As for the simulation of different contexts, mostly parameters such as delivery density and delivery distances are used to create different delivery scenarios. For instance, (Kirschstein, 2020) compares a drone fleet with both, diesel and electric delivery trucks, departing from a central Berlin distribution center, comparing a more rural and an urban scenario, through varying delivery density and distances.

Table 2 shows that most of the research used a quadcopter drone to model energy consumption, some of the employed models are the MD4-3000 from Microdrones (Figliozzi, 2017, 2020) or the M199 from DJI Matrice (Rodrigues et al., 2022). The employed drones in literature consist in commercially available drones used for purposes such as aerial photography. Even though Figliozzi (2020), used a quadcopter configuration in his research, he highlights the potential of fixed-wing VTOL configuration for drone delivery due to its higher energy efficiency than multi-copter drones.

In order to estimate drone energy consumption, some works use theoretical models (Figliozzi, 2017; Kirschstein, 2020), while others use a regression approach employing field data (Rodrigues et al., 2022) or available manufacturer data (Yowtak et al., 2020). For instance, Kirschstein (2020) provided a detailed energy consumption model for the drone, considering three separate flight regimes of take-off, hovering and landing, further considering wind conditions. In the case of Rodrigues et al. (2022), 188 field measurements using a quadcopter drone were used to obtain generalizable coefficients to calculate drone energy use. This model distinguished between the three different flight regimes.

Theoretical energy consumption models have mainly been designed for multi-copter configurations, nevertheless it is known that energy consumption estimations for drones vary quite considerably in literature. Zhang et al. (2021) documented the discrepancies between the drone energy consumption models available in literature and standardized the nomenclature. Even though the same operating conditions and drone specifications were tested using the various drone energy consumption models, the energy per meter and operating range fluctuated considerably.

Other configurations other than drone-only systems have also been explored to a lesser extent. Chiang et al. (2019) explored drones delivering in tandem with delivery trucks and determined that the usage of UAVs could reduce vehicles employed and total time for delivery, when used in combination with trucks.

Few works consider further life cycle phases of drone delivery services. In a preliminary study, Figliozzi (2017) considered the greenhouse gas (GHG) emissions of production and disposal of vehicles, denominated as "vehicle phase emissions". Furthermore, emissions from warehouse infrastructure were considered by Stolaroff et al. (2018). This study suggested that drone delivery systems could consume less energy per package than a diesel truck system, but that the energy employed in warehouse operations should be minimized in order to bring about potential benefits of drone delivery.

A Life Cycle Analysis (LCA) perspective is considered by a lesser amount of studies. For example, Koiwanit (2018) carried out a non-comparative cradle-to-grave LCA in the context of online shopping, which provided initial insights into the potential environmental impact of drone parts production. Later on, Yowtak et al. (2020) performed a comparative LCA comparing a UAV grocery delivery service with a combustion engine vehicle and an electric vehicle system. The study suggested that the UAV use phase predominates over other life cycle phases in several environmental impact categories. However, Mitchell et al. (2023) recently revealed that insufficiently reliable data is usually employed to model the drone part production and end-of-life, given that parts and composite materials are not generally analyzed in detail. Furthermore, they highlighted the need for comparable and more robust LCAs of drone delivery.

1.2.4 Research gaps to be addressed

As identified in the state of knowledge, past research usually bases the drone energy consumption on multi-copter configurations. Yet, several technological advancements have occurred up to the current time, for instance fixed-wing VTOL configurations, multi-delivery and an improved flight range.

A further relevant observation is that most of the explored literature based the distance calculations on analytical approximation or simulation experiments. These methods used unique sets of assumptions and can be categorized as employing artificially generated data. However, in order to represent large-scale last-mile delivery operations in a more accurate manner, there is a need to explore the environmental implications based on realistic parameters or historical data.

In essence, it is worth noting that different studies use varying assumptions both for calculating a) traveled distance and b) drone energy consumption. For this reason, given the multiplicity of approaches and need to obtain a clearer picture of the implications of last-mile drone delivery, it is crucial to address these two aspects.

1.3 Objective of the study

1.3.1 Objective and scope

The objective of this master's thesis is to explore and quantify the environmental implications of a potential widespread adoption of drone delivery systems in last-mile home delivery.

For this purpose, scenario modelling is carried out using a first-of-its-kind historical last-mile delivery dataset from Amazon. This case study considers a set of real-world delivery routes carried out in 5 metropolitan cities in the U.S. included in the dataset, therefore setting the context within urban last-mile delivery.

The scope of this master thesis focuses on the environmental performance of the operational delivery phase, meaning the transportation phase. Emissions resulting from other life cycle phases are out of the scope of this study.

To evaluate the case of this potential technology transition, drone-based delivery systems are compared to ground-based delivery systems in terms of energy consumption and CO_2 emissions in the operational delivery phase (see Figure 2).

Road-based technologies such as electric or diesel delivery vehicles are used as a benchmark system. The chosen drone technology is a fixed-wing VTOL drone configuration, in order to reflect the possible future drone delivery operations. The developed drone energy consumption model by Prieto Camarillo (2022) is employed.

Several alternative drone delivery scenarios including multi-delivery and alternative depots are proposed and evaluated.

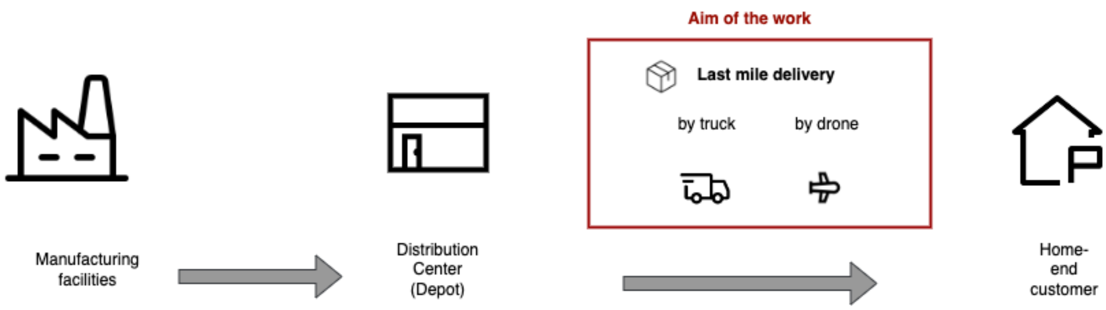


Figure 2. Scope and aim of the work, environmental implications of last mile delivery by truck and drone. Based on (Borghetti et al., 2022)

1.3.2 Research questions

Based on this, two main research questions to be addressed in this thesis have been identified:

> ***RQ1:*** *How do widespread drone-delivery systems environmentally perform in comparison to current and future ground-based delivery systems, in the context of last-mile home package delivery during the operational delivery phase?*
>
> ***RQ2:*** *Which measures could be implemented to reduce the environmental impact of drone-delivery systems in an urban delivery context and how much improvement could these potentially bring?*

Research question 1, is answered by modelling the drone and ground delivery systems, through the base-case scenarios departing from the original depots found in the dataset. The environmental aspects considered will be energy consumption and carbon emissions.

Research question 2, is explored by: a) technical routing measures: which include proposing alternative delivery scenarios for drone delivery (multi-delivery or alternative distribution centers); and b) technological measures: meaning through the simulation of the effects of the decarbonization of the electricity mix.

In this master's thesis, current ground delivery systems are defined as diesel-based delivery fleets, whereas future ground delivery systems include battery electric delivery trucks.

1.3.3 Main contributions of the work

Based on the identified literature research gaps mentioned in sub-section 1.2.4, this master's thesis aims to contribute in three main ways:

1) Using real-world data to evaluate the environmental operational performance of a drone vs. a ground-based last-mile delivery system.
2) Employing a current fixed-wing VTOL drone configuration to better represent the operation of future drone delivery fleets.
3) Simulating the multi-delivery of drone package delivery considering realistic battery and range constraints.

2 Methods

This section describes the methods utilized to model the drone and ground delivery systems used in the comparative analysis. Firstly, the case study of this master thesis, as well as all utilized datasets are described. Secondly, the specifications behind the ground vehicle and drone delivery models are provided. Thirdly, the modeled delivery cases are exemplified. Lastly, a special section dedicated to providing details of the technical implementation of the model in Python is provided, which was a substantial part of this master thesis.

2.1 Case study and datasets description

2.1.1 Relevance of real-world data

The objective of this thesis is to model and compare widespread last-mile home delivery operations in terms of energy consumption and CO_2 emissions, specifically with the focus on home package delivery with truck or drone. Taking this into account, the decision to model a specific scenario or case study is highly relevant, as this inevitably impacts the results obtained.

The usage of different assumptions in route simulations can lead to results with varying degrees of accuracy when it comes to representing realistic last-mile delivery operations. Additionally, last-mile operations are highly case-specific as well. For instance, the design of the operation network or the traveled distances change in an urban, rural or even a time-sensitive setting.

It is assumed that previous research is mainly based on simulations or assumptions, due to the scarcity of publicly available operations data from logistics service providers. Given the existence of various logistic service providers within the market, the last-mile delivery system operates in a decentralized manner with multiple actors. This poses a challenge to obtain insights into the overall operations within the system. Usually LSPs, do not make their operations data publicly available. In this way, routing problem research usually makes use of so-called *benchmark datasets* with the purpose of evaluating the performance of routing solutions. However, these benchmark datasets are solely based on synthetic data (Merchán et al., 2022).

In order to model routing conditions as realistically as possible, this thesis bases its modelling on one of the only publicly available real-world last-mile delivery datasets up to this date. This dataset was recently published by the e-commerce company Amazon and contains a sample of historical home delivery operations data.

In addition, in order to contextualize the Amazon dataset in terms of the demographic properties of the regions covered, two other complimenting datasets were used in this study.

In summary, the three main types of data that were used were: delivery data, population data and income data. Although the focus of this study primarily lies in the usage of historical delivery data, Table 3 provides an overview of the main characteristics and source of all datasets used in this study. Each of the datasets will be described in detail in the following sections.

	Delivery data	**Population data**	**Income data**
Dataset Source	Amazon	WorldPop	IPUMS NHGIS
Type of information	Coordinates from delivery stops of historical routes, sequences of stops and corresponding route IDs	Estimated population density per grid-cell	Estimates for per capita income in the past 12 Months (American Community Survey)
Year of data	2018	2020	2021
Geographical resolution	Geographical latitude and longitude of stops within routes	Approx. 1 km at the equator (30 arc)	5-digit ZIP code areas
Units	Geographical coordinates of delivery points and sequence of delivery points	Number of people per km^2	Inflation-adjusted USD per capita per year
Format	JSON files	TIF file	CSV and SHAPEFILE
Geographical extent	Austin, Boston, Chicago, Seattle and Los Angeles	U.S. wide	U.S. wide
Access to dataset	Amazon Web Services: https://registry.opendata.aws/amazon-last-mile-challenges/	WorldPop Hub: https://hub.worldpop.org/geodata/summary?id=39730	IPUMS NHGIS data finder: https://data2.nhgis.org/main

Table 3. List of datasets used in the study with their main characteristics and sources

2.1.2 Context and relevance of Amazon dataset

The delivery dataset was made available in the context of the Amazon Last Mile Routing Research challenge in 2021. This is a research competition, supported by the MIT Center for Transportation and Logistics.

The research challenge of 2021 had the purpose of solving vehicle routing problems with real-life data. Specifically, the challenge consisted in leveraging machine learning and other data-driven computational approaches, in order to create new routing solutions that learn from and include the tacit knowledge of last-mile delivery drivers (Amazon Last-Mile Routing Research Challenge, 2021). Even though the original context of the dataset was for a logistics optimization challenge, Amazon made the dataset publicly available for further research to build upon it.

This is an unparalleled dataset, given that it is the "first large and publicly available dataset" in literature to include historical last-mile delivery routing information that moves away from previous synthetic benchmark datasets (Amazon, 2021; Merchán et al., 2022).

2.1.3 Amazon dataset description

This dataset contains a sample of more than 9,000 historical route sequences from the year 2018 in the United States of America. It provides the stop coordinates from delivery routes departing from 18 distribution centers (DCs) in five different metropolitan cities. The stop coordinates were obfuscated and the "route identifiers were randomly generated" in order to ensure the anonymity of the recipients and the drivers (Merchán et al., 2022).

The cities included are Austin, Boston, Chicago, Los Angeles and Seattle. Figure 3 shows a visual representation of the geographical location of the distribution centers (DCs) or depots, as well as the amount of depots located in each city. Each DC has an assigned identifier including the name of the city, e.g. *LA* for Los Angeles or *SE* for Seattle. These distribution centers are the departure point, from which the delivery trucks then perform the home-delivery routes. A full list with the DC codes and coordinates can be consulted in the appendix A.1.

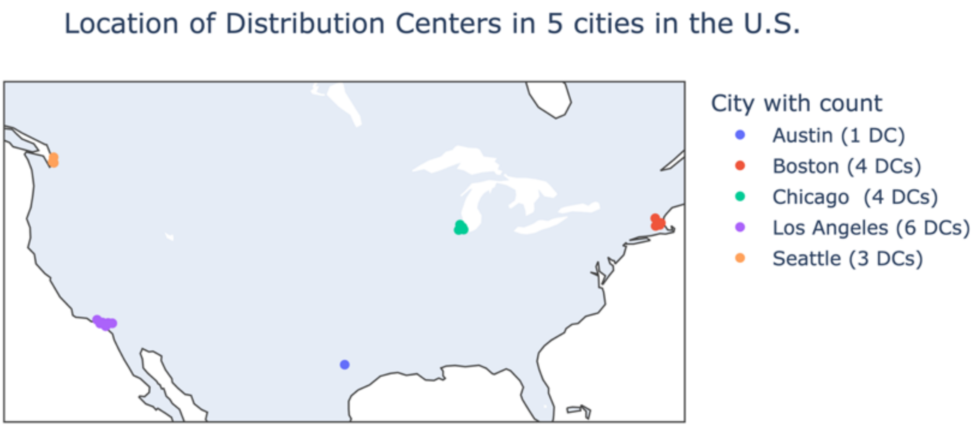


Figure 3. Geographical location of the distribution centers included in the dataset

The data are available on the Registry of Open Data on Amazon Web Services (AWS), as a S3 bucket resource type, which can be accessed through the AWS CLI [here](). The resource includes several files in a JSON format with a 3.34 GB size.

The dataset has a complex structure and is divided into three main sub-components: transit-level, package-level, and route-level information (Amazon Last-Mile Routing Research Challenge, 2021). The transit-level information includes the anonymized coordinates for the stops in each route and was therefore a crucial component of the employed data for this study. The route-level features included important data such as the order of the observed sequences. Finally, package-level information contained data to describe the characteristics of the shipments such as dimensions of the packages. This last data section was not utilized for this thesis, given to its secondary relevance to the modelling.

In this way, the relevant information employed in this study consists in the route identifier (route ID), station code (DC code), coordinates of each stop made in the route (latitude/longitude) and the observed sequence of the stops within each route. An overview of each data category with an example value can be seen in Table 4.

Data category	Exemplary value
DC code	e.g. DLA8
Route ID	e.g. RouteID_00ae3f5e-a708-4c37-b9c4-ebd3964dbdac
Stop ID	e.g. NH
Sequence	e.g. 67
Coordinates	e.g. 30.445236, -97.709418

Table 4. Exemplary values of the relevant information used in this study, contained in the Amazon dataset

The data are divided into two separate datasets, each of approximately 3,000 and 6,000 routes, known as the evaluation and the training datasets (Merchán et al., 2022). This division of the data is relevant when evaluating the performance of a routing proposition made with machine learning, which would be benchmarked with the evaluation dataset. The difference between these two datasets lies in the perceived route quality (Merchán et al., 2022). However, for this purpose and since the scope of this thesis excludes routing optimization, the division between training and evaluation data is not considered relevant for this study. For this reason, the totality of 9,000 routes were used in the modelling.

For simplification and feasibility purposes in modelling, it is assumed that one stop in the dataset is equivalent to one package being delivered. Figure 4 provides a close-up of the coordinate stops located in the Los Angeles area.

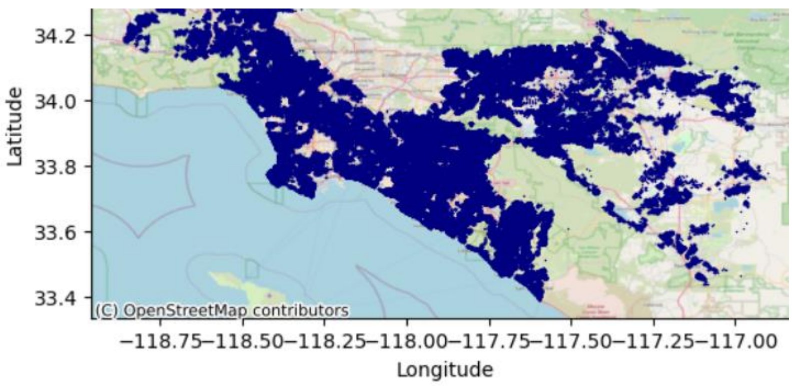


Figure 4. Close-up of the coordinate stops located in the Los Angeles area

During the research project in the fall semester, a small set of routes corresponding to about 0.2% of the routes were explored to serve as a small-scale proof of concept of the basic delivery model (Prieto Camarillo, 2022). This small sample corresponded to 166 routes departing from one distribution center in a suburban area named Everett, which is located around 40 km to the North of Seattle. The DC code of the explored distribution center is DSE4.

2.1.4 Contextualizing datasets description

With the purpose of contextualizing the geographical data contained in the Amazon dataset, additional demographic information was gathered. In this case, the relevant identified demographic factors were population and income. This was done for the purpose of exploring the representativeness of the data by obtaining additional information about the regions that the Amazon dataset covers.

As for the dataset containing income data, the geographical resolution corresponds to the areas of U.S. ZIP codes, which is one of the best available resolutions for income data in the U.S. The data contained is the estimates of per capita income per year, gotten from the American Community Survey (ACS) for the year 2021.

The dataset was obtained from the IPUMS National Historical Geographic Information System (Parks et al., 2022). The data is in an CSV file format with the corresponding income values per ZIP-code, along with a separate shapefile containing the geodata with the borders of the ZIP codes. Which in this case, the two files were merged at a later stage.

Regarding the population dataset, a dataset from WorldPop was used, which contains the population density in the geographical resolution of approximately 1 km at the equator (WorldPop & CIESIN Columbia University, 2020). WorldPop is a high-density demographic dataset project funded by the Bill and Melinda Gates Foundation and carried out by a consortium of universities, among them are the University of Southampton, University of Louisville, Universite de Namur and Columbia University.

The data is contained in two possible formats a CSV or a TIF file. In this study, the TIF file format was used. The information contained is classified as raster data, since it is contained in a grid structure.

There is an alternative dataset with the highest geographical resolution available of approx. 30 meters at the equator from Meta (Data for Good at Meta & Columbia University, 2020). The feasibility of utilizing this dataset was tested. However, it was found that due to the higher resolution, the dataset contained a considerable amount of null population values in the coordinates of the Amazon dataset. In addition to this, a higher processing power for such a large dataset was required. Since Meta’s dataset exhibited an unnecessarily high geographical granularity for this study’s purpose, the resolution of the WorldPop dataset was deemed more appropriate for this contextualization.

2.2 Delivery system specifications

In this subsection, the specifications used to model the ground-based and the drone-based delivery systems will be described in further detail.

The model makes use of the specifications of certain chosen vehicles. These vehicles were chosen with the purpose of serving as an appropriate representation of the transportation vehicles that might be used in future or are already being used in last-mile delivery systems. In the following sub-sections, the relevant specifications of each vehicle, as well as their specific energy consumption and carbon emission conversion factors will be mentioned.

As this master thesis is the continued work of the research project (Prieto Camarillo, 2022), there are parts of the model that were previously established. These were already described in the context of the research project (Prieto Camarillo, 2022). However, in order for this thesis to be a stand-alone document and to fully specify the methods used in this study, they will also be explained in this master thesis.

For this reason, it is important to note that sections 2.2.1, 2.2.2 and 2.2.3, are based on sections included in the research project and are therefore identified with extra indent.

2.2.1 Diesel delivery vehicle

To represent current diesel last-mile delivery systems, the Ducato model by Fiat was selected. Three main reasons contributed to its selection.

The first reason is that the online van configurator from Fiat, offers all the required specifications regarding dimensions, weight and fuel consumption. As shown with other relevant specifications in Table 5, the long base length was chosen, which leads to a gross vehicle weight of 3.5 tons and a carrying volume of 15 m^3.

The second reason is that there is an electric equivalent to the Ducato, the E-Ducato, which can be used as a direct electric comparison vehicle with the same dimension specifications.

NEW DUCATO VAN (35 LH3 140 HP MULTIJET III MY21)	
Characteristic	**Specification**
Carrying volume	15 m^3
Base	long
Gross vehicle weight	3.5 tons
Engine	Diesel
CO_2 emission Combined	227 g/km
Fuel consumption (WLTP, combined)	8.6 l/100km or 0.0861 l/km
Emission Levels	EURO6D_FINAL

Table 5. Specifications for Fiat's Ducato based on Fiat, 2020 (Prieto Camarillo, 2022)

The third reason consists in the comparability of the Ducato to other large vans in the same segment. For instance, vans that are widely used in last-mile delivery such as the Mercedes Sprinter and the Ford Transit. These are comparable regarding carrying capacity and fuel consumption.

As for the vehicle fuel consumption, the Ducato is estimated to consume 8.6 liters of diesel per 100 km (Fiat, 2020). Subsequently, the fuel consumption was translated to contained energy, in order to facilitate a comparison with the energy consumption of the drone and the electric vehicle.

To obtain the equivalent of the energy available or energy consumed in a diesel engine (McKinsey Energy Insights, n.d.), the lower heating value was employed. According to the U.S. Department of Energy, (n.d.) the lower heating value of low sulfur diesel, which refers to common diesel, corresponds to 128,488 Btu/gal. This is equivalent to 9.9476 kWh/l.

The vehicle fuel consumption is translated to contained or consumed energy per km and per 100 kms with the following formulas below

$$Energy\ per\ km = 9.947\frac{kWh}{l} * 0.0861\frac{l}{km} = 0.8565\ kWh/km$$

$$Energy\ per\ 100\ km = 9.947\frac{kWh}{l} * 8.61\frac{l}{100\ km} = 85.65\ kWh/100km$$

Equation 1. Calculations for energy consumption per km and 100 km (Prieto Camarillo, 2022)

On a last step, the conversion to CO_2 emissions is carried out. In this case, the well-to-tank (WTT) and the tank-to-wheel (TTW) emissions were considered. WWT emissions correspond to the emissions originating in the upstream supply chain of fuel production (European Commission JEC, 2016), while TTW are equivalent to the tailpipe emissions due to the fuel combustion in the utilization phase of the vehicle (ANL, n.d.). The emissions resulting from WTT and TTW stages are collectively referred to as well-to-wheel (WTW) emissions.

The 2021 GREET (Greenhouse Gases, Regulated Emissions, and Energy Use in Transportation) model was used (ANL, 2021). The GREET model has also been used by Figliozzi (2017) and Jang & Song (2015) in past studies. The relevant value for the model is the emissions of 0.326 kg of CO_2 per kWh contained in a U.S. diesel mix (see Table 6).

Phase	GHG Emissions: Diesel US Mix	Unit
WTT (well-to-tank)	0.056	kg CO_2eq /kWh
TTW (tank-to-wheel)	0.270	kg CO_2eq /kWh
WTW (well-to-wheel)	0.326	kg CO_2eq /kWh

Table 6. GHG emissions along the fuel cycle from GREET WTW Model (Prieto Camarillo, 2022)

2.2.2 Battery electric delivery vehicle

It was necessary to select an electric vehicle that was comparable in terms of dimensions and weight to the chosen diesel vehicle. For this reason, the E-Ducato from Fiat, the electric counterpart of the diesel Ducato, was chosen with the same specifications. On a further note, the E-Ducato was recently announced to be added to the last-mile delivery fleet of DHL Express in Europe (DHL, 2021).

The E-Ducato has an electric consumption of 0.349 kWh per km according to the WLTP, combined test (see Table 7). However, there is a difference gap between the certification and the real setting values, since factors such as operational mass and ambient temperature are difficult to factor into the tests (Fontaras et al., 2017).

In the simulated context of last-mile package delivery there are two main factors that differ compared to a WLTP test. Firstly, since the WLTP test applies for a vehicle without loaded cargo, a loaded vehicle with packages would have a higher energy consumption rate. Secondly, the package delivery driving behavior is significantly different to the behavior of a conventional passenger car. This involves increased stopping and accelerating for the package delivery. To compensate for these differences in the case of package delivery, a moderate 15% increase on the WLTP value was considered. Thus, the energy consumption value used in the model was of 0.401 kWh per km. Nonetheless, further data and research would be needed to define a more accurate value for package delivery.

E-Ducato (35 LH3 Van 79kWh MY20)	
Characteristic	**Specification**
Carrying volume	15 m^3
Base	long
Gross vehicle weight	3.5 tons
Battery	79 kWh
Electric range (WLTP, combined)	162.8 km
Electric consumption (WLTP, combined)	34.9 kWh/100 km or 0.349 kWh/km

Table 7. Specifications for Fiat's E-Ducato based on Fiat, 2020 (Prieto Camarillo, 2022)

Regarding the CO_2 emissions coming from the electricity grid in the U.S., the average value for the year 2019 of 0.433 kg of CO_2 per kWh was employed (EPA, 2023). This value factors in line losses, yet one disadvantage is that it does not account for other greenhouse gases other than carbon dioxide emissions. This value was also used for the electricity employed to charge the battery of the drone.

2.2.3 Drone model

The chosen drone specifications come from the Wingcopter 198, a state-of-the-art drone for cargo delivery from the German company Wingcopter.

A significant reason why this drone was chosen is that it has a fixed-wing VTOL configuration. As previously mentioned, this configuration has clear advantages in terms of range and energy efficiency compared to copter drone configurations due to the lift of the wing. What enables it to fly both modes, hover mode and cruise mode, is its tilt-rotor mechanism (Wingcopter, 2021b).

A further reason is that this drone model has the ability to carry out package multi-delivery with a triple drop when having a maximum payload of 5 kg, whereas the majority of copter drones can usually manage a single payload (see Figure 5).

Additionally, section 1.2.2 described the trend of current drone delivery programs to utilize a fixed-wing VTOL configuration over copter drones, which were more prevalent in a previous stage of drone delivery. For this reason, in order to reflect drone operations as best as possible, it was relevant to choose a drone specifically meant for package delivery purposes, instead of a drone for other applications.

Figure 5. Wingcopter 198 Triple Drop (Wingcopter, n.d.)

The specifications of the drone were obtained from a fairly comprehensive technical detail specification sheet (Wingcopter, 2021a). As can be seen on Table 8, the maximum range of the drone while fully loaded is 65 km.

In order to get the values for energy consumption, during the fall research project, a specific energy model for this drone was developed based on the model of D'Andrea (2014). In existing literature, energy consumption models for drones are usually designed for copter configurations and the energy per meter values vary highly (Zhang et al., 2021). This energy consumption model for a fixed-wing VTOL configuration was described in further detail in the research project and can be consulted in the supplementary material.

D'Andrea's model was adapted using the specifications of the Wingcopter 198, and in addition a lift-to-drag (L/D) ratio suited for such a fixed-wing VTOL drone was calculated.

The L/D ratio was on average around 8 and varied depending on the weight of the payload, under the assumption that this influenced the reference area of the drone. This L/D ratio is found within the expected value, given that a quadcopter would have a L/D ratio of around 3 (D'Andrea, 2014) and a commercial aircraft of about 15 (Martinez-Val et al., 2005).

In addition, the developed energy consumption model was validated with the specifications of a similar commercially available drone, the CW-15 from JOUAV (see supplementary material).

In the case of the drone, the conversion to CO_2 emissions, was carried out analogue to the electricity conversion process explained for the electric vehicle.

Wingcopter 198 Delivery Variant	
Characteristic	**Specification**
Weight empty	10 kg
Weight with batteries	20 kg
Payload capacity	Max. 5 kg
Li-Ion Battery (at 80% capacity)	1302.4 Wh
Max. range estimation with 1x intermediate delivery	65 km (with a 5 kg payload)

Table 8. Wingcopter 198 Delivery Variant Specifications, based on Wingcopter, 2021 (Prieto Camarillo, 2022)

2.3 Modeled delivery cases

The distance calculation process is highly relevant for this study, considering that the energy consumption is based on the distance traveled during the delivery route. In this master thesis, different cases were modeled, all of which influenced the traveled distance.

The simulation of several cases was carried out in order to test different settings for drone delivery and to compare them to a baseline case. There is a total of six modeled cases which were assigned a case number. To discern the cases, the denominations starting with G, refer to the modelling with ground vehicles and D, with drones.

Firstly, the baseline cases were modeled using the given positions of the distribution centers in the Amazon dataset. The baseline cases refer to G1, G2 and D1. However, to address RQ2, three additional measures were proposed to explore factors that could potentially impact the performance of widespread drone delivery.

Therefore, in addition to the baseline cases departing from the original depot the following three measures in case of the drone were proposed:

- Alternative DC (D2)
- Multi-delivery (D3)
- Combination of alternative DC & multi-delivery (D4)

Table 9 provides an overview with the most important attributes of each modeled case. These attributes include the case number, case name, which distribution center the route is departing from, if the scenario was modeled for both truck and drone delivery or only for drone delivery and if the case includes drone-multi-delivery. Each of the four cases will be described in further detail in the following sections.

Overview of the modeled cases

Case No.	**Case name**	**Distribution Center**	**Drone multi-delivery**	**Vehicle**		
				Battery Electric Truck	**Diesel *Truck***	**Drone**
G1	Baseline / Original DC	Original		X		
G2	Baseline / Original DC	Original			X	
D1	Baseline / Original DC	Original				X
D2	Alternative DC	Alternative				X
D3	Multi-delivery	Original	X			X
D4	Alternative DC & multi-delivery	Alternative	X			X

G: ground transportation; D: drone

Table 9. ***Overview of the different cases modeled in this study***

2.3.1 Baseline case

The baseline cases (G1, G2, D1) consist in modelling the routes departing from the original DC coordinates specified in the Amazon dataset. This is the baseline or reference, to which the other cases can be subsequently compared to. The baseline case was modeled both for truck delivery and drone.

In the case of truck delivery, the distances covered by the battery electric vehicle and diesel vehicle are identical and therefore the distance calculation was carried out only once for both cases. These distance calculations were consecutively translated to energy consumption using their different respective specific conversion functions.

For truck delivery, the route calculation consists of the distance from the original DC to each of the given stops in the right sequence, as can be seen on the left side of Figure 6. The figure illustrates how a route has n intermediate stops and when all deliveries are concluded, the truck returns to the original DC.

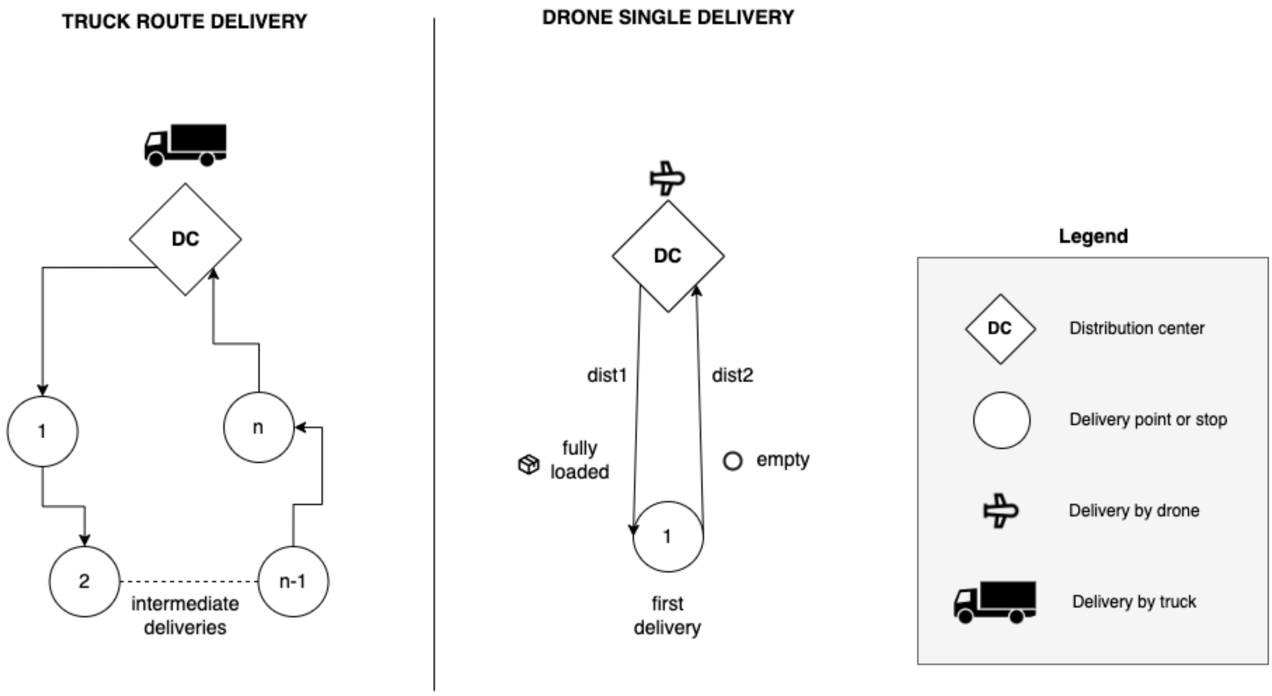


Figure 6. Illustration of difference between truck and drone delivery

For drone delivery, this translates into a point-to-point distance calculation departing from the original DC to the delivery stop and thereafter back to the original DC, as can be seen on the right side of Figure 6. This process is then done by the drone for every stop, to deliver all packages within the designated route. Accordingly, if a route consists of 160 stops, the drone would undertake 160 individual flights from the distribution center to each delivery point and back.

Figure 7 presents an exemplary calculation of one truck delivery route (right) and the drone delivery flights required to cover the packages on that route (left). It can be perceived that there is a single route in the case of the truck delivery, while for the drone delivery there are numerous straight-line flights to and from each delivery point in the route.

Ground Vehicle Delivery Route

Multiple Drone Delivery Flights

Figure 7. Exemplary delivery route for ground vehicles (left) and drones (right) (Prieto Camarillo, 2022)

2.3.2 Alternative DC case

A factor that might influence the performance of widespread drone delivery is the distribution hub proximity. Considering this, an alternative distribution center location from which drones could carry out deliveries was proposed (D2).

The proposed alternative depot consisted in the center of gravity of all the stops that assigned to a specific distribution center. On the grounds that one delivery coordinate is assumed to represent one delivered package, the coordinate of the alternative DC corresponds to the average of the stops coordinates.

Figure 8 illustrates the location of the original and alternative DCs for the DSE4 in Everett, located north of Seattle. As can be seen in the figure, the alternative DC is located south to the original DC. The reason for that, is that only sparse stop clusters are located in the north, in contrast to the more densely distributed stop clusters in the south. In this manner, the alternative depot is located closer to where most of the delivery stops are.

It is important to note that this center of gravity approach is used to decrease the average point-to-point distances covered. It does not minimize all individual point-to-point distances, as some may increase and others decrease. Nonetheless, this approach is meant to lower the overall average distance covered.

In this way the drone flights are calculated in the same manner as described in section 3.3.3.1, with the only modification being the departure point, which is now the alternative DC.

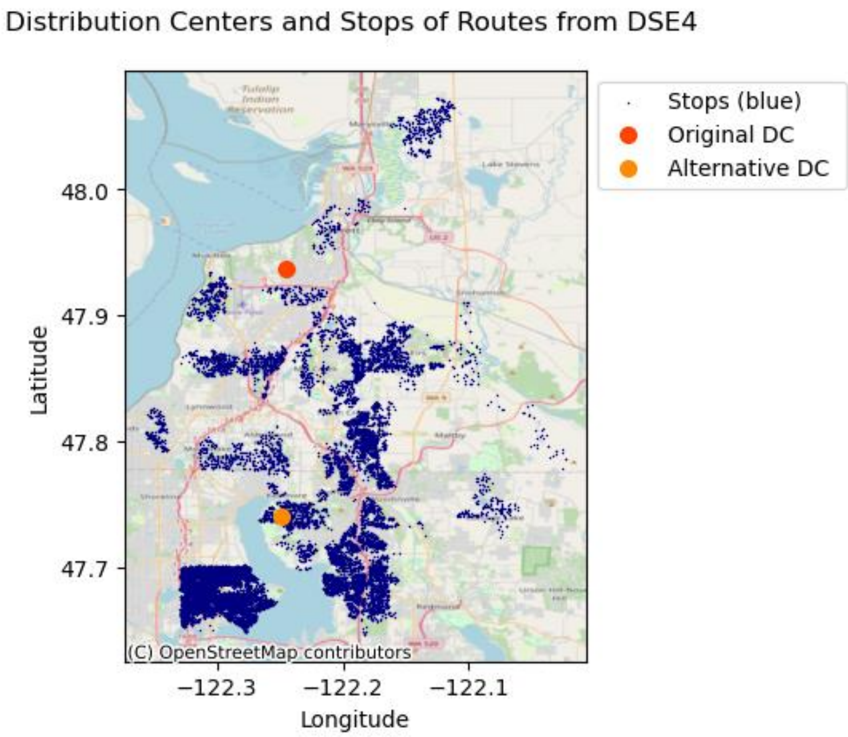


Figure 8. Illustration of the original and alternative DC in a case study, with the delivery stops (blue), the original distribution center DSE4 (red) and alternative drone distribution center (orange) (Prieto Camarillo, 2022)

2.3.3 Multi-delivery case

Multi-package delivery could be a further way to increase the efficiency of drone delivery (D3). As mentioned before, the selected drone (Wingcopter 198) has the capacity of carrying and delivering up to three packages with a triple drop system. This triple delivery can be carried out with a total payload of up to 5 kg. Ergo, the proposed scenario is to make use of this triple drop system and to model it as realistically as possible.

To model this multi-delivery case for the drone, a standard individual payload weight of 1.5 kg was chosen. Meaning that the fully loaded three package payload weight would amount to 4.5 kg in total.

The basic idea of the distance calculation with multi-delivery is illustrated on the right side of Figure 9. There is a distinction between double and triple package delivery, but in general multi-delivery consists in the following steps:

- First, the drone travels fully loaded, either with two or three packages, from the DC to the first delivery point and delivers the package.

- The next step is to travel to the following delivery point(s), however with one less package in each flight segment. This is repeated until all packages have been delivered to the intermediate stops.
- Finally, the drone flies back to the DC without any payload.

It is noteworthy, that the drone energy consumption for each of the flight segments will vary depending on the payload weight, having a different value in a fully loaded, half-loaded or empty drone. This is included in the energy consumption model of the drone, so instead of modelling one single delivery segment and multiplying it by two, the energy needed for each distance segment with the corresponding weights is calculated.

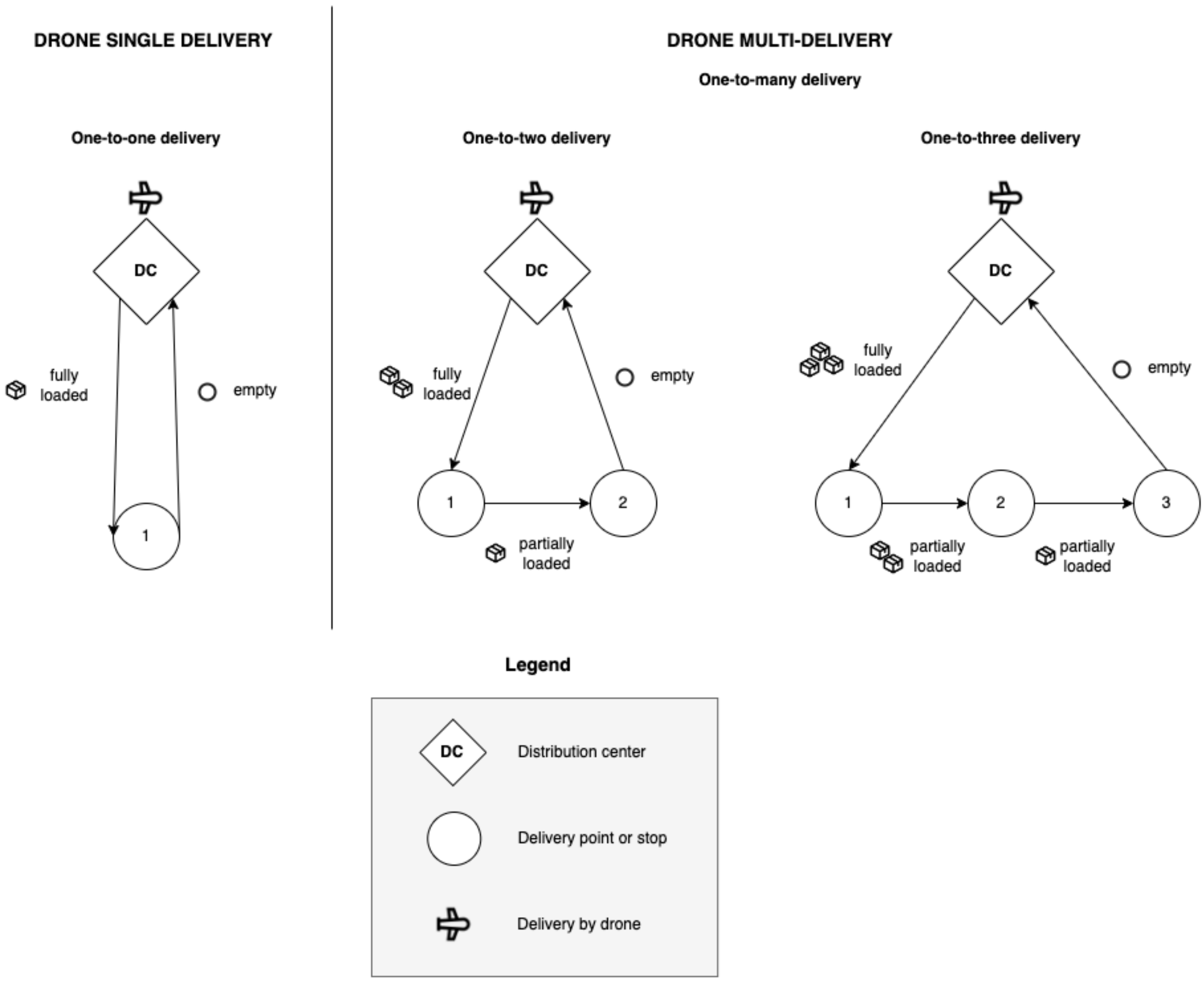


Figure 9. Illustration of the difference between single and multi-delivery in the drone case

A further relevant factor is the maximum drone range in km. The range is impacted by two main factors in this study: 1) by the maximum battery capacity and 2) by the amount of intermediate deliveries within one flight. Regarding the first factor, the maximum battery capacity of the Wingcopter drone is specified at 80% capacity in this study with 1304 Wh.

On the other hand, the number of intermediate deliveries is a significant constraint. This is because the hover mode, which refers to the descent and ascent when a package delivery is carried out, requires considerably more energy than the cruise mode. Due to the lift force generated from the wings that is present in the cruise mode, less energy is needed to counteract the force of gravity during the flight.

Given the finite energy budget of the battery, it could not be assumed that all deliveries could be accomplished with triple delivery. For this reason, a variable delivery case was modeled, which would decide whether a single, double or triple delivery could be carried out. This decision was made depending on the distance that needed to be covered and considering the battery size constraint. In certain cases, single delivery was still required due to the long-traveled distances.

This variable delivery scenario was modeled in a sequential manner. Meaning that the drone delivers to the next delivery point in the sequenced list, assuming that that point would be in close proximity to the previous one. In this way, the coordinate stops within one route were assigned into groups of 1, 2 or 3 packages per drone flight.

For instance, if a truck route served 167 stops, the delivery points were sequentially assigned into the individual groups which would represent one drone flight. Each drone flight would need to consume less energy than the 1304 Wh, which were deemed as the maximum energy consumption available. In this example, this route would have n groups of 1, 2, or 3 packages.

2.3.4 Combination of alternative DC and multi-delivery case

The last proposed case (D4) combines D2 and D3, which consist in the alternative DC location and the multi-delivery. Therefore, the variable delivery model is used, and instead of employing the original DC coordinates, the alternative DC coordinates are used. In this way, the two proposed distance-reducing measures are combined in one case.

2.3.5 Electricity-mix decarbonization scenario

The three proposed drone cases (D2, D3, D4) consist in the exploration of technical routing measures in the delivery system design. However, in order to explore the influence of mere technological measures, a future decarbonization scenario of the employed electricity mix was suggested.

A report by the UC Berkley’s Center for Environmental Public Policy, in which a clean electricity mix in 2035 is explored, assumed a share of 90% clean electricity case of the grid for the United States (Wooley et al., 2020). This study assumed the usage of non-biomass renewables, mostly wind as well as solar energy.

Employing this value, the carbon intensity of the electricity-mix decarbonization scenario would correspond to 44.3 g of CO_2 per kWh (see Table 10). This value is specifically relevant for the energy consumption of the EV and the drone and does not affect the operational phase of a diesel vehicle.

It surely can be that the decarbonization is not as prominent as 90%, nevertheless this scenario is meant to explore the effect that a possible decarbonization of the electricity mix would have in the future.

	Original electricity-mix	Decarbonization scenario
Carbon intensity per kWh	443 g CO_2	44.3 g CO_2
Source	EPA, national average for 2019	Assumption of a 90% electricity-grid decarbonization

Table 10. Electricity-mix decarbonization scenario

2.4 Implementation of the model in Python

This section provides an overview of the technical implementation of the previously explained delivery models in *Python*. An explanation of the code in overview terms, as well as the encountered challenges and the tools or packages used in the model, is found in this section. The goal is to provide a technical understanding of the functionality of the backend, as the code development constituted a significant component of this master thesis. The main sections of the code can be found in the appendix section B.

The model was developed and programmed on Python (version 3.8). Python is a widespread programming language, often used for purposes such as data analysis or scientific research. *VS code* was used as a source-code editor. Additionally, *conda environments* were used to manage Python packages. The list of the packages in the employed conda environment is found in appendix section B.5.

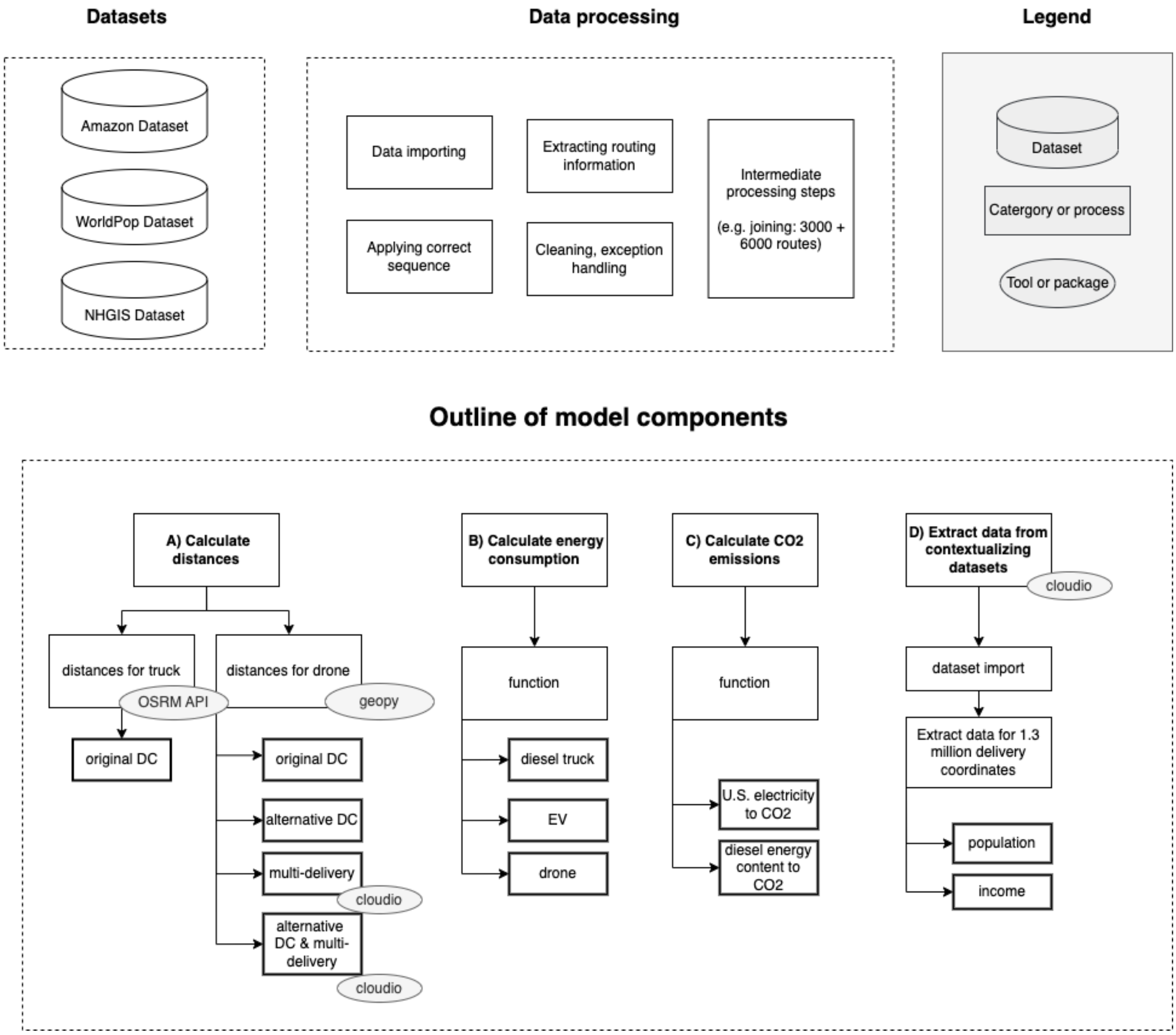


Figure 10. Outline of the main processes in the Python code, the datasets employed as well as the data processing elements

The various elements of the model can be summarized in Figure 10. There are three main parts to the model:

- The mainly employed Amazon dataset, which was described in section 2.1. In addition, two other datasets were employed to contextualize the delivery data in a secondary step.
- The Data Processing stages, which refer to all the initial cleaning processes as well as any intermediate merging or cleaning processes for dataframes.
- The Model Structure itself, which is divided into three main stages of distance calculation, energy consumption calculation and CO_2 emissions calculation. Additionally, the contextualization with demographic data is included in the last step.
- Moreover, relevant tools or packages employed are shown in gray circles.

2.4.1 Data Processing

Data extraction from the Amazon Dataset

Firstly, a script to extract the necessary routing data from the Amazon dataset was developed. This data includes the route ID, DC code, stop coordinates and observed sequence of the stops. The dataset is found in a JSON format that was imported using the *pandas* library in Python. Most of the further processing was carried out with pandas dataframes.

Route sequencing

There were two parts of the available datasets that needed to be joined. The first dataset had the necessary information such as the geographical coordinates of the stops and their corresponding route ID and DC code, however these stops were in an unordered sequence. The second dataset had the actual observed sequence of the stops e.g. the first delivery point had the sequence number 1. Therefore, these two datasets were joined. Subsequently, the stops were sorted in the right sequence, before starting the analysis. This initial stage of the dataset import and pre-processing was developed during the fall research project in order to access a small fraction of the routes (Prieto Camarillo, 2022).

Intermediate processing steps

On the left side of the Figure 10, the Amazon dataset is illustrated. As mentioned in section 3.2.3, the Amazon dataset is divided into two parts. Namely, a dataset with approximately 3,000 routes and a dataset containing approx. 6,000 routes. These had to be joined and/or filtered in later intermediate stages.

2.4.2 Model structure

2.4.2.1 Distance calculation

After the initial loading, sorting and cleaning of the data, the distance calculation for the original DC case and the alternative DC was programmed.

This was carried out in a script that looped over a given list of DC codes (see code in appendix B.1). To facilitate the distance calculation process and evade the API bottleneck for distance calculation that will be explained in the following sub-section, the script performed the calculation in parts. This meant that instead of running the script for all DCs, they were calculated incrementally in smaller groups.

In the fall research project, this gradual distance calculation approach was unnecessary as it focused on one distribution center with 166 associated routes. However, significant modifications of the script were necessary to scale the code for all DCs, given the substantial increase to more than 9,000 routes, 18 distribution centers and around 1.3 million coordinate points.

2.4.2.1.1 Distances truck

In order to calculate the truck driving distances, the *API* (Application Programming Interface) employed was *Open Source Routing Machine* (OSRM). OSRM is a routing engine employed to estimate the most optimal path in road networks. The input for the API request were the ordered geographical coordinates of the routes. The result of the API request came in a JSON file from which the total distance needed to be extracted from.

Ground distance calculation bottleneck

Using the API to calculate routes with over a million stop coordinates in total, resulted in a bottleneck in processing time, given the finite OSRM server capacity and traffic levels. For this reason, a locally hosted server of OSRM was set up. This meant not relying on the API servers, but on a local machine. However, this measure did not significantly speed up the request processing time. Therefore, the truck distances were calculated gradually, sending the API requests of 2 or 3 distribution centers at a time, as to not overload the API traffic with excessive requests.

Exception handling of distance calculations

Some routes were not able to be calculated by the API. Presumably some smaller roads might have not been registered in the employed road network, leading to some routes lacking a distance path from one point to the next one.

In total seven routes were not able to be calculated with OSRM. These routes were located in Los Angeles, departing from the distribution centers DLA7 and DLA4. In terms of stops in each route, this was equivalent to 886 stops that were not able to be calculated (see Table 11 and appendix A.2).

However, this represents a negligible percentage compared to the totality of the stops. These exceptional route IDs, for which a full distance of the route was not able to be calculated, were removed from all further processing.

	Total contained	Exceptions: non-explored	Total calculated
Routes	9,164	7	9,157
Stops	1,337,758	886	1,336,872

Table 11. Exception handling in calculating ground distances with OSRM

2.4.2.1.2 Distances drone

The drone distances were calculated as point-to-point distances, however, given the spherical surface of the globe, these distances cannot be calculated as a straight line. For this reason, the great-circle distance calculation was utilized by the *GeopPy* package using the WGS-84 ellipsoid projection (GeoPy Contributors, n.d.).

The geodesic distances were calculated departing for the distribution center to each of the stops contained in the route and back. As an input, the function employed required the two sets of coordinates, the distribution center and the individual stop. This resulting distance was then multiplied by two, in order to represent the round trip by the drone.

For the alternative DC scenario, a different set of departure coordinates was used in place of the original DC coordinates. The original as well as the alternative DC coordinates can be found in appendix A.1.

2.4.2.2 Calculation of energy consumption and CO_2 emissions

After calculating the ground and flight distances, these needed to be expressed in terms energy consumption and CO_2 emissions.

Since the translation into energy consumption and CO_2 emissions emulates that of a formula, this was carried out using *functions* in Python.

Functions can be defined as a section of code or set of instructions meant to be utilized repeatedly, which return a specific result and that can be called within the code whenever needed (Willems, 2020).

For this reason, there was a function for each conversion. Which took as an input, the distance calculation of either the drone or the truck and then converted it into energy consumption values depending on the vehicle. Subsequently this energy consumption value was converted to CO_2 emissions, with another set of functions.

In total there were five main utilized functions, which can also be found in appendix section B.2:

- Drone energy consumption from distance
- EV energy function from distance

- Diesel vehicle energy contained function from distance
- CO_2 emissions function for diesel vehicle
- CO_2 emissions from electricity mix for drone and electric vehicle

2.4.2.3 Drone multi-delivery

Given that the drone battery constraints needed to be considered, the modelling of the multi-delivery scenario required a higher complexity than the initial distance calculation for a single drone delivery.

The multi-delivery script can be found in appendix B.3 and it classifies routes in groups of one, two or three deliveries, depending on the use of energy needed to make those deliveries.

Classification into delivery groups

The main approach to build the groups of deliveries, was to calculate the energy needed to make the delivery in a sequential manner. This sequential approach was chosen under the assumption that the observed sequences in the Amazon dataset would go to the next closest stop. This meant calculating the distance and the energy needed to make the first delivery. Subsequently, if there was remaining battery capacity, then the distance and energy consumption to go to the next delivery point was calculated. If after the second delivery there was remaining battery capacity, the last calculation for the third delivery was carried out. If at any point during the sequential calculations, the energy needed exceeded 80% of the drone battery capacity, then the previous value in which the upper limit was not exceeded, was taken.

This was implemented through a set of *if*, *elif*, and *else* conditions. In order to determine how many deliveries were possible, two exit conditions were defined: 1) if the energy expended in the current trip exceeded 80% of the current battery capacity of the drone and 2) if already three deliveries were carried out by the drone.

Additionally, the function needed to calculate the energy needed to be adapted for a variable payload and several segments of distances.

Utilization of a virtual machine

To optimize long processing times, the script was run in a *virtual machine* (VM), which had a higher computational capacity. Virtual Machines are a virtual representation of a physical computer (IBM, n.d.). *Cloudio* is the VM is available for research within the Industrial Ecology Program. Cloudio has 88 virtual central processing units (vCPU) and 1280 GB of RAM (Industrial Ecology Digital Lab, 2023).

Cloudio can be run directly in the Linux terminal and it has an additional interface with visual studio code. The same script was run for a group of approximately 3 to 4 distribution centers in 5 separate vCPUs.

2.4.3 Processing of the contextualizing datasets

As described in section 2.1.4, there were two extra contextualizing datasets in order to shed a light on the demographic characteristics of the areas covered by the Amazon dataset. The employed dataset for population density was from WorldPop, while the dataset with information about income per capita was from IPUMS NHGIS.

Especially for these larger datasets, the usage of cloudio was necessary. A higher computational capacity was required, given the 1.3 million coordinates for with the script needed to extract data from the two additional datasets.

Population density dataset

As for the WorldPop dataset, the processing time in the standard computer was time consuming, with 1.5 minutes for 20 coordinates. For this reason, the 1.3 million coordinates were divided into five dataframes and the data was extracted for each section simultaneously in separate vCPUs in cloudio.

The format of the WorldPop dataset was a *.tif* file. A GeoTIFF file contains geodata, in the form of a raster or pixel-like structure (Wasser et al., 2018). This geographical data is stored in the so-called "bands" of the .tif file.

Firstly, the .tif file was loaded with the help of the *rasterio* package. Posteriorly, the population density values for the 1.3 million delivery stop coordinates was extracted from the corresponding raster in the dataset. The code for the population data extraction can be found in appendix B.4.

Income dataset

Regarding the income data extraction from IPUMS NHGIS, there was an intermediate processing step needed, because of the two data files. Therefore, the first step was to join the shapefile that contained the geographical information of the ZIP code areas, together with the CSV file that contained the income per capita data.

This merged information was then exported as a new shapefile, which was used to extract the corresponding income values for the 1.3 million geographical coordinates of the Amazon dataset. The script for the income data extraction is located in appendix B.4.

3 Results & analysis

This section presents the results of the analysis carried out with the ground vehicle and drone delivery models.

Firstly, the dataset is contextualized in terms of demographic characteristics such as population and income. Secondly, an overview of the results per simulated case is given in terms of the most important measurement criteria: distance, energy consumption and carbon emissions. Thirdly, an equivalent-route analysis is given in order to compare the performance of the same data points or routes. Furthermore, additional insights are given regarding the multi-drop scenario. Lastly, the technological measure of the decarbonization of the electricity mix is presented.

3.1 Contextualization of the data

As a part of the contextualization of the Amazon dataset, further demographical data in terms of income and population density was obtained. This served as ab assessment of the representativeness of the data and to provide information of where the dataset stands in comparison to the U.S. average.

In Figure 11, each blue cross represents the average value corresponding to each of the 9,157 routes. On the horizontal axis, the average income per capita for the corresponding ZIP codes of the route is shown. Along the vertical axis, the average population in people per km^2 per route ID is depicted.

The green horizontal line represents the average population value in the U.S., which amounts to 36.2 people per km^2, this value was calculated based on the area and total population values (U.S. Census Bureau, 2022; World Bank, 2023). On the other hand, the vertical orange line represents the average income per capita, amounting to 37,638 USD according to the 2021 American Community Survey (U.S. Census Bureau, 2022).

It is apparent from the figure, that the majority of the of the routes can be found above the green average population density line. This indicates that the majority of the delivery routes included in the Amazon dataset, take place in areas that are more densely populated than the U.S. average.

According to the most recent U.S. Census Bureau definition (2022), an urban area consists, among other criteria, mainly in a cluster of more than 2,500 or 5,000 people. For reference, the city of Los Angeles possesses an estimated population density of approximately 3,292 people per km^2, while the city of New York approximately of 10,812 people per km^2 (U.S. Census Bureau as cited in Open Data Network, 2018). As can be observed by the cluster of blue crosses a at the bottom of the figure, more routes take place in areas with a population density similar to that of Los Angeles rather than New York.

Regarding income per capita on the horizontal axis, it can be observed that some routes are found to the left of the average U.S. income line. However, a larger proportion of the routes is located to the right of the orange average line. This means that while there is a proportion of the routes in the dataset that deliver in areas with a socioeconomic level below the average income, the majority of the deliveries are carried out in areas with a higher than average income per capita.

It is necessary to consider these two aspects, when determining for which context the results are valid for. In this sense, the results that are presented in the following sub-sections apply, firstly, in the context of U.S. urban areas. Secondly, it needs to be considered that a large proportion of the routes were carried out in areas with an income higher than the U.S. average.

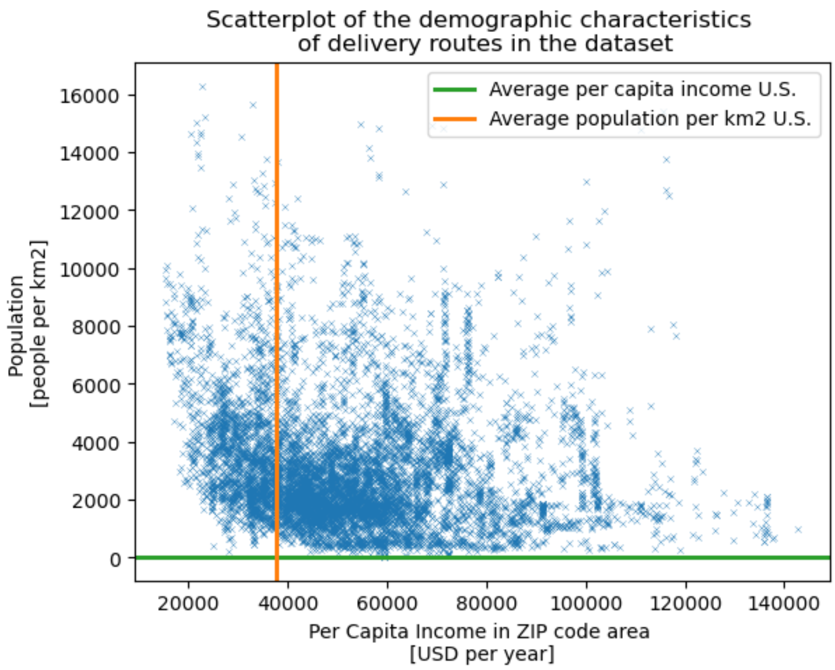


Figure 11. Scatterplot of the demographic characteristics of the delivery routes in the dataset

3.2 Results per modeled case

In this subsection, an overview of the results for each modeled case will be given in terms of distance, energy consumption and carbon emissions.

As for the modeled cases, these were explained in detail in section 2.3. Firstly, the baseline cases correspond to the ground vehicles departing from the original distribution center (G1, G2). Additionally, there are four drone cases: one departing from the original DC (D1), another departing from the alternative DC (D2), one implementing the multi-delivery of up to three packages (D3) and, lastly, the case combining multi-delivery and the alternative DC (D4). This last drone case will be referred to as the "combined" drone case (D4).

To see the complete generated dataset from which the summary statistics were obtained, consult the CSV file provided as supplementary material.

3.2.1 Distances (km)

To understand the energy consumption and carbon emissions sections, the distance calculation results need to be described first. Table 12 presents an overview of the distance and distance ratio for every modeled case. These are the averages for all route IDs in the dataset across all cities and distribution centers.

There is no distinction between EV and diesel vehicles in the table, given that both share the same traveled ground distances, which are referred to as "Truck: Original DC" (G1, G2). This baseline ground transportation case is shown in blue. Below that, the four calculated drone cases can be found (D1, D2, D3, D4).

On the first column of the table, the average traveled distance in kilometers for each case can be found. While on the next column, the distance ratio between drones and ground vehicles is provided.

Regarding the distance per route, the drones execute different flights to deliver each package—or in the case of multi-delivery, a cluster of up to three packages. However, to achieve the comparability between truck and drone distances in the analysis, the drone distances that cover the equivalent number of truck stops per route were aggregated.

The distance ratio is an important measure to analyze the distances flown by the drone in comparison to the distances traveled by the ground vehicle. In this study, the distance ratio is defined as the kilometers that a drone is required to fly compared to every kilometer driven by the truck to deliver the same amount of packages. In this manner, the original DC of the truck is used as the comparison point or baseline.

In the case of D1, for every km the truck drives, the drone would need to fly 49.6 km on average to satisfy the same amount of package deliveries within that established route. This corresponds, on average to 5,222.5 km that the drone would need to fly compared to the 100.2 km driven by the truck to deliver all the packages within one route ID.

Regarding the alternative distribution center case with the drone (D2), there is a reduction of the mean distance ratio from 49.6 departing from the original distribution center, to 42.7 departing from the alternative depot location.

Mean distance per route [km] and Mean distance ratio [km drone: km truck]		
Case	**km**	**km drone: km truck**
(G1, G2) Truck: Original DC	100.2	-
(D1) Drone: Original DC	5,222.5	49.6
(D2) Drone: Alternative DC	4,459.7	42.7
(D3) Drone: Multi-delivery	3,076.1	26.9
(D4) Drone: Alternative DC & multi-delivery	2,355.9	21.1

Table 12. Mean distance per route [km] and Mean distance ratio [km drone: km truck] for all modeled cases, baseline case shown in blue

Interestingly, the most significant reduction compared with the original DC drone case, can be observed with multi-delivery (D3) where the average distance ratio decreases to around 26.9. This indicates that multi-delivery is a significant measure to reduce the distances flown by the drone in comparison with the original distribution center case. This reduction is highly important, since for every extra package the drone delivers it saves one trip back and forth from the distribution center. This makes drone delivery considerably more effective than with a single delivery.

In the case of the combined scenario (D4), multi-delivery and alternative DC, a further but not so large stepwise reduction can be observed from 26.0 to 21.1 drone km per truck km. This was found to be the maximum average reduction. In this case, the average flown distance of the drone was ultimately reduced from 5,222.5 km to almost less than half that value to 2,355.9 km. This further improvement in distance ratio can be traced to the combination with the alternative DC, a closer alternative depot location enables more multi-deliveries of two or three packages.

3.2.2 Energy consumption (kWh)

Based on the distance calculations, the equivalent energy consumption for each case is calculated. The energy consumption models use corresponding conversion factors for each vehicle type (see section 2.2).

On Table 13, the average energy consumption per route (right) and per package (left) are shown. To have the distance normalized by the number of packages allows a comparison of the same unit. This is because every route is composed of a varying amount of packages, implying that different routes are not necessarily directly comparable. Thus, it is important to show these two indicators and not only the energy consumption per route.

On the table, the baseline scenarios departing from the original distribution center for the EV and diesel vehicle are shown in blue (G1, G2).

Focusing on the cases that have an energy consumption that is less than the diesel vehicle average consumption (G2), the last two cases fall under this category (D3, D4). These are highlighted in green. The diesel vehicle has an average energy consumption of 0.60 kWh per package, while the drone multi-delivery and combined case an average consumption of 0.55 and 0.48 kWh per package respectively.

In the case of the drone alternative DC case (D2), a reduction from 0.70 to 0.62 kWh per package can be observed compared to the original drone DC case. While the energy consumption of 0.62 kWh is relatively close to the diesel average consumption of 0.60 kWh per package, this single measure does not suffice to reach lower energy consumption levels than the diesel vehicle.

It becomes apparent that none of the drone cases would be able to contend with the EV baseline case (G1), which has an average energy consumption of 0.28 kWh per package. In the last drone combined case, the average consumption per package would need to be reduced by at least 0.2 kWh in order to reach the EV average consumption levels.

Average energy consumption per route [kWh/route] and Average energy consumption per package [kWh/package]

Case	kWh/route	kWh/ package
(G1) EV: Original DC	40.2	0.28
(G2) Diesel: Original DC	85.5	0.60
(D1) Drone: Original DC	101.7	0.70
(D2) Drone: Alternative DC	91.8	0.62
(D3) Drone: Multi-delivery	80.0	0.55
(D4) Drone: Alternative DC & multi-delivery	71.4	0.48

Table 13. Average energy consumption per route [kWh/route] and Average energy consumption per package [kWh/package] for all modeled cases. Baseline cases shown in blue, while the cases with an average value less than scenario G2 are shown in green

In order to explore the distribution of the data, a boxplot of the energy consumption can be seen in Figure 12. The variation of the data is the largest in the case of the drone cases, especially in both the original and the alternative distribution center cases (D1, D2). Of the drone cases, the combined case (D4) is the one with the least variation. It can be observed that multi-delivery (D3, D4) tends to reduce the variation in the energy consumption data.

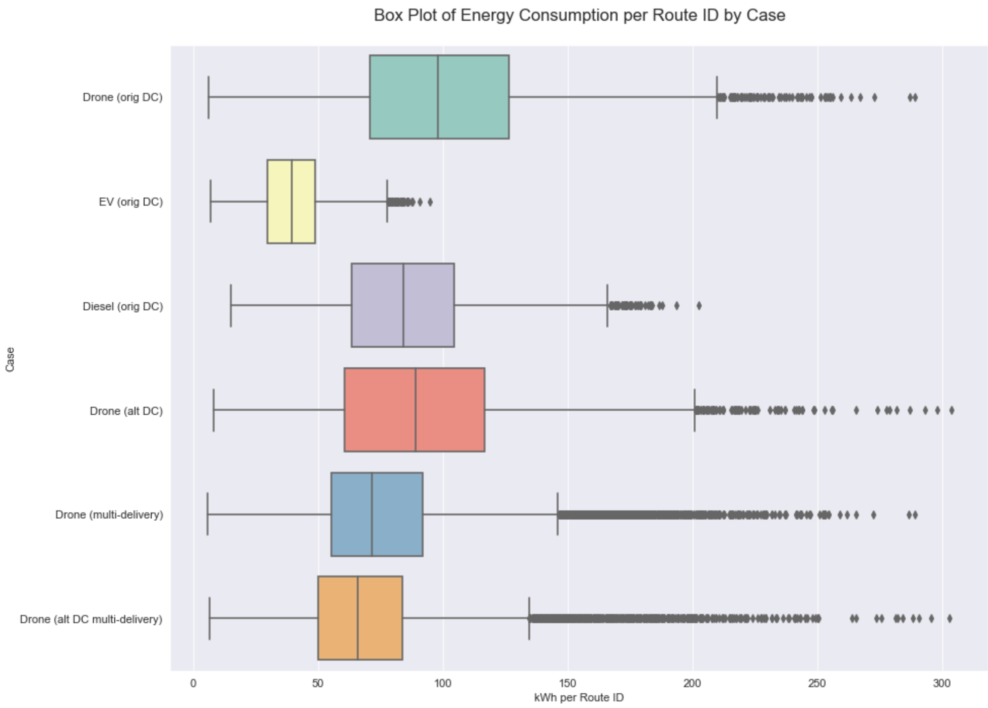


Figure 12. Boxplot of the energy consumption in kWh per route ID for each case

3.2.3 Carbon emissions (kg of CO_2)

Based on the energy consumption values, the corresponding conversion factors were used to obtain the carbon emissions for each route (see Table 14). Additionally, carbon emissions per package are shown on the right side of the table, given that each route varies in amount of packages.

What is interesting compared to the energy consumption table, is that the last two scenarios do not perform better than the average diesel case in terms of carbon emissions, whereas the opposite was the case with energy consumption. These two scenarios were shown in green on the previous Table 13.

These diverging results can be traced back to the conversion factors. In the case of diesel, the conversion factor of 0.326 kg CO_2 /kWh is lower than the carbon intensity of the average U.S. electricity mix, which equals to 0.433 kg CO_2 /kWh.

Compared to the employed electricity mix, the diesel vehicle has relatively low values for carbon emissions. However, diesel technology possesses a limited potential for decarbonization. A leverage of electricity-powered vehicles is the greater probability of decarbonization for the electricity mix. A scenario considering the electricity decarbonization potential is shown in section 3.5.

Average carbon emissions per route [kg CO_2/route] and Average energy consumption per package [kg CO_2/package]		
Case	**kg CO_2/route**	**kg CO_2/package**
(G1) EV: Original DC	17.4	0.12
(G2) Diesel: Original DC	27.9	0.19
(D1) Drone: Original DC	44.0	0.30
(D2) Drone: Alternative DC	39.7	0.27
(D3) Drone: Multi-delivery	34.7	0.24
(D4) Drone: Alternative DC & multi-delivery	30.9	**0.21**

Table 14. Average carbon emissions per route [kg CO_2/route] and Average energy consumption per package [kg CO_2/package], baseline cases are shown in blue

In order to better visualize the distribution of the CO_2 emissions per case, histograms with kernel density estimation (KDE) lines are provided for each case in Figure 13. Along the horizontal axes, the values are classified into bins according to the level of CO_2 emissions. In the vertical axes, the frequency of occurrence of that emission bin is shown.

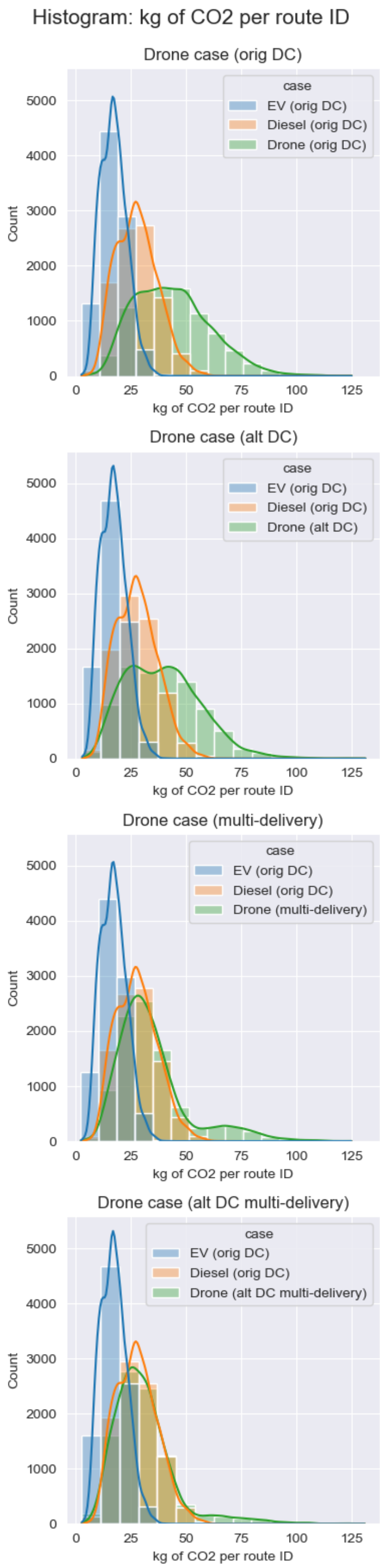


Figure 13. Histograms with KDE lines of kg CO_2 emissions per route ID. Each drone case in each panel (D1, D2, D3, D4) is shown in green. While in all panels, the ground vehicle baselines (G1, G2) are shown in blue and orange.

Each panel shows one of the modeled drone cases (D1, D2, D3, D4), which are depicted in green. As can be observed in all panels, the blue and green lines (G1, G2) stay in the same position, since they correspond to the baseline cases.

It can be observed that the green KDE line of the drone cases shifts closer to the left, increasingly overlapping with the EV case in blue. This occurs as the lowest panel is reached and more routing measures are implemented for drone delivery.

This overlap between the green and blue bars means that the performance of some of the drone routes could reach the same levels of some of the EV routes. However, it is relevant to note that this histogram contains all route IDs. A same-point comparison, analyzing the equivalent route IDs is carried out in the subsequent section.

3.3 Equivalent-route comparison

In subsection 3.2, an aggregate perspective of all routes in the dataset was presented. That section examined the overall performance of all routes in the dataset; however, it did not necessarily compare the same route ID for each of the cases. This subsection compares the CO_2 performance of the drone routes compared to the diesel and EV baseline cases, based on the equivalent route or datapoint. This is relevant, considering that different routes have different sets of packages to be delivered and would be different comparison points.

It is to be mentioned, that although the set of stops is the same, the departing distribution center might be different when comparing a drone case departing from the alternative DC.

Table 15 shows the number of drone routes per case with a better performance compared to the equivalent routes in the EV and diesel cases (G2, G1), which depart from the original depot. This performance refers the carbon emissions in kg of CO_2 per route ID. Additionally, the proportion of better performing drone routes in comparison with the total 9,157 routes is shown.

Better performing drone routes in terms of CO_2 per route ID					
Cases	**Compared with Diesel vehicle (G2)**	**% of total routes**	**Compared with EV (G1)**	**% of total routes**	**Total routes**
(D1) Drone: Original DC	191	2%	1	0%	9,157
(D2) Drone: Alternative DC	1,447	16%	71	1%	9,157
(D3) Drone: Multi-delivery	1,582	17%	19	0%	9,157
(D4) Drone: Alternative DC & multi-delivery	3,546	39%	269	3%	9,157

Table 15. Better performing drone routes in terms of CO_2 per route ID (equivalent-route ID comparison)

For instance, in the case of the original distribution center (D1), only 191 drone routes or 2% of the total routes, perform better when compared to the emissions of the diesel case (G2) for the same route ID. When considering the comparison with the EV case (G1), even fewer drone routes perform better, with only one route emitting less carbon emissions.

For the comparison with the diesel case, it can be observed, that the alternative DC measure (D2) brings a similar improvement as does the multi-delivery case (D3). Meaning that around 1,500 routes would perform better with the drone compared to the diesel vehicle. Further, the most improvement can be seen with the alternative DC multi-delivery case (D4) with 3,546 routes performing better, which are equivalent to a significant share of 39% of the total 9,157 routes.

As to be expected from the aggregate analysis, there are very few drone routes that perform better compared to its EV route counterpart (G1). However, as the measure of alternative DC is implemented with multi-delivery this number increases to 269 better performing routes. Compared to the total routes, the better performing drone routes are still few compared to the EV routes. This is due to the low distance that a ground vehicle needs to travel in comparison with the drones, therefore ending up with low carbon emissions compared to the drone counterparts.

3.3.1 Equivalent-route comparison of D4 with G2

Given that the D4 case had the most available best performing drone routes compared with the diesel baseline case (shown in green in Table 15), a more in-depth analysis was carried out for this case. The aim of this section is to explore the general characteristics of these 3,546 better performing drone routes. A set of summary statistics comparing the better and worse performing routes can be found in appendix C.2.

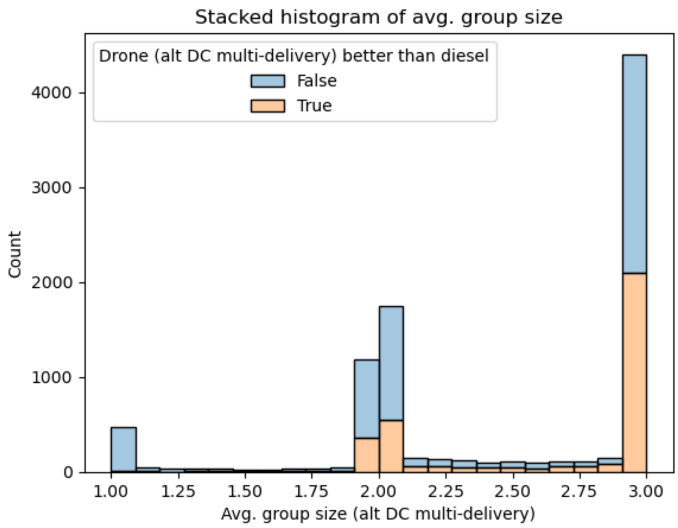


Figure 14. Stacked histogram of average group size per flight in the alternative DC multi-delivery (D4) case. In blue are the worse performing routes and in orange the better performing routes compared to the G2 case.

Firstly, the group size attribute was explored. Figure 14 presents a stacked histogram of the average group size for the alternative DC & multi-delivery drone case (D4). Group size refers to the amount of intermediate deliveries carried out within one single drone flight. The blue fragment of the bar represents the worse performing routes, while the orange is equivalent to the better performing routes compared to the diesel case for the equivalent route IDs.

It can be observed that the better performing drone routes (orange), tend to have 2 or 3 packages per flight. A larger proportion of routes tend to perform better than their diesel counterpart when 3 deliveries are made.

Since there are also other routes with either 2 or 3 package deliveries that are not within the better performers (blue), it can be concluded that group size is not the only determining factor to determine whether a route could perform better or not than the diesel baseline case. However, none of the routes with single-package delivery, performed better than the diesel baseline scenario. Therefore, it is apparent that multi-delivery is a key factor to improve drone performance in terms of CO_2 emissions.

Looking for further factors that might be related to the performance of the drone routes, various variables were plotted against one another. This was done with so-called pairplots of variables, which serve to visualize pair-wise relationships. The most relevant pairplots are shown in this section, for the complete set of pairplots consult appendix C.3.

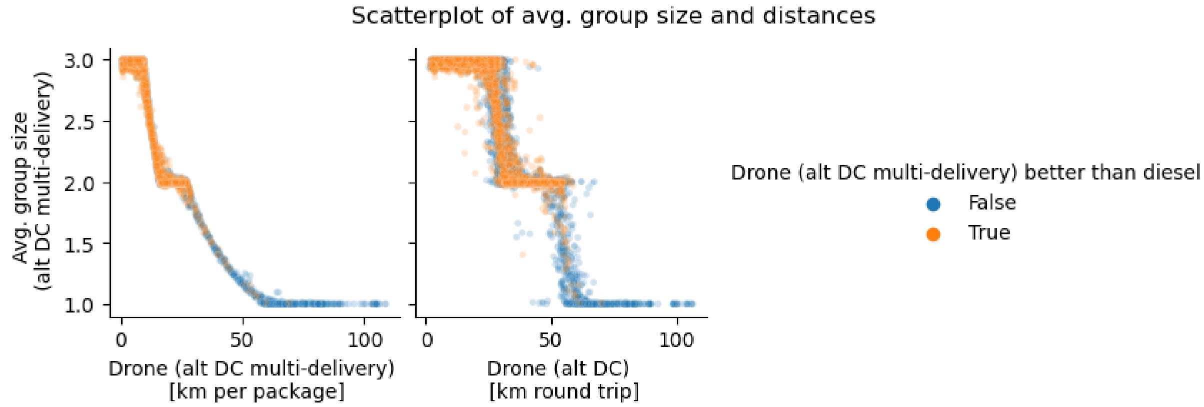


Figure 15. Scatterplots of average group size and km flown per package in the case of D4, alternative DC & multi-delivery, (left) and the average round-trip distance from the alternative DC to the delivery stops (right)

Figure 15 shows two scatterplots of the relationship between average group size and two different types of distances: km flown per package in the case of alternative DC & multi-delivery (D4) and the average distance from the alternative DC to the delivery stops. Both panels show that most of the better performing drone routes, tend to have 2 or 3 delivery stops per flight, as discussed in the previous Figure 14. However, these scatterplots indicate that there is an additional relevant parameter other than the group size, namely the distance flown per package and the distance between the DC and the respective stops.

In the case of the left panel, drone flights with two stops or more have a distance per package of around less than 30 or 40 km. It can also be observed that the better performing routes (orange) are mostly found within this range of higher group size and lower distance per package.

Turning to the right panel, the average distance from the alternative DC to each of the stops can be observed on the horizontal axis. The right panel shows that shorter distances from the DC to the delivery points, allows more double and triple deliveries. This in turn, leads to better performing drone routes.

It can be observed that most of the better performing routes are found below a maximum round trip distance of 60 km in a single-delivery scenario, which would be equivalent to 30 km straight-point distance between the alternative DC to the delivery stop.

From these panels it can be concluded that a further influential factor apart from group size in multi-delivery, is the distance from the distribution center to the delivery points.

3.4 Multi-drop insights

Multi-drop has proven to be a highly relevant routing measure in the previous result sections. Firstly, this measure in combination with a closer departure depot enables better performing routes and has the biggest potential to reduce energy consumption and carbon emissions on average when comparing the drone cases. Additionally, as multi-drop was one of the novel aspects of this thesis, this section provides further insights on multi-delivery.

Group formation

Within the 9,157 routes, their 1.3 million coordinates were clustered into multi-delivery groups with varying group sizes depending on whether the trip could be completed with the determined 80% of the drone battery capacity. At the end, 659,620 multi-delivery groups were able to be formed. Of the formed groups the majority (44 %) of the drone flights were able to carry out double deliveries. On the other hand, 28% would have been able to perform a triple delivery, while 27% would have only been able to perform a single delivery.

Group Size	Amount of groups	Proportion (%)
Single delivery	179,233	27%
Double delivery	293,355	44%
Trible delivery	187,032	28%
Total	659,620	100%

Table 16. Group size of the drone flights in the multi-delivery scenario from the original DC (D3)

Insights from exemplary triple-drop calculation

To provide further insights into multi-drop, an exemplary calculation is described in this section. This example calculates the energy consumption for each flight segment, assuming the maximum of three deliveries. In Figure 16, the straight-line distance from the distribution center DAU1 in Austin and the cluster of three deliveries, corresponding to 27.7 km, is shown in black. Figure 17 shows a close-up of the distances between the three deliveries carried out by the drone. It can be observed that the distances between each intermediate stop is not so significant compared to the distance to the distribution center.

For a triple delivery, there are 4 segments in the trip:

- Segment 1: is the flight from the distribution center to the first stop coordinate.
- Segment 2: corresponds to the short distance traveled from the first stop coordinate to the next delivery point.
- Segment 3: corresponds to the distance between the second stop and the third.
- Segment 4: is the final flight back to the distribution center.

There are three intermediate landings, which have a different rate of energy consumption than the hovering flight mode in the previously described segments. During these segments, as seen in Table 17, the payload weight varies as packages are delivered in each intermediate landing. The difference in payload weight leads to the variation of the L/D ratio in the drone energy consumption model.

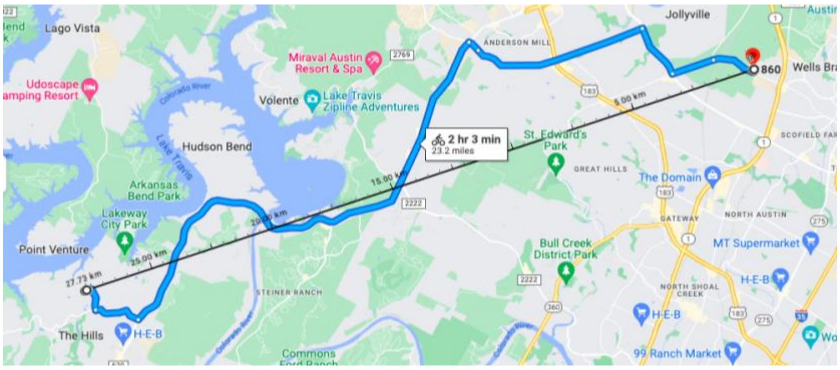


Figure 16. Straight-line trajectory between the distribution center in Austin and the cluster for triple deliveries (black line corresponds to a distance of 27.8 km)

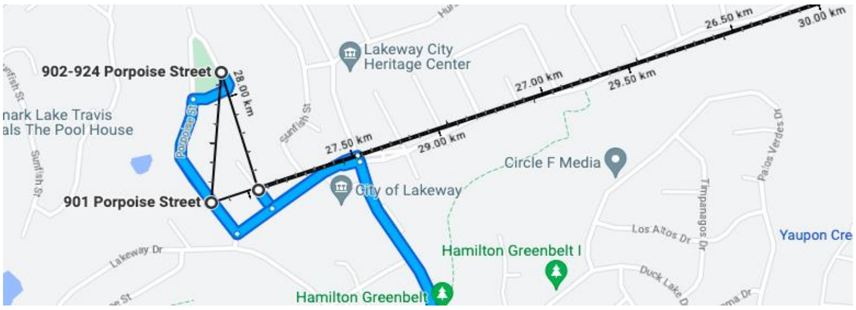


Figure 17. Close-up of the location of the three intermediate deliveries in the exemplary calculation

As can be seen in Table 17, the distances flown between intermediate deliveries are relatively small, in this exemplary case corresponding to 0.32 and 0.34 km. This implies, that if the total route distances between a single, double and triple delivery flight were to be compared, there would not be a significant difference. Delivering double or triple the amount of packages, almost doubles or triples the efficiency of drone delivery per package.

Exemplary triple drop delivery calculation by trip segment

Trip segments	Distance km	Payload weight at segment [kg]	Total weight at segment [kg]	LD ratio [unitless]	Distance energy [kWh]	Intermediate landing energy [kWh]
Segment 1	27.8	4.5	24.5	7.87	0.47	0.34
Segment 2	0.32	3	23	8.21	0.00	0.28
Segment 3	0.34	1.5	21.5	8.54	0.00	0.23
Segment 4	27.9	0	20	8.88	0.34	-
Total	56.4	-	-	-	0.82	0.85

Table 17. Exemplary triple drop calculation broken down to trip segments

As can be observed in Table 18, the energy required for an intermediate landing is greater than the energy required to fly the 56.4 km. Since the energy required for intermediate landings is substantial, distance flown is a limiting factor in determining whether a further delivery can be realized or not.

As can be seen in Table 18, this exemplary case would have had a maximum delivery capacity of two deliveries and not three. This is given that the 1.67 kWh exceed the maximum battery capacity of 1.3 kWh.

In short, this is what the model did for all the 1.3 million coordinates, in terms of assigning them into groups according to the maximum allowed energy consumption.

Breakdown of energy consumption [kWh]	
Phase	**kWh**
Energy distance	0.82
Energy intermediate landing	0.85
Total energy	1.67

Table 18. Breakdown of energy consumption per phase [kWh]

3.5 90% decarbonizartion scenario of the electricity mix

In addition to the technical routing measures explained in the previous subsections, the scenario of a decarbonization of the electricity mix in the U.S. was explored. This falls under the category of a technological measure, rather than a routing measure. In this scenario, the values for a decarbonization of the grid of 90% as explained in section 2.3.5 were used.

Figure 18 shows several histograms of operational carbon emissions employing two different electricity mix scenarios. The left column uses the values for the current electricity mix in the U.S., whereas the right column employs the values of a 90% decarbonized electricity grid.

The horizontal axes show the CO_2 emissions in kg normalized by packages per route. The hues represent the three different delivery vehicles: EV (blue), diesel (orange) and drone (green). The original DC case is shown in all graphs for the EV and the diesel vehicles, in order to serve as a comparison baseline. In contrast, the 4 different drone cases are shown in each row.

In the histograms, the values of carbon emissions are clustered into bins on the horizontal axis, while the vertical axes show the frequency of occurrence for each bin. Meaning, that the vertical axes show the count of routes, falling within a specific CO_2 emissions bin.

Firstly, a shift of the drone (green) and EV (blue) cases to the left of the plots can be observed for all 4 drone cases when employing the decarbonized electricity mix. As the carbon emission values become smaller, it can be seen that more routes fall into the same bin. However, this is a mere scaling effect, since the same distances are used for the calculation of the CO_2 emission values.

On the contrary, as can be seen in the orange bars from Figure 18, the diesel values stay constant. This is because the emissions coming from diesel in the delivery phase do not depend on the electricity mix and therefore are not affected by a decarbonization of the electricity mix.

The difference between the right and left column highlights that changing the electricity mix to a less-carbon intensive one, favors vehicles powered by batteries. As for the current package delivery system, when comparing drones (green) and diesel delivery vehicles (orange), a clear advantage can be seen in the case of drone delivery in terms of CO_2 emissions.

On the other hand, when looking at future delivery systems comparing drones (green) and EVs (blue), the histograms suggest that EVs would be favored, even if it is by a lesser order of magnitude than in the original electricity mix scenario. Even though EVs would be better in the decarbonized electricity scenario, the impact of both drones and EVs would be substantially smaller, when compared to the diesel case.

It can still be observed that multi-delivery is the drone case with the most potential for reduction in CO_2.

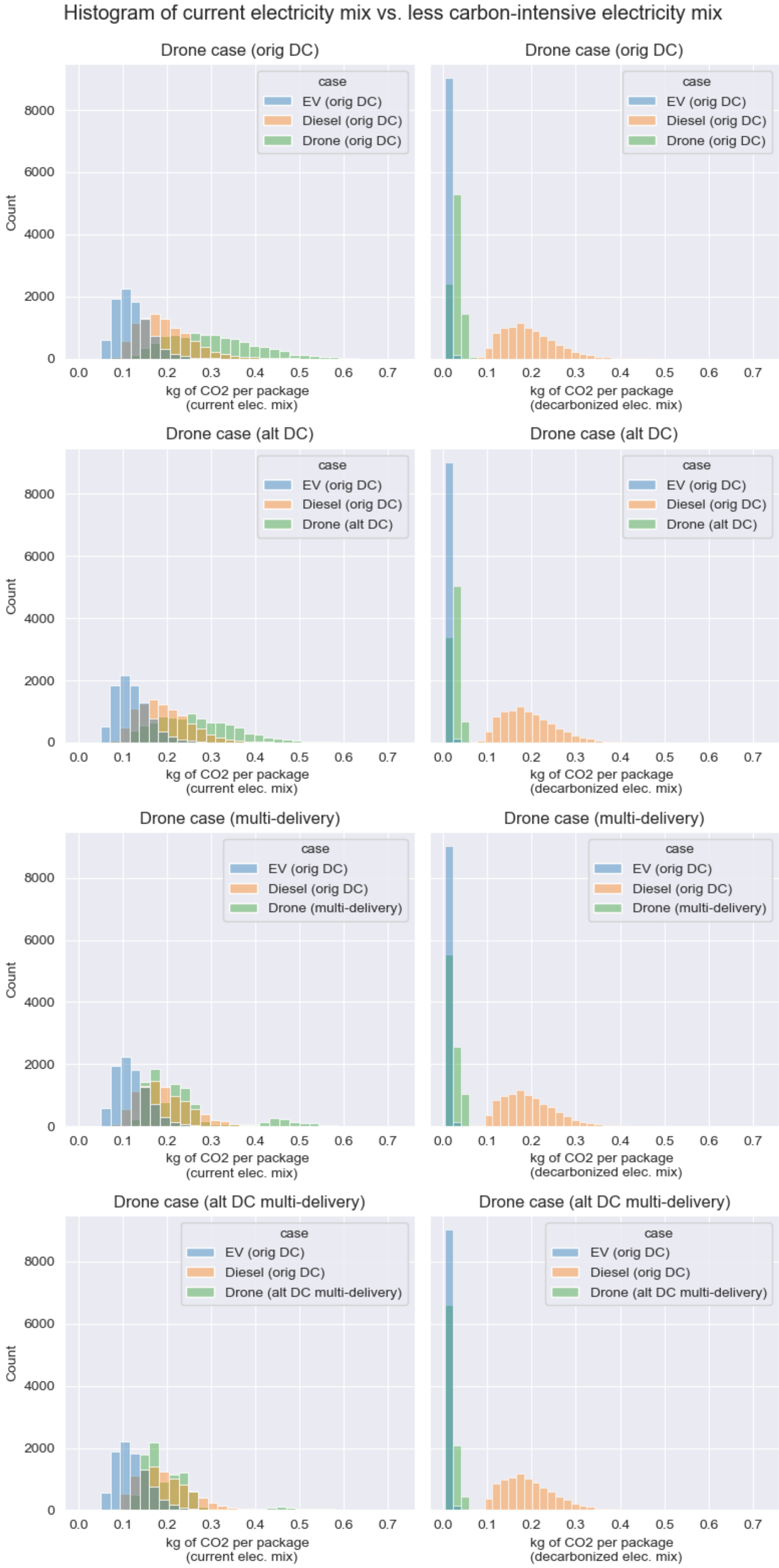


Figure 18. Histogram of carbon emissions for two electricity mixes, including all drone cases in each row, compared to the baseline EV (blue) and diesel (orange) cases. The left column refers to the current electricity mix and the right one to the decarbonized scenario.

4 Discussion

The aim of this study was achieved by evaluating and quantifying the operational environmental impact of widespread adoption of a drone versus a ground delivery system, in the context of last-mile home delivery.

The goal of this section is to discuss the results by a) providing an answer to the proposed research questions, while reflecting on the further implications of this research, b) setting the results of this study in the context of other works, c) analyzing and reflecting on the contributions and limitations of this study, and d) finally, giving pointers for future research.

4.1 Main findings and implications for drone delivery

Revisiting the research questions posed in section 1.3.2, RQ1 referred to the environmental operational performance of drone delivery systems compared to current & future ground-based delivery systems. RQ1 was analyzed by the modelling of the delivery systems themselves, specifically by the baseline cases (G1, G2, D1) departing from the original distribution center.

RQ2 addressed possible measures to improve the performance of drone delivery systems, referring to the decarbonization of the delivery operation/transportation phase. RQ2 was explored by analyzing two possible types of measures 1) the three further proposed drone cases on the technical side (D2, D3, D4) and 2) additionally, through the future decarbonization scenario of the electricity sector on the technological side.

The analyzed dataset contains a sample of urban home package last-mile deliveries carried out in the U.S. e-commerce sector in five main metropolitan cities. Within this context, the results provide valuable insights regarding RQ2. These insights will be discussed in the form of six key findings. Additionally, general implications and recommendations are mentioned conjointly with each of the six main findings. The main findings read as follows:

1. **The results show that a closer distribution center to the delivery points would be beneficial in terms of a reduction in average flight distances, in the case of widespread urban drone delivery.**

   In order to reduce the average distances, the approach would consist in placing the distribution center in a location closer to the delivery stops. In the case of this study, it was placed in the center of gravity of all stops served from each distribution center, assuming a single delivery per stop coordinate. In this study this was referred to as the alternative distribution center (DC).

   In the context of an urban case, this would likely imply that the distribution center would be located closer to or in the center of a metropolitan area, in contrast to the current DC locations which tend to be located in the peripheries of cities.

   As for the implications of this measure, there are unique challenges when it comes to locating a distribution center closer to an urban area. Given that large unused surfaces tend to usually be scarce and/or expensive in metropolitan areas, there would be an incentive to build warehouses that occupy as less surface area as possible.

On the one hand, a ground vehicle depot would require a large surface intended for parking and warehouse storage for bulky packages, which would not be the most viable option for a central truck depot. On the other hand, the compact drone size and focus on small lightweight packages means that a centrally located DC would be more feasible for drones than for trucks. These drone distribution hubs could be built vertically, in a beehive manner to save space and be built in concurred areas.

Regarding how this concept could be implemented, Amazon has filed a patent for the concept of "multi-level fulfillment centers" (Curlander et al., 2017). This fulfillment center concept consists in a beehive shaped drone port, with multiple levels from which unmanned aerial vehicles can take-off for the purpose of package delivery in a densely populated area (see Figure 19). Even though this is still in the concept phase, this could be a future measure to be implemented in a package delivery system that includes drones.

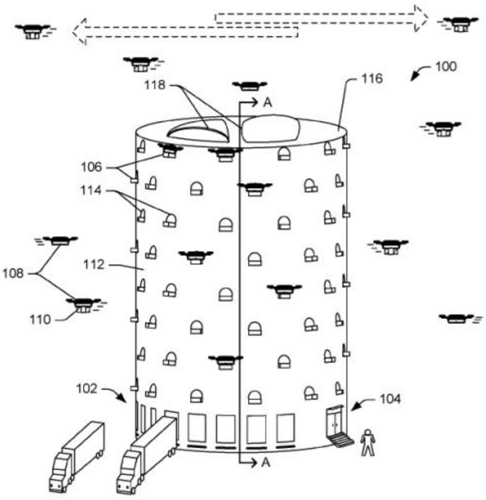


Figure 19. Multi-level Fulfillment Center, sourced from the Patent Application Publication US-20170175413-A1 filed by Amazon (Curlander et al., 2017)

Another implication of urban package delivery in the future is that as there is more congestion and traffic in metropolitan cities, other means of delivery will become more relevant. For truck delivery this could mean that routes could take longer to drive. Here lies a potential advantage of drones over ground delivery, given that drones would not contribute to ground-traffic congestion.

A further implication of widespread drone usage is the increased congestion of air space (Li et al., 2023). For instance, an effort to integrate drones in existing logistic chains while enabling the coexistence of other aircraft traffic, is the UK's "Skyway" project to establish an air corridor for drones. This project could connect cities such as Cambridge, Oxford and Reading with an 265 km corridor which would open other channels for supply chain operations (Tyrrell, 2023).

2. **Multi-delivery of packages is a relevant measure to improve the performance of a drone delivery system.**

   This is mainly because the drone would not need to return to the departing depot every time one package is delivered, but rather only after the multiple loaded packages are delivered. This considerably reduces the total flight distance to be traveled by the drone per each package.

   A limitation for multi-delivery found in the context of the chosen drone specifications, was the large amount of energy consumption needed for an intermediate delivery in the hover mode. This considerably decreases the flight range of the drone.

   Thus, the multi-delivery case could reach its full efficiency potential if the energy density of the drone battery were to increase in the future. In this way, distance flown would not be a significant limiting factor and the efficiency of drone multi-delivery would be further increased by allowing more packages to be delivered within the same flight, leading to less energy and CO_2 emitted during delivery operations.

   However, further research is needed to determine the maximum technical limitations or future developments of drone technology. Regardless of that, considering the limits of current battery-powered drone technology, closer distances from the package delivery stations are needed for multi-delivery to work, which introduces the next key point.

3. **The results suggest that a combination of the two previous measures, a closer distribution center and multi-delivery, would be beneficial for urban drone delivery systems.**

   Even though the combined effect is smaller than the summed individual effects of both measures, having closer distances to the delivery stops allows for a higher grouping of deliveries.

   The less range the drone needs to fly to the cluster of multi-deliveries, the more energy it has available for the energy-intensive intermediate deliveries. As a result, the drone can perform more intermediate deliveries and there is an even further reduction of distance flown per package.

4. **Compared with the current delivery system, drones could perform better than diesel truck delivery fleets in certain cases in terms of delivery carbon emissions.**

   In this kind of urban context, certain drone cases were found to be competitive compared to diesel trucks when comparing the equivalent set of stops to be served. Mainly, drones perform better than a diesel truck when the distances between the distribution center and the delivery stops are short, allowing for a double or triple delivery and lowering the distance flown per package.

   However, it is expected that with increased electrification of delivery truck fleets, drones will not be able to keep pace in the future, which leads to the next finding.

5. **Regarding the future ground-based delivery systems, drones were not found to be competitive with electric vehicle fleets in the simulated cases of this study.**

   Even though drones are relatively small and energy efficient vehicles, there is a big disparity in the distances covered when comparing truck and drone delivery in this urban case. Meaning that the drone distance needed in order to cover the same set of stops is considerably higher than for ground transportation.

   In the context of this dataset, multi-delivery drones would require almost twice the energy consumption on average than the baseline EV case and would fly almost 27 times the ground distance of the EV. Meaning, that even multi-delivery is not enough to make drones contentious with EVs, since the drone remains at a considerable distance disadvantage or distance penalty.

   In terms of a breakeven point in the multi-delivery and alternative DC case, in order to be competitive with electric ground vehicles, drones would on average need to become 43.7% more energy efficient. This energy efficiency needed could be implemented through further routing measures or in technical efficiency measures for the drone.

   As for possible measures to increase the drone energy efficiency, a higher lift-to-drag ratio would be needed. Having a drone with longer and narrower wings leads to a better aspect ratio. A better aspect ratio means more lift, which would lead to less energy needed when in cruise mode. Nonetheless carrying out deliveries with a bigger drone could also lead to difficulties, especially in an urban context where landing space might be limited.

6. **A possible decarbonization of the grid, would give vehicles powered by batteries, a considerable advantage over a diesel vehicle fleet, in this case it would favor the drone and the BEV.**

   As grid decarbonization would gradually exclude diesel vehicle fleets, ergo it is expected that in the future there will be more cases where drones have an advantage over diesel vehicles.

   In this case, the BEVs would still stay the most competitive, nonetheless the difference between the impact caused by the drone and the EV would be of a lesser magnitude. Additionally, the rates of electricity-mix decarbonization need to be considered depending on the geographical context, for instance they might differ between the U.S. and Europe.

   As an implication of minimizing the operational impact of the delivery phase, the non-operational phase is expected to become more significant in terms of environmental impact. Therefore, other life cycle stages would need to be considered, for instance vehicle production, warehouse energy consumption or impact of infrastructure.

In general, it can be concluded that drones possess the advantage of being energy efficient and the main disadvantage of distance disparity and package carrying capacity. Drones were found to be competitive with diesel vehicles in select specific cases with multi-delivery. Nonetheless, electric delivery vehicles proved to be the best-performer in terms of energy consumption and CO_2 emissions.

This clearly implies that the advantage of truck delivery lies in the capacity to deliver a higher amount of packages, thereby reducing the distance driven per package. To contend with this, drones would need to become much more efficient by routing or technical measures, or would be more suitable in single-package and remote delivery cases.

The results suggests that multi-delivery is a crucial routing measure to improve the performance of drone delivery compared to the base case (D1). Triple delivery was observed to be feasible in cases were lower distances needed to be traveled. A significant limitation regarding multi-delivery is the limited drone battery range. In spite of this, technological advancements are expected to allow longer-range operations for drones (Li et al., 2023)

As an energy efficient technology, drones could surely be a constituent of a sustainable last-mile delivery system. However, the delivery system design would need to change to consider the specific cases where a drone could be advantageous. This includes exploring different ways of integrating drones in the delivery system in a more efficient way, such as tandem configurations or other delivery scenarios, which is mentioned as a point for further research in sub-section 4.4.

4.2 Findings in the context of literature

Regarding the relationship of this study with other research, it is important to highlight that not all available papers are comparable with this study. This is due to the multiplicity of approaches in drone last-mile delivery literature, specifically from the sustainability perspective.

The approaches in literature differ, to name a few aspects, with respect to:

- the comparison systems,
- distance calculation methods,
- energy consumption estimation methods,
- considered life cycle phases, and
- modeled scenarios or contexts.

Worth mentioning, is that last-mile delivery and routing is extremely case specific. Depending on whether the case is located in a rather metropolitan or a rural area, and depending on the assumed delivery density, last-mile operations could differ greatly.

Similarly, depending on the modeled situation and comparison point, the result could favor drone or ground transportation more. For instance, if the case is to deliver one package at a higher distance, then it is highly likely that the drone would be more beneficial than driving a single package with a ground vehicle. This could for instance be the case in the context of food delivery.

This "reference scenario" dilemma that complicates comparison between studies is observed in Rodrigues et al.'s (2022) study. The results of this master thesis would not be in line with the findings of that study, given that it suggests that drones could have up to 94% less energy consumption per package than other delivery modes. Such an optimal value for drone delivery is attributed to the fact that the study was carried out under a very specific set of assumptions. For instance, the specific values for stops per km for drones and trucks that were chosen by the authors, which might not correspond to the context of this study focusing on widespread urban operations.

As for more comparable literature, the results of this study align with the general findings of Goodchild & Toy (2018), in the sense that drones tend to perform better when the service zones are closer to the distribution center and that a truck has advantage when the recipients are located far from the depot and there is a high delivery density.

Similarly, the results align with the conclusions of Kirschstein (2020) with regards that electric trucks tend to be the most efficient and that drones could be advantageous when parcel delivery density is not that high. This study is interesting to compare to this Kirschstein's, since it is simulated in the context of a metropolitan area, although in a European city in Berlin.

In the context of grocery delivery, Yowtak et al. (2020) suggested in the same way, that drones could potentially perform better than ICEV systems, however they could not outperform EVs.

When it comes to determining the sustainability of last-mile drone delivery, the comparison systems play a crucial role. The decision of whether last-mile drone delivery is sustainable or not, would therefore benefit from a careful case-by-case assessment and an evaluation of the underlying assumptions.

4.3 Methodological contributions and limitations

4.3.1 Contributions

On the subject of the employed methodology, this study makes three main contributions.

Firstly, this study employs real-world last-mile delivery data for the modelling of the system. As mentioned in section 2.1, there is a limited availability of historical last-mile delivery routing data that is publicly available. For this reason, the majority of previously published works use alternative methods for distance calculations such as continuous approximation or simulation experiments based on chosen assumptions or considering population density.

Depending on the employed method, the assumptions used to determine the drone flight distances in comparison with ground vehicles distances may differ greatly. Therefore, also the results of the studies vary. At the same time, as the carbon emissions from delivery are highly dependent on the distance calculations, this poses the question of to which extent each type of modelling reflects a realistic last-mile package delivery situation.

Consequently, by using a historical last-mile delivery sample dataset this uncertainty is no longer a major concern, considering that the modelling is directly built on a sample of real-world data. To the author's knowledge, this is the first study in this context, that employed real-world last-mile package delivery coordinates to explore the environmental performance of the operation of drone delivery systems.

Secondly, this study based the drone energy consumption calculations on a fixed-wing VTOL drone configuration specifically made for drone delivery, rather than on a commercially available copter drone.

Most of existing works employ a commercially available quadcopter configuration to calculate energy consumption. It is likely that this copter configuration was chosen due to the lack of data availability, concerning drone specifications and drone energy consumption models, at the time when the studies were carried out. In spite of this, the critique that commercially available quadcopter drones might not representative of future delivery systems has also been raised by Zhang et al. (2021).

Drone energy consumption values greatly differ in literature, even within the quadcopter category, for this reason it was important to choose a drone that would represent the operations of the modeled future drone delivery system more accurately.

With this in mind, the drone energy consumption model was specifically tailored for a state-of-the-art drone model, used specifically for package delivery as opposed to a commercial drone for other applications. This comprised a methodical challenge since most of the drone energy consumption models are not developed for this type of configuration.

The fixed-wing VTOL drone configuration was chosen because of the expectation of widespread adoption of this design in future drone delivery systems. Being that it possesses an advantage in terms of energy efficiency, most logistic service providers have shifted to this kind of configuration.

Thirdly, a key strength of this study is the modelling of multi-delivery packages by drone in a realistic setting, which took into consideration the drone battery capacity and range constraints.

For traditional quadcopter drone configurations in existing literature, deliveries were usually modeled for a single package. However, from the technical perspective, multi-delivery of packages is a relatively new capability for drones and reflect the state-of-the-art in last-mile drone delivery (Salama & Srinivas, 2022). In this way, a further measure to increase the efficiency of drone deliveries was able to be explored in this study.

4.3.2 Limitations

While significant contributions were made, it is equally relevant to recognize the limitations and areas of opportunity in this study that could be addressed in future research.

Firstly, it is pertinent to consider the context in which the employed data was collected and therefore for which types of scenarios the results of this study could provide insights. As previously mentioned, the Amazon dataset contained historical routes primarily driven in five big metropolitan cities in the United States. The general findings of this study are therefore meaningful when discussing an urban context in the U.S. and do not necessarily make a statement regarding other scenarios.

For instance, the results in this study are not applicable for scenarios in an increasingly rural or suburban context, in which deliveries tends to be fewer and further away. The proposed routing solutions in this study, such as a closer distribution center, might therefore not have been the most fitting measures in a rural context.

Depending on the premises of the modeled case, which include different assumptions regarding depot proximity and delivery density, the drone’s performance might have been more favorable than in the results of this study without the implementation of additional measures. For instance, in cases when only few packages are delivered by the ground vehicle, drone delivery tends to be more favorable. One example for this might be the case of food delivery, where the ground vehicle would deliver few orders or even a single order.

The sample data employed was collected in the year 2018 and for 18 specific distribution centers. It is possible that the last-mile operations have changed from that point in time, considering that it corresponds to the pre-pandemic period. Additionally, in the case of online delivery there might be effects of seasonality in high peak events, such as leading up to Christmas or Black Friday, which modify the normal package delivery patterns. This was a factor that was not considered in this study. In addition, some of the Amazon distribution centers in the dataset have been ceased operations for reasons of excess warehouse capacity.

The previous points emphasized that last-mile delivery is highly case-specific, hence these results cannot be considered to represent the totality of drone and ground last-mile delivery scenarios. However, this case study type of research can provide general pointers or indicators for this context.

Secondly, this study employed the simplified assumption that a delivery point in the route corresponds to one package delivery. This is an assumption that many other studies have employed as well, however this might not always hold true. Especially in a scenario with a high package delivery density. Nonetheless, this study could be improved by also considering several package deliveries per stop.

Thirdly, it is also important to acknowledge that the results, in addition to the distance calculations, also depend on the assumed energy consumption values. Even though the best-available estimations and vehicles were taken into consideration, there will likely still be a difference compared with real operations.

As for the drone energy consumption model, it is simplified in two main ways. The first improvement that could be implemented is the consideration of separate flight regimes for the initial takeoff and final landing. Given of the attainable specifications of the Wingcopter 198, no separate regimes could be modeled other than for the intermediate delivery and steady flight phases. In order to include this in the energy consumption model, disaggregated data for takeoff and landing would be needed, such as the specific energy consumption or average height and time needed for those two phases.

The second limitation is the simplified calculation of the L/D ratio. Since the chosen energy consumption model was adapted to the fixed-wing VTOL drone, a reverse calculation of the L/D ratio was carried out. In this case, the L/D was dependent on mass, which was assumed to be correlated with projected area. Nonetheless, a more sophisticated way of calculating the L/D ratio, which could probably be directly based on projected area or speed would be an improvement.

The scope of this study was limited to the environmental impact of the delivery or transportation phase in terms of carbon emissions. This means that the study did not consider emissions coming from other life-cycle phases, nor other environmental impact category indicators.

In addition, route optimization was outside of the scope of this thesis, as it constitutes a separate field of study. Meaning, that the prescribed route sequences were utilized.

4.4 Reccomendations for future work

Further work is necessary to explore in which other cases drones could be a part of a sustainable transportation system. In this section three main lines of research that can enrich this field are proposed.

4.4.1 Combinatory drone-truck systems

The results of this study show that delivery done completely by drone could in some specific cases be competitive with diesel truck delivery. However, with an increased ground vehicle fleet electrification in the future, all-drone fleets are not likely to be competitive with electric vehicles. Given that an all or nothing solution such as the one explored in this study might not be the most suitable, it is relevant to explore hybrid truck-drone solutions.

For this reason, future research should definitely address how drones and trucks might be used in combination to achieve a lower overall operational footprint in package delivery. Such combinatory systems are a novel and complex field of research, in which increasingly new variants in routing models are being researched. It would be worthwhile to evaluate these higher-complexity combinatory systems in the context of historical data and in terms of environmental performance.

These could include for instance:

1) Tandem drone-truck delivery configurations, in which a drone collaborates with a truck, which is serving as a recharging station and depot (Salama & Srinivas, 2022). In this case, factors such as truck design or capacity constraints, and the operation of such tandem systems by a driver would be interesting to consider. In addition, aspects such as flexible stop locations for the truck and the comparison of the turnover time of tandem systems would be worthwhile to explore.

2) Cases where a proportion of deliveries made by drone and a proportion by truck, which would involve identifying the cases where drone delivery is the most beneficial in terms of emissions and energy efficiency. For instance, if drones would be used for the further away and single deliveries, while trucks for the multiple clustered high-density deliveries.

3) A case with intermediate urban depots from which drones could deliver from. The packages that the drone then needs to deliver, could be brought by a ground vehicle to the intermediate depot in the context of urban delivery.

Essentially, exploring new frontiers of combinatory systems would be a valuable contribution to this line of research.

4.4.2 Comparison & evaluation of different case studies

Future research might also consider evaluating different home-package delivery scenarios against each other in a comparable manner.

This could on the one hand mean proposing cases with different delivery densities and distances, which could, for instance, entail comparing a rural and an urban scenario. This would probably imply considering factors such as population density and consumer behavior depending on the geographical location.

On the other hand, this could include evaluating other case propositions other than online package delivery, for instance food delivery or time-sensitive deliveries. This would provide further insights to determine whether drones could adopt a widespread or a niche role in home package delivery.

4.4.3 Life cycle impact of delivery fleet

A highly relevant aspect to consider when studying the environmental impact of last-mile drone package delivery, and something that this study explored merely a fraction of, are the emissions throughout the lifecycle of the delivery fleet.

Relatively few studies have been able to consider more aspects of fleet sizing and lifecycle impact of last-mile delivery. Yet, as the non-operational impact of transportation becomes smaller with fleet electrification, other life cycle phases will become more relevant.

Going beyond the operational transportation phase, would imply additionally considering factors such as the extraction of raw materials needed for the vehicles, the vehicle production phase, warehouse operations, end-of-life of the vehicles and warehouse facility demolition.

On the other hand, each vehicle has a different environmental footprint and therefore, fleet sizing needs to be considered. The size of a drone fleet will be different to a ground vehicle fleet, since one ground vehicle would not be substituted by one drone, but probably by more.

In addition to fleet sizing, the different vehicle life cycle lengths need to be regarded. For instance, it is plausible that the production impact of a drone is less than that for a ground vehicle. However, it could be that more drones are needed than ground vehicles in the fleet.

Exploring the possible impacts and trade-offs of drone vs. truck delivery systems, considering more aspects of the lifecycle & fleet sizing, would allow for a more comprehensive analysis of drone last-mile delivery.

As recently identified by Mitchell et al. (2023), there is a lack of robust data for drone part production and end-of-life, nonetheless in order to holistically evaluate the sustainability of drone delivery services the complete product’s life cycle needs to be considered.

5 Conclusion

The main aim of this master thesis was to quantify the environmental impacts of the operational delivery phase of a widespread drone-based home delivery system, by comparing it to a ground-based delivery system.

The analysis was carried out using available real-world U.S. package delivery data in an urban last-mile logistic context. At the same time, the implementation of the model in Python was a major component of this master's thesis.

For this study, baseline-cases of the delivery systems were modeled based on a diesel delivery truck, an electric delivery truck and a fixed-wing VTOL drone (G1, G2, D1). The base-case included the original distribution centers and delivery points included in the dataset. In addition, three drone delivery cases including different technical routing measures (D2, D3, D4) and one decarbonization scenario were evaluated. The operational environmental performance was analyzed in terms of distance, carbon emissions and energy consumption.

For this urban context of widespread home delivery in last-mile operations in the U.S., the main findings of this case study can be summarized with the following points:

1. The results of this study suggest that a closer distribution center (D2) would be beneficial for drone delivery in an urban context, resulting in an overall reduction in distances to be flown.
2. Multi-delivery of packages (D3) was found to be a highly relevant measure to improve the performance of a drone delivery system. However, a significant limitation regarding multi-delivery is the limited battery range of drones.
3. A combination of both measures (D4), a closer distribution center and multi-delivery, would be beneficial, given the additional energy available for the energy-intensive intermediate deliveries through the reduction of flown distances.
4. Compared with the current delivery system, drones could perform better than diesel truck delivery fleets in certain cases, in terms of carbon emissions during delivery. These cases included the usage of drone package multi-delivery.
5. Regarding the future ground-based delivery systems, drones were not found to be competitive with electric vehicle fleets in the simulated cases of this study. This is mainly due to the significant distance disparity in the case of drone delivery.
6. Transitioning to a less-carbon intensive electricity mix, would considerably favor vehicles powered by batteries, EVs and drones, over diesel fleets. This would however draw the focus to other stages of the delivery life cycle.

The results suggests that the main advantage of ground vehicle delivery lies in the ability to deliver more of packages, hence reducing the distance traveled per package. For this reason, drones would need to become significantly more efficient by routing or technical means or would be more appropriate in single-package and distant delivery.

Regarding the methodology, this study contributes in three main aspects: 1) this is one of the pioneering works employing real-world data to evaluate the environmental impacts of last-mile drone and ground-based delivery systems, 2) a more efficient and state-of-the-art fixed-wing VTOL drone configuration was employed in the modelling of the system to reflect the operations of a future drone delivery fleet and, lastly, 3) the multi-delivery of up to three packages with an UAV was modeled considering realistic constraints for range and maximum battery capacity, which represents a shift from the existing modeling considering solely single deliveries.

While this study provided relevant insights into the widespread use of drone delivery systems, the findings of this study are subject to some limitations. These limitations include the specific context of the evaluated case study, the scope limited to the operational phase and the assumptions employed in the modelling of the delivery systems.

Lastly, as lightweight and energy efficient vehicles, drones could be a valuable component of a sustainable last-mile delivery system in the future. Nonetheless, to determine an optimal and more efficient role for drones in existing logistic systems, the potential of combinatory systems should be further researched. In addition, with a sinking operational footprint, a focus on other life cycle phases should be increasingly considered.

References

Amazon. (2021). *2021 Amazon Last Mile Routing Research Challenge Dataset*. https://registry.opendata.aws/amazon-last-mile-challenges

Amazon. (2022). *Amazon reveals the new design for Prime Air’s delivery drone*. US About Amazon; US About Amazon. https://www.aboutamazon.com/news/transportation/amazon-prime-air-delivery-drone-reveal-photos

Amazon Last-Mile Routing Research Challenge. (2021). *About the Challenge | Amazon Last-Mile Routing Research Challenge*. MIT. https://routingchallenge.mit.edu/about-the-challenge/

ANL. (n.d.). *GREET - Well to Wheels*. Greet.Es.Anl.Gov. Retrieved January 1, 2022, from https://greet.es.anl.gov/greet/gettingstarted/wtw.html

ANL. (2021). *GREET - WTW Calculator 2021*. Argonne National Laboratory. https://greet.es.anl.gov/results

Awwad, M., Shekhar, A., & Sundaranarayanan Iyer, A. (2018, September 27). *Sustainable Last-Mile Logistics Operation in the Era of E- Commerce*. Proceedings of the International Conference on Industrial Engineering and Operations Management, Washington DC, USA.

Bányai, T. (2022). Impact of the Integration of First-Mile and Last-Mile Drone-Based Operations from Trucks on Energy Efficiency and the Environment. *Drones*, *6*(9), 249. https://doi.org/10.3390/drones6090249

Borghetti, F., Caballini, C., Carboni, A., Grossato, G., Maja, R., & Barabino, B. (2022). The Use of Drones for Last-Mile Delivery: A Numerical Case Study in Milan, Italy. *Sustainability*, *14*(3), 1766. https://doi.org/10.3390/su14031766

Boudette, N. (2023). Electric Vans, Delayed by Production Problems, Find Eager Buyers. *The New York Times*. https://www.nytimes.com/2023/05/16/business/energy-environment/electric-vehicle-delivery-vans.html

Briest, P., Dragendorf, J., Ecker, T., Mohr, D., & Neuhaus, F. (2019). *The endgame for postal networks: How to win in the age of e-commerce*. McKinsey & Company. https://www.mckinsey.com/~/media/mckinsey/industries/travel%20logistics%20and%20infrastructure/our%20insights/the%20endgame%20for%20postal%20networks%20how%20to%20win%20in%20the%20age%20of%20e%20commerce/the_endgame_for_postal_networks_how_to_win_in_the_age_of_e-commerce.pdf

Capgemini Research Institute. (2019). *The last-mile delivery challenge*. https://www.capgemini.com/wp-content/uploads/2019/01/Report-Digital-%E2%80%93-Last-Mile-Delivery-Challenge1.pdf

Carter, S., Stephan, L., Robin, R., Leonard, T., & Tore, J. (2022). *Drone delivery: More lift than you think*. McKinsey & Company. https://www.mckinsey.com/industries/aerospace-and-defense/our-insights/future-air-mobility-blog/drone-delivery-more-lift-than-you-think

Chauhan, S., Hans, M., Moritz Rittstieg, & Zafar, S. (2022a). *Fleet decarbonization: Operationalizing the transition*. McKinsey & Company; McKinsey & Company. https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/fleet-decarbonization-operationalizing-the-transition

Chauhan, S., Hans, M., Moritz Rittstieg, & Zafar, S. (2022b). *Getting to carbon-free commercial fleets*. McKinsey & Company; McKinsey & Company. https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/getting-to-carbon-free-commercial-fleets

Chiang, W.-C., Li, Y., Shang, J., & Urban, T. L. (2019). Impact of drone delivery on sustainability and cost: Realizing the UAV potential through vehicle routing optimization. *Applied Energy*, *242*, 1164–1175. https://doi.org/10.1016/j.apenergy.2019.03.117

Curlander, J. C., Gilboa-Amir, A., Kisser, L. M., Koch, R. A., & Welsh, R. D. (2017). *MULTI-LEVEL FULFILLMENT CENTER FOR UNMANNED AERIAL VEHICLES- US-20170175413-A1*. https://image-ppubs.uspto.gov/dirsearch-public/print/downloadPdf/20170175413

D’Andrea, R. (2014). Guest Editorial Can Drones Deliver? *IEEE Transactions on Automation Science and Engineering*, *11*(3), 647–648. https://doi.org/10.1109/TASE.2014.2326952

Data for Good at Meta, & Columbia University. (2020). *High Resolution Settlement Layer (United States: High Resolution Population Density Maps + Demographic Estimates)* [Data set]. https://data.humdata.org/dataset/united-states-high-resolution-population-density-maps-demographic-estimates#

DHL. (2021). *DHL Express Partners with Fiat Professional for Further Electrification of Last-mile Delivery*. DHL. https://www.dhl.com/global-en/home/press/press-archive/2021/dhl-express-partners-with-fiat-professional-for-further-electrification-of-last-mile-delivery.html

D’Onfro, J. (2019). *Amazon’s New Delivery Drone Will Start Shipping Packages “In A Matter Of Months.”* Forbes. https://www.forbes.com/sites/jilliandonfro/2019/06/05/amazon-new-delivery-drone-remars-warehouse-robots-alexa-prediction/

EPA. (2023). *Greenhouse Gases Equivalencies Calculator—Calculations and References | US EPA*. US EPA. https://www.epa.gov/energy/greenhouse-gases-equivalencies-calculator-calculations-and-references

Eskandaripour, H., & Boldsaikhan, E. (2023). Last-Mile Drone Delivery: Past, Present, and Future. *Drones*, *7*(2), 77. https://doi.org/10.3390/drones7020077

European Commission JEC. (2016). *Well-to-Wheels Analyses*. Joint-Research-Centre.Ec.Europa.Eu; European Commission Joint Research Centre. https://joint-research-centre.ec.europa.eu/welcome-jec-website/jec-activities/well-wheels-analyses_en

European Environment Agency. (2020). *The first and last mile: The key to sustainable urban transport : transport and environment report 2019.* Publications Office. https://data.europa.eu/doi/10.2800/200903

FAA. (2022). *Package Delivery by Drone (Part 135) | Federal Aviation Administration*. https://www.faa.gov/uas/advanced_operations/package_delivery_drone

Fiat. (2020). *E-Ducato Configurator| Electric Van | Fiat Professional UK*. https://www.fiatprofessional.com/uk/e-ducato/configurator/#/version

Figliozzi, M. A. (2017). *Lifecycle modeling and assessment of unmanned aerial vehicles (Drones) CO2e emissions*. 11.

Figliozzi, M. A. (2020). Carbon emissions reductions in last mile and grocery deliveries utilizing air and ground autonomous vehicles. *Transportation Research Part D: Transport and Environment*, *85*, 102443. https://doi.org/10.1016/j.trd.2020.102443

Fontaras, G., Ciuffo, B., Zacharof, N., Tsiakmakis, S., Marotta, A., Pavlovic, J., & Anagnostopoulos, K. (2017). The difference between reported and real-world CO 2 emissions: How much improvement can be expected by WLTP introduction? *Transportation Research Procedia*, *25*, 3933–3943. https://doi.org/10.1016/j.trpro.2017.05.333

Frota Neto, J. Q., Bloemhof-Ruwaard, J. M., van Nunen, J. A. E. E., & van Heck, E. (2008). Designing and evaluating sustainable logistics networks. *International Journal of Production Economics*, *111*(2), 195–208. https://doi.org/10.1016/j.ijpe.2006.10.014

Garland, M. (2023). *Walmart made over 6,000 drone deliveries in 2022*. Retail Dive. https://www.retaildive.com/news/walmart-6000-drone-deliveries-droneup-flytrex-zipline-2022/639837/

GeoPy Contributors. (n.d.). *GeoPy's documentation: Calculating Distances*. Retrieved January 1, 2022, from https://geopy.readthedocs.io/en/stable/#module-geopy.distance

Goodchild, A., & Toy, J. (2018). Delivery by drone: An evaluation of unmanned aerial vehicle technology in reducing CO 2 emissions in the delivery service industry. *Transportation Research Part D: Transport and Environment*, *61*, 58–67. https://doi.org/10.1016/j.trd.2017.02.017

Grant, D. B., Trautrims, A., & Wong, C. Y. (2015). *Sustainable logistics and supply chain management* (Revised edition). Kogan Page.

Heid, B., Hensley, R., Knupfer, S., & Tschiesner, A. (2017). *What's sparking electric-vehicle adoption in the truck industry?* McKinsey & Company; McKinsey & Company. https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/whats-sparking-electric-vehicle-adoption-in-the-truck-industry

IBM. (n.d.). *What are virtual machines?* IBM. Retrieved January 1, 2023, from https://www.ibm.com/topics/virtual-machines

Industrial Ecology Digital Lab. (2023). *cloudio—NTNU Wiki*. Ntnu.No. https://www.ntnu.no/wiki/pages/viewpage.action?spaceKey=iedl&title=cloudio

IPCC. (2022). *Summary for Policymakers. In: Climate Change 2022: Mitigation of Climate Change. Contribution of Working Group III to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*. Intergovernmental Panel on Climate Change. https://www.ipcc.ch/report/ar6/wg3/downloads/report/IPCC_AR6_WGIII_SPM.pdf

Jang, J. J., & Song, H. H. (2015). Well-to-wheel analysis on greenhouse gas emission and energy use with petroleum-based fuels in Korea: Gasoline and diesel. *The International Journal of Life Cycle Assessment*, *20*(8), 1102–1116. https://doi.org/10.1007/s11367-015-0911-x

JOUAV. (n.d.). *CW-15 Multi-purpose and long endurance VTOL drone*. JOUAV. Retrieved January 1, 2022, from https://www.jouav.com/products/cw-15.html

JOUAV. (2022). *Different Types of Drones and Uses*. JOUAV. https://www.jouav.com/blog/drone-types.html

Kirschstein, T. (2020). Comparison of energy demands of drone-based and ground-based parcel delivery services. *Transportation Research Part D: Transport and Environment*, *78*, 102209. https://doi.org/10.1016/j.trd.2019.102209

Kirschstein, T. (2021). Energy demand of parcel delivery services with a mixed fleet of electric vehicles. *Cleaner Engineering and Technology*, *5*, 100322. https://doi.org/10.1016/j.clet.2021.100322

Koiwanit, J. (2018). Analysis of environmental impacts of drone delivery on an online shopping system. *Advances in Climate Change Research*, *9*(3), 201–207. https://doi.org/10.1016/j.accre.2018.09.001

Laseinde, O. T., & Mpofu, K. (2017). Providing solution to last mile challenges in postal operations. *International Journal of Logistics Research and Applications*, *20*(5), 475–490. https://doi.org/10.1080/13675567.2017.1288712

Lenihan, R. (2022). *Amazon Finally Makes Its Drone-Delivery Dream Come True*. TheStreet; TheStreet. https://www.thestreet.com/retailers/amazon-finally-makes-its-drone-delivery-dream-come-true

Li, X., Tupayachi, J., Sharmin, A., & Martinez Ferguson, M. (2023). Drone-Aided Delivery Methods, Challenge, and the Future: A Methodological Review. *Drones*, *7*(3), 191. https://doi.org/10.3390/drones7030191

Macioszek, E. (2018). First and Last Mile Delivery – Problems and Issues. In G. Sierpiński (Ed.), *Advanced Solutions of Transport Systems for Growing Mobility* (pp. 147–154). Springer International Publishing.

Martinez-Val, R., Perez, E., & Palacin, J. (2005, January 10). Historical Perspective of Air Transport Productivity and Efficiency. *43rd AIAA Aerospace Sciences Meeting and Exhibit*. 43rd AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada. https://doi.org/10.2514/6.2005-121

McKinsey Energy Insights. (n.d.). *Lower heating value*. McKinsey Energy Insights Resources. Retrieved January 1, 2022, from https://www.mckinseyenergyinsights.com/resources/refinery-reference-desk/lower-heating-value/

Merchán, D., Arora, J., Pachon, J., Konduri, K., Winkenbach, M., Parks, S., & Noszek, J. (2022). 2021 Amazon Last Mile Routing Research Challenge: Data Set. *Transportation Science*. https://doi.org/10.1287/trsc.2022.1173

Mitchell, S., Steinbach, J., Flanagan, T., Ghabezi, P., Harrison, N., O'Reilly, S., Killian, S., & Finnegan, W. (2023). Evaluating the sustainability of lightweight drones for delivery: Towards a suitable methodology for assessment. *Functional Composite Materials*, *4*(1), 4. https://doi.org/10.1186/s42252-023-00040-4

Open Data Network. (2018). *Population Density Data for Los Angeles, CA, Los Angeles County, CA, Chicago, IL and New York, NY - Population on the Open Data Network*. https://www.opendatanetwork.com/entity/1600000US0644000-0500000US06037-1600000US1714000-1600000US3651000/Los_Angeles_CA-Los_Angeles_County_CA-Chicago_IL-New_York_NY/geographic.population.density?year=2018&ref=compare-entity

Oršič, J., Jereb, B., & Obrecht, M. (2022). Sustainable Operations of Last Mile Logistics Based on Machine Learning Processes. *Processes*, *10*(12), 2524. https://doi.org/10.3390/pr10122524

Parks, S., Schroeder, J., Van Riper, David, Kugler, T., & Ruggles, S. (2022). *National Historical Geographic Information System: Version 17.0* (17.0) [Data set]. Minneapolis, MN: IPUMS. https://doi.org/10.18128/D050.V17.0

Patella, S. M., Grazieschi, G., Gatta, V., Marcucci, E., & Carrese, S. (2020). The Adoption of Green Vehicles in Last Mile Logistics: A Systematic Review. *Sustainability*, *13*(1), 6. https://doi.org/10.3390/su13010006

Pitney Bowes. (2022). *Parcel shipping index 2022*. Pitney Bowes. https://www.pitneybowes.com/content/dam/pitneybowes/us/en/shipping-index/22-pbcs-04529-2021-global-parcel-shipping-index-ebook-web-002.pdf

Prieto Camarillo, K. L. (2022). *Sustainability of widespread drone use in last-mile logistic delivery* [Research Project]. Norwegian University of Science and Technology.

Rodrigues, T. A., Patrikar, J., Oliveira, N. L., Matthews, H. S., Scherer, S., & Samaras, C. (2022). Drone flight data reveal energy and greenhouse gas emissions savings for very small package delivery. *Patterns*, *3*(8), 100569. https://doi.org/10.1016/j.patter.2022.100569

Salama, M. R., & Srinivas, S. (2022). Collaborative truck multi-drone routing and scheduling problem: Package delivery with flexible launch and recovery sites.

Transportation Research Part E: Logistics and Transportation Review, *164*, 102788. https://doi.org/10.1016/j.tre.2022.102788

Statista. (2022). *Global retail e-commerce sales 2026*. Statista. https://www.statista.com/statistics/379046/worldwide-retail-e-commerce-sales/

Stolaroff, J. K., Samaras, C., O'Neill, E. R., Lubers, A., Mitchell, A. S., & Ceperley, D. (2018). Energy use and life cycle greenhouse gas emissions of drones for commercial package delivery. *Nature Communications*, *9*(1), 409. https://doi.org/10.1038/s41467-017-02411-5

Tyrrell, J. (2023). *Supply chain innovation: Drone corridor unlocks pilotless future*. TechHQ; TechHQ. https://techhq.com/2023/01/supply-chain-innovation-drone-corridor-unlocks-pilotless-future/

U.S. Census Bureau. (2022). *Population & Income Estimates United States*. Census Bureau; United States Census Bureau. https://www.census.gov/quickfacts/fact/table/US/POP010220#POP010220

U.S. Census Bureau. (2023). *Urban and Rural*. Census Bureau. https://www.census.gov/programs-surveys/geography/guidance/geo-areas/urban-rural.html

U.S. Department of Energy. (n.d.). *Alternative Fuels Data Center: Fuel Properties Comparison*. Retrieved October 19, 2022, from https://afdc.energy.gov/fuels/properties

Wasser, L., Holdgraf, C., & Morrisey, M. (2018). *Earth Analytics: Geotiff (.tif) Raster File Format: Raster Data in Python*. Earth Data Science - Earth Lab. https://www.earthdatascience.org/courses/use-data-open-source-python/intro-raster-data-python/fundamentals-raster-data/intro-to-the-geotiff-file-format/

Willems, K. (2020). *Python Functions Tutorial*. DataCamp. https://www.datacamp.com/tutorial/functions-python-tutorial

Wingcopter. (n.d.). *Media Assets – Wingcopter*. Wingcopter. Retrieved January 1, 2023, from https://wingcopter.com/media-assets

Wingcopter. (2021a). *Wingcopter 198 Delivery Variant: Technical Details*. https://wingcopter.com/wp-content/uploads/2022/06/Specsheet-Wingcopter-198.pdf

Wingcopter. (2021b). *Partnership Air Methods – Wingcopter*. Wingcopter. https://wingcopter.com/air-methods

Wooley, D., Phadke, A., & O'Conell, R. (2020). *2035 Report: Plummeting Solar, Wind, and Battery Costs Can Accelerate Our Clean Energy Future*. UC Berkeley's Center for Environmental Public Policy. http://www.2035report.com/wp-content/uploads/2020/06/2035-Report.pdf?hsCtaTracking=8a85e9ea-4ed3-4ec0-b4c6-906934306ddb%7Cc68c2ac2-1db0-4d1c-82a1-65ef4daaf6c1

World Bank. (2023). *Land area (sq. Km) United States- FAO*. World Bank Open Data. https://data.worldbank.org/indicator/AG.LND.TOTL.K2?locations=US

WorldPop, & CIESIN Columbia University. (2020). *The spatial distribution of population density in 2020, United States (Global High Resolution Population Denominators Project)* [Data set]. https://dx.doi.org/10.5258/SOTON/WP00674

Yang, L., Cai, H., & Szeto, W. Y. (2023). Environmental implications of emerging transportation technologies. *Transportation Research Part D: Transport and Environment*, *116*, 103655. https://doi.org/10.1016/j.trd.2023.103655

Yang, Z. (2023). *Food delivery by drone is just part of daily life in Shenzhen*. MIT Technology Review; MIT Technology Review. https://www.technologyreview.com/2023/05/23/1073500/drone-food-delivery-shenzhen-meituan/

Yowtak, K., Imiola, J., Andrews, M., Cardillo, K., & Skerlos, S. (2020). Comparative life cycle assessment of unmanned aerial vehicles, internal combustion engine vehicles and battery electric vehicles for grocery delivery. *Procedia CIRP*, *90*, 244–250. https://doi.org/10.1016/j.procir.2020.02.003

Zhang, J. (2021). *Economic and Environmental Impacts of Drone Delivery*. https://irl.umsl.edu/dissertation/1052

Zhang, J., Campbell, J. F., Sweeney II, D. C., & Hupman, A. C. (2021). Energy consumption models for delivery drones: A comparison and assessment. *Transportation Research Part D: Transport and Environment*, *90*, 102668. https://doi.org/10.1016/j.trd.2020.102668

Appendix A

Appendix A.1. DC codes with geographical coordinates for the alternative and the original cases

DC code	City	Original DC		Alternative DC	
		latitude	longitude	latitude	longitude
DAU1	Austin	30.445236	-97.709418	30.320506	-97.798549
DBO1	Boston	42.394375	-71.055555	42.4393756	-71.09921
DBO2	Boston	42.233353	-71.141617	42.2611008	-71.086707
DBO3	Boston	42.139891	-71.494346	42.2120441	-71.409523
DBO6	Boston	42.791897	-71.529552	42.8693812	-71.512646
DCH1	Chicago	41.840375	-87.683736	41.8532971	-87.748323
DCH2	Chicago	42.031368	-87.776596	41.980439	-87.788413
DCH3	Chicago	41.803295	-88.097259	41.8099891	-88.152812
DCH4	Chicago	42.254346	-87.985697	42.1858242	-88.051792
DLA3	Los Angeles	34.007369	-118.14393	34.0477919	-118.26369
DLA4	Los Angeles	34.23485	-118.58421	34.208539	-118.49661
DLA5	Los Angeles	33.937865	-117.29738	33.9438584	-117.22814
DLA7	Los Angeles	33.965477	-117.6533	33.9822395	-117.71938
DLA8	Los Angeles	33.918699	-118.32484	33.9208332	-118.32953
DLA9	Los Angeles	33.688122	-117.84718	33.6675966	-117.82574
DSE2	Seattle	47.54217	-122.32805	47.6114342	-122.31832
DSE4	Seattle	47.937344	-122.24495	47.7676674	-122.24296
DSE5	Seattle	47.464945	-122.23107	47.3197238	-122.28783

Appendix A.2. Detailed breakdown of OSMR route exception handling

Parts of Dataset	A (3,000)	B (6,000)	Total
Routes	3,052	6,112	9,164
Stops	433,231	904,527	1,337,758
Calculated routes	3,051	6,106	9,157
Non-calculated routes	1	6	7
Non-calculated stops	121	765	886
Calculated stops	433,110	903,762	1,336,872

Appendix B

Below are the most important code segments for the carried-out research referred to in section 2.4. These codes exclude intermediate processing steps such as cleaning, joining or merging.

Appendix B.1. Data import and distance calculation

```
# %%
""" DESCRIPTION:

  - this code calculates the distances for drone and truck for a given list of DCs
      modify => DC_trial_list=["DSE4","DLA8"]

  - the bottleneck is the API OSRM, there are two options:
      a) getting data from API (depends on the traffic, should not be of excessive use)
      b) getting data from locally hosted server

  - OUTPUT OF THIS CODE are 2 excel files:
      1) csv with drone distances per stop, stop coordinates
      2) csv with truck + drone distances per route ID

  - This uses the conda environment: GISenv
"""

#%%
# ------------------------------ import packages ----------------------------- #
# * import packages for data extraction
import pandas as pd
import numpy as np
import json
import requests # import requests for the API's get request
import pathlib
import os
from IPython.display import display
from pyproj import Geod #for drone distances

# ------------------------------define paths-------------------------------------------- #

# * define project path
project_folder_path = pathlib.Path("../../../")
```

```python
# goes to the folder where the Amazon dataset is contained
dataset_folder_path = "…"
interim_data_folder_path = "…"

# navigate inside the data folders
# NOTE: these folder paths do not change
    # these two paths correspond to the locations corresponding to the files corresponding to the 3,000 or the 6,000 routes
routes_path_3000 = str(dataset_folder_path) + "/data/almrrc2021-data-evaluation/model_apply_inputs/eval_route_data.json"
sequence_path_3000 = str(dataset_folder_path) + "/data/almrrc2021-data-evaluation/model_score_inputs/eval_actual_sequences.json"

routes_path_6000 = str(dataset_folder_path) + "/data/almrrc2021-data-training/model_build_inputs/route_data.json"
sequence_path_6000 = str(dataset_folder_path) + "/data/almrrc2021-data-training/model_build_inputs/actual_sequences.json"

dc_coord_path = str(interim_data_folder_path) + "/DC COORDINATES/dc_coordinates.csv"


# -------------------------------- import DC coordinates  ------------------------------- #

dc_coord = pd.read_csv(dc_coord_path)
display(dc_coord)

# -------------------------------- JSON import ------------------------------- #
# * import the json into a pandas dataframe or table

# * 1) ROUTES DATASET (r)
    # with pandas, get a table
    # columns contain the individual route IDs
    # the stops row contains dictionary with all the stops & their data
dfr = pd.read_json(routes_path_6000)

# drop the excess rows such as
    # departure time, executor capacity, ...
dfr.drop(["date_YYYY_MM_DD", "departure_time_utc", "executor_capacity_cm3"], axis=0, inplace=True)
display(dfr.head())

# dataframe transposed for a column overview
dfr_Vertical = dfr.T
display(dfr_Vertical.head())

# drop the extra columns to just keep route ID + DC code
dfr_DC= dfr_Vertical["station_code"].to_frame()
dfr_DC.reset_index(inplace=True)
dfr_DC.columns= ["route ID", "DC code"]
display(dfr_DC)

# * 2) ACTUAL SEQUENCES DATASET (s)
# dfs: dataframe cointaining sequences
```

```
dfs = pd.read_json(sequence_path_6000).T
display(dfs.head())

# %%
# --------------------- get a list of the unique stations -------------------- #
stationlist = sorted(dfr.loc["station_code"].unique())
print(stationlist)

# %% # create a list of the DCs, to explore now
# -------------------------------------------------------------------------- #
#                         list of DCs to explore
# -------------------------------------------------------------------------- #

# NOTE: later on replace by the list of all DCs or a certain part of the DCs

DC_trial_list= ['DSE4', 'DSE5']

# TODO: Run each DC cluster
# ['DAU1','DBO1', 'DBO2', 'DBO3']
# ['DLA3', 'DLA4', 'DLA5', 'DLA7']
# ['DLA8', 'DLA9','DCH1', 'DCH2']
# ['DCH3', 'DCH4','DSE2']
# ['DSE4', 'DSE5']

# %%
# -------------------- create lists to store the loop dfs -------------------- #

# created two lists where the dfrs are gonna be saved to
dfr1_drone_stops_coordinates_distances_list= []
dfr2_both_routeID_distances_list= []

# %%
# ------------------- FUNCTION TO CALCULATE DRONE DISTANCES ------------------ #

# defined a function for the drone distances
def drone_dist_calculator(lon1,lat1,coordinate_table) ->list:
    """ Args:
        lon1 (_type_): DC/ departure coordinate longitude
        lat1 (_type_): DC/ departure coordinate latitude
            a) ORIGINAL:
                lon1_origDC
                lat1_origDC
            b) ALTERNATIVE:
                avg_lat = lat1_altDC
                avg_long = lon1_altDC

        coordinate_table (_type_): where to fetch the coordinates from
            a) ORIGINAL:  dfr_routesDC_sorted
```

```
        b) ALTERNATIVE: dfr_routesDC_sorted_modDC
    """
    drone_dist_list=[]
    lat_lon_list=[]

    # Calculate drone distances
    for idx in routeIDlist :
        sorted_routeID_coords = coordinate_table[coordinate_table["route ID"]== idx]
        sorted_routeID_coords.sort_values(by="sequence", inplace=True)
        for _ ,waypoint in sorted_routeID_coords.iterrows():
            lon2 = waypoint["longitude"]
            lat2 = waypoint["latitude"]
            lat_lon_list.append([lon2,lat2])
            # function to calculate the distances
            angle1, angle2, dist1 = wgs84_geod.inv(lon1, lat1, lon2, lat2)
            #append in a list of drone distances
            drone_dist_list.append(dist1)

    return drone_dist_list
    print(lat_lon_list)

# -------------------------------------------------------------------------- #
#                          LOOP for each DC                                  #
# -------------------------------------------------------------------------- #
for dcx in DC_trial_list:
    # * Get the routeID list for the currect dcx
    subset2DC= dfr_DC[dfr_DC["DC code"].isin([str(dcx)])]
    # pass the route IDs to a list, that will be used by later for loops
    routeIDlist = subset2DC["route ID"].to_list()
    routeIDlist.sort()

# ------------- 1) Extract lat/longs of all stops of current dcx ------------- #
    all_listlatlong= []
    # for each route ID in the list "..."
    for idx in routeIDlist :
        # for each stop in the column of that route ID
        for stopids in dfr.loc["stops",idx].keys() :
            td = dfr.loc["stops",idx][stopids]
            all_listlatlong.append({"route ID":idx,"stop ID": stopids, "latitude": td["lat"], "longitude": td["lng"]})

    dfr_routesDC_unsorted_noDC = pd.DataFrame.from_dict(all_listlatlong)
    dfr_routesDC_unsorted = dfr_routesDC_unsorted_noDC.merge(dfr_DC, on= "route ID", how="left")
    # set multilevel index for better visualization, first by routeID, then by stop no.
    dfr__routesDC_unsorted_mindex =dfr_routesDC_unsorted.set_index(["DC code","route ID","stop ID"])
# ---------------------- 2) extract the sequences (for) ---------------------- #
    all_listsequence=[]

    # for every route ID in the selected list
```

```
for idx in routeIDlist :
    # go into each of the keys for that specific route ID
    for stopids2 in dfs.loc[idx, "actual"].keys() :
        # go into the content of the key, in this case the observed sequence
        #dfs: dataframe where the sequences were initially extracted to
        td2 = dfs.loc[idx, "actual"][stopids2]
        all_listsequence.append({ "route ID": idx, "stop ID": stopids2, "sequence": td2})

# Result: Create table with actual sequences for each route ID from the DC
dfr_routesDC_sequences = pd.DataFrame.from_dict(all_listsequence)
# multilevel index: Route ID -> then stop ID
dfr_routesDC_sequences_mindex = dfr_routesDC_sequences.set_index(["route ID", "stop ID"])

# ---------------- 3) sort route IDs (for that DC) by sequence --------------- #
## * OUTPUT: merge the two tables (sequence + stops lat/longs)
dfr_routesDC_merged = dfr_routesDC_sequences.merge(dfr_routesDC_unsorted, on=["route ID","stop ID"], how="outer")

## * OUTPUT: SORT table by sequence
dfr_routesDC_sorted = dfr_routesDC_merged.sort_values(by=["route ID","sequence"])
dfr_routesDC_sorted = dfr_routesDC_sorted.reset_index(drop=True)
# multilevel index for better visualization
dfr_routesDC_sorted_mindex = dfr_routesDC_sorted.set_index(["route ID","stop ID"])

# -------------------- 4) distances ground vehicles (OSMR) ------------------- #
""" USE OSRM:
Open Sourced Routing Machine
    - it's a http API
    - f string to fill in
    - the result of the get request is in a json file

* API format: lon, lat
"""
## * extract the DC coordinate
subset1 = dc_coord[dc_coord["DC code"]== str(dcx)].copy()
# need to reset index, so that index =0 always works
subset1.reset_index(inplace= True)
display(subset1)
DClon_exact = lon_exact = subset1.longitude[0]
print(DClon_exact)
DClat_exact= lat_exact = subset1.latitude[0]
print(DClat_exact)

## * loop to create API request
# first part of the http request => add the first coordinate of the warehouse
# create empty distance list
url_list=[]
distance_list= []
exception_counter = 0
```

```
    # idx refers to each route ID in our route list
    for idx in routeIDlist :
        #SERVER OF LOCAL MACHINE:
        #urlrequest= (f"http://187.192.4.216:8080/route/v1/car/{DClon_exact},{DClat_exact}")

        #API:
        #urlrequest= (f"http://router.project-osrm.org/route/v1/car/{DClon_exact},{DClat_exact}")
        coords_sorted_per_routeID = dfr_routesDC_sorted[dfr_routesDC_sorted["route ID"] == str(idx)]
        coords_sorted_per_routeID.sort_values(by="sequence", inplace=True)
        urlrequest= (f"http://router.project-osrm.org/route/v1/car/{DClon_exact},{DClat_exact}")
        # "_" refers to each rows index, its another variable => this is how iterrows works
        for _, waypoint in coords_sorted_per_routeID.iterrows():
            #for _, waypoint in dfr_routesDC_sorted[dfr_routesDC_sorted["route ID"] == str(idx)].iterrows():
            urlrequest+=(f';{waypoint["longitude"]},{waypoint["latitude"]}')

        # add the DC return once more at the end!
        urlrequest+= f";{DClon_exact},{DClat_exact}"""
        urlrequest+="?overview=false"""

        # save url request result
        apirequest = requests.get(urlrequest)
        # convert to dict from json
        json_request = json.loads(apirequest.content)
        # append to the list, with exception handling
        url_list.append(str(urlrequest)) # TODO how tosave the urls (per DC?)

        try:
            distance_list.append(json_request["routes"][0]["distance"])
        except:
            distance_list.append(0)
            exception_counter+= 1

        # distance is given in meters
    print(str(dcx)+"exception counter:"+str(exception_counter))
    #list of distances for each route from the DC
    print(distance_list)
    len(distance_list)

# ---------------------------- 5) drone distances ---------------------------- #
    """
    FOR DRONE DISTANCES:  pyproj package
        from pyproj import Geod

    geod = Geod(ellps="WGS84")
    angle1, angle2, dist1 = geod.inv(lon1, lat1, lon2, lat2, *args, **kwargs)

    # this returns distance in m
```

```
"""

#  Have two alternatives to calculate drone distances:
# a) original DC
# b) average of the points (similar to center of gravity assuming 1 delivery per stop)

# get the mean of the column
mean_coord= dfr_routesDC_sorted_mindex[["latitude","longitude"]].mean()
# save the means
avg_lat = mean_coord["latitude"]
avg_long = mean_coord["longitude"]

## * REPLACE ORIGINAL DC WITH AVERAGE OF ALL STOPS (CENTER OF GRAVITY SIMILAR, ASSUMING 1 PACKAGE PER
STOP)
# Approach: create a new table and modify the sequence 0 (DC coordinates) to the new one
dfr_routesDC_sorted_modDC= dfr_routesDC_sorted.copy()

# INDEXING SEQUENCE COLUMN
# to be able to locate the 0's with .loc
dfr_routesDC_sorted_modDC.set_index("sequence",inplace=True)
# locate, when the row index has a value of 0 (sequence=0)
# and the column name is "latitude" & equal it to our averages
dfr_routesDC_sorted_modDC.loc[0,"latitude"]=avg_lat
dfr_routesDC_sorted_modDC.loc[0,"longitude"]=avg_long
dfr_routesDC_sorted_modDC.reset_index(inplace=True)

column_order_modDC= ['route ID', 'stop ID', 'sequence', 'latitude', 'longitude', "DC code"]
dfr_routesDC_sorted_modDC=dfr_routesDC_sorted_modDC.reindex(columns=column_order_modDC)

# ------------------------ 5b) distance calculation drone ------------------------ #

#  projection used
wgs84_geod = Geod(ellps="WGS84")
# use Geod.inv (to calculate distance)
# angles are azimuths
# NOTE: first value should be distance 0, since they're the same coordinates

# * ORIGINAL DC COORDINATE
lon1_origDC = DClon_exact
lat1_origDC = DClat_exact
# * ALTERNATIVE AVERAGE: center of gravity of all the stops served by that DC
lon1_altDC = avg_long
lat1_altDC = avg_lat
# create new dataframe with drone distances
# here the distances of the drone will be appended
dfr_dist_drone_bystop = dfr_routesDC_sorted.copy()

# create a new list for drone distances
```

```
drone_dist_list_ORIG = []
# create a new list for the distances => from the alternative DC
drone_dist_list_ALT = []

# run function for the original DC
drone_dist_list_ORIG = drone_dist_calculator(lon1=lon1_origDC , lat1=lat1_origDC , coordinate_table= dfr_routesDC_sorted) #call
fucntion

# run function for the alternative DC
drone_dist_list_ALT = drone_dist_calculator(lon1= lon1_altDC , lat1=lat1_altDC , coordinate_table= dfr_routesDC_sorted_modDC)

## * append to the columns of df

# * original DC
# create a new column for drone distances in df, which includes the content of our distance list
dfr_dist_drone_bystop["drone distance one way"] = drone_dist_list_ORIG
# create a new column for round trip distances, multiply the original distances times 2
dfr_dist_drone_bystop["drone distance x2"] = dfr_dist_drone_bystop["drone distance one way"]*2

# * alternative DC
# create a new column for drone distances in df, which includes the content of our distance list
dfr_dist_drone_bystop["drone distance one way (alt DC)"] = drone_dist_list_ALT
# create a new column for round trip distances, multiply the original distances times 2
dfr_dist_drone_bystop["drone distance x2 (alt DC)"] = dfr_dist_drone_bystop["drone distance one way (alt DC)"]*2

# EXPORT
dfr1_drone_stops_coordinates_distances_list.append(dfr_dist_drone_bystop)

# * sum drone distances, by routeID
dfr_dist_drone_byrouteID = dfr_dist_drone_bystop.groupby("route ID")[["drone distance x2","drone distance x2 (alt DC)"]].sum()
# add back the DC code (it got lost because of groupby sum)
dfr_dist_drone_byrouteID["DC code"]= str(dcx)
# reset index & reindex with both, so that later the columns can be converted to kms
dfr_dist_drone_byrouteID.reset_index(inplace=True)
dfr_dist_drone_byrouteID.set_index(["DC code", "route ID"], inplace=True)

display(dfr_dist_drone_byrouteID)
print(type(dfr_dist_drone_byrouteID))

# ------------- 6) create table for both drone & truck distances ------------- #
# * create new df for distances per route ID (drone + truck)

# create a new colum for the truck distances
dfr_dist_both_byrouteID = dfr_dist_drone_byrouteID.copy()
dfr_dist_both_byrouteID["truck distance round trip"] = distance_list
#dfr_dist_both_byrouteID["truck route url"] = url_list # is this the error?
display(dfr_dist_both_byrouteID)
```

```
# convert drone distances to km
dfr_dist_both_byrouteID_km = dfr_dist_both_byrouteID/1000

# create the distance ratio column
    # divides the drone distance / truck distance
    # how many km the drone has to travel, per km of ground vehicle
dfr_dist_both_byrouteID_km["distance ratio (km drone: km truck)"]=(
    dfr_dist_both_byrouteID_km["drone distance x2"]/dfr_dist_both_byrouteID_km["truck distance round trip"])

# ---------------------- get amount of stops per routeID --------------------- #
## use group by and then count
### convert from series to dataframe
stops_in_route = dfr_routesDC_sorted.groupby("route ID")["stop ID"].count().to_frame()
### rename column to stops in route
stops_in_route = stops_in_route.rename(columns ={"stop ID":"stops in route"})
# reset index, in order to index DC code & route ID
stops_in_route.reset_index(inplace=True)
stops_in_route["DC code"]= str(dcx)
stops_in_route.set_index(["DC code", "route ID"], inplace=True)

# * merge it in your final table
display(stops_in_route)
print(stops_in_route.index.name)
print(stops_in_route.columns.values)

# * get total stops per DC (sum stops of each route)
total_stops_DC= stops_in_route["stops in route"].sum()
print(type(total_stops_DC))
print(total_stops_DC)

# * add the number of stops to the dataframe: using MERGE
dfr2_dist_both_byrouteID_stops_km = dfr_dist_both_byrouteID_km.merge(stops_in_route, on= ["DC code","route ID"], how= "left")
display(dfr2_dist_both_byrouteID_stops_km)
# the number of stops are found in the following df:
## stops_in_route
display(dfr2_dist_both_byrouteID_stops_km)

# append to list
dfr2_both_routeID_distances_list.append(dfr2_dist_both_byrouteID_stops_km)
```

Appendix B.2. Energy and CO_2 conversion functions

Drone energy function (single delivery)

```
# this function can be generalized to the sum of a distance and an amount of stops
## would take 1/2 the distance with payload and 1/2 without payload
## would also consider the energy needed for x amount of stops

def e_model_drone_round_several_stops(me:float, mp:float, dkm: float, stops: float) -> float:
    """Calculates energy consumption of X amount of drone round trips
        => includes energy consumption of 1x intermediate delivery

    Args:
        me (float): mass of the drone empty[kg] => without a payload, with battery
                            e.g. 14 kg for JOUAV, 20 kg for Wingcopter
        mp (float):mass of the payload [kg]
        dkm (float): distance of total trip [km] => initial trip + return trip
        stops (float): amount of intermediate deliveries

    Returns:
        float: Energy per round trip with 1x intermediate delivery of X km is: X Wh
    """
    # * Define constants
    # gravitational acceleration constant
    g = 9.8
    # energy transfer efficiency
    eta= 0.5

    # * Define further variables
    #distance of trip in m
    dm=dkm*1000
    #mass total (with payload)
    mt= mp+me
    # half trip distance
    dhalf= dm/2

    # * Define L/D ratio & energy required for landing
    # L/D depending on payload
    ## L/D with payload
    rt= -0.2242*(mp) + 8.8773
    ## L/D without payload
    re=-0.2242*(0) + 8.8773

    # le (landing energy) depending on payload in Wh
    l_e= 34.77*(mt) - 515.77 # based on total weight with payload (mt)

    # * Calulate energy per trip (with intermediate landing)
```

```
    EnergypertripJ= ((mt*g*dhalf)/(rt*eta)) + ((me*g*dhalf)/(re*eta)) + stops*(l_e*3600)
    EnergypertripWh=EnergypertripJ/3600
    EnergypertripkWh= EnergypertripWh/1000

    return EnergypertripkWh # returns energy per roundtrip with 1x intermediate delivery, for all stops grouped in that route ID [in kWh]
```

EV energy function

```
def e_model_EV(total_dist: float)-> float:
    """this function calculates the energy consumption of an EV, based on the specifications of the E-Ducato from Fiat

    Args:
        total_dist (float): give the total distance of the route in km

    Returns:
        float: returns energy consumption in kWh/km
    """
    # energy consumption WLTP combined + a 15% premium given that the delivery van
    # is gonna have a different driving behavior with a higher energy consumption
    # (conservative assumption)
    energy_EV= total_dist*0.349*1.15
    return energy_EV # kWh/km
```

Diesel energy contained function

```
def e_model_diesel(total_dist: float)-> float:
    """This function returns the energy consumption of a diesel vehicle
    based on a per km input

    Args:
        total_dist (float): give the total distance of the route in km

    Returns:
        float: returns energy consumption in kWh/km
    """
    energy_diesel_pkm= total_dist*0.8565
    return energy_diesel_pkm   # kWh/km
```

CO_2 diesel function

Based of GREET WTW model

```
def co2_model_diesel_GREET(energy_cont:float) -> float:
    """ This function calculates the CO2 emissions of a diesel vehicle,
    when given an energy value (kwh) for a certain quantity of diesel
```

```
    Args:
        energy_cont (float): insert total kWh in the quantity of diesel consumed in that trip

    Returns:
        float: kg of CO2 emissions (for that specified distance)
    """
    # * DEFINE PARAMETERS
    # to consider well to wheel emissions
    well_to_wheel_co2_emissions_GREET= 0.325699758

    # * DEFINE FUNCTION
    co2_emissions_per_route_GREET= energy_cont* well_to_wheel_co2_emissions_GREET

    return co2_emissions_per_route_GREET
```

CO_2 electricity function

```
def co2_model_electricity(kwh:float) -> float:
      """ This function calculates the average CO2 emissions based
      on electricity consumption in the U.S.

    Args:
        kWh (float): input kWh

    Returns:
        float: returns CO2 consumption, from the kWh of electricity consumed
    """
    co2_per_kwh= 0.433
    # kg CO2/kWh (national average 2019)
    # source: https://www.epa.gov/energy/greenhouse-gases-equivalencies-calculator-calculations-and-references

    co2_per_elec_consump= kwh*co2_per_kwh
    return co2_per_elec_consump
```

Appendix B.3. Multi-delivery: distances and energy consumption

This script was run in cloudio due to the required computational capacity.

```
"""
This script attempts to classify routes in groups of 1,2 or 3
   depending on the use of energy needed to make 1,2 or 3 deliveries

CUTT-OFF CRITERIA:
   if it doesnt exceed 80% of the current battery capacity of the drone
   if it's already 3 deliveries

Sequential: check point by point if we can go to the next one (if battery capacity is enough)

"""

# import needed packages
import pandas as pd
from IPython.display import display
import pyproj # for distances
geod = pyproj.Geod(ellps='WGS84')
import sys # for the arguments

# -------------------------------- IMPORT DFS -------------------------------- #
# import dataframe with all coordinates
path_coordinates_cloudio = "…"
df_coordinates = pd.read_csv(path_coordinates_cloudio)
# import dataframe with the DC coordinates
interim_data_folder_path = "…"
dc_coord_path = str(interim_data_folder_path) + "/DC COORDINATES/dc_coordinates.csv"
dc_coord_path_cloudio = "/home/karlalp/0_DC_coordinates/dc_coordinates.csv"
df_dc_coord = pd.read_csv(dc_coord_path_cloudio)

# ----------------------------- COORD DATA PRE-PROCESS ----------------------------- #

# delete ineccesary columns
df_coordinates_clean_1 = df_coordinates[["DC code","route ID","stop ID", "sequence", "latitude", "longitude"]].copy()
# sort alphabeticaly DCs, routes ID, => then sequence
df_coordinates_sorted_1 = df_coordinates_clean_1.sort_values(by=["DC code", "route ID","sequence"])
df_coordinates_sorted_1.reset_index(drop=True, inplace=True)
display(df_coordinates_sorted_1)

#  drop when sequence =0, given that sequence = 0 are the DC coordinates
df_coordinates_sorted_1 = df_coordinates_sorted_1[df_coordinates_sorted_1.sequence != 0]
df_coordinates_sorted_1.reset_index(inplace=True, drop=True)
```

```python
display(df_coordinates_sorted_1)

# %%
# * SUBSET DATAFRAME BY THE DC in the arguments
df_subset_DC = df_coordinates_sorted_1[df_coordinates_sorted_1["DC code"].isin(sys.argv[1:])].copy()
display(df_subset_DC)

# example
        # ['DAU1', 'DLA7', 'DBO2', 'DSE4']
        # !How to run on terminal: python file.py DC1 DC2 DC3 DC4

# %%
# * merge df with DC coordinates
stops= df_subset_DC.copy()
dcs = df_dc_coord.copy()
dcs.columns=['dc_latitude', 'dc_longitude', 'DC code']
display(dcs)
display(stops)

merged = pd.merge(stops, dcs, on=['DC code'])
merged.groupby(["DC code", "route ID"]).count()

# %%
# ----------------------------- ENERGY CALC. FUNCTION ----------------------------- #
# define function to calculate energy
def e_model_drone_triple_drop(me:float, mp:float, dist_list_km: list) -> float:
  # INPUTS
  #- mass of individual payload
  #- mas sof the empty drone
  #- distances list (however long)

  # * PRE-PROCESS DISTANCES (for making decision with len & converting to meters)
  # multiply by *1000 each element # inline for loop
  dist_list_m = [x*1000 for x in dist_list_km]
  # filter out 0s
  distances_filtered_m = [x for x in dist_list_m if x !=0]
  #distance of trip in m
  dm_list = distances_filtered_m
  # * Define constants
  # gravitational acceleration constant
  g = 9.8
  # energy transfer efficiency (n)
  eta= 0.5

  # * Define energy for a flying distance (dependent on L/D ratio) & energy required for landing
    # * component 1: flying distance (dependent on L/D ratio)
  def energy_distance(mass_total:float, distance_m: float, mass_payloads: float):
    # distance_m: must be in meter
```

```
        def LD_ratio(mass_payload:float):
            rt= -0.2242*(mass_payload) + 8.8773
            return rt # LD ratio
        d_e =(
            (mass_total * g * distance_m) / (LD_ratio(mass_payload= mass_payloads) * eta) )
        return d_e #(distance energy) [in Joules]

        # * component 2: energy required for (intermediate) landing
    def energy_landing(mass_payloads: float):
        l_e= (34.77*mass_payloads +179.63) * 3600
        return l_e #landing energy [in Joules]

# ------------------------ ENERGY CALC. WITH VARIABLE PAYLOADS AND DIST.  ------------------------ #

    # * IFs: CALCULATE CORRESPONDING DISTANCES
                # now use len to make a decision
    if len(distances_filtered_m) == 4: # 3 stops(payloads), 4 distances
        payload_no =3
        # * DEFINE MASSES
        m_s1= mp*payload_no+me # ! MASS TOTAL: mass payload*3 + mass empty (drone weight + battery)
        m_s2= m_s1 -mp
        m_s3= m_s2 -mp
        m_s4= me
        # * Calulate energy per trip (with intermediate landing)
            # call funtion for (intermediate landing energy and for the other one)
        Energy_total_J  =((energy_distance(mass_total=m_s1,  distance_m=dm_list[0],  mass_payloads  =  payload_no*mp))+  #  dist_1:  3
payloads
                        (energy_landing(mass_payloads=payload_no*mp))+ # int landing 1
                        (energy_distance(mass_total=m_s2, distance_m=dm_list[1], mass_payloads = (payload_no-1)*mp))+ #  dist_2: 2 payloads
                        (energy_landing(mass_payloads=(payload_no-1)*mp))+ # int landing 2
                        (energy_distance(mass_total=m_s3, distance_m=dm_list[2], mass_payloads = (payload_no-2)*mp))+ #  dist_3: 1 payload
                        (energy_landing(mass_payloads=(payload_no-2)*mp))+ # int landing 3
                        (energy_distance(mass_total=m_s4, distance_m=dm_list[3], mass_payloads = (payload_no-3)*mp)) #  dist_4: 0 payloads
                        )
        # STRUCTURE:
        # = dist 1 + dist2_dist3 + dist 4 + 3 stops (varying weights)

        EnergypertripWh=Energy_total_J/3600

        EnergypertripkWh= EnergypertripWh/1000

    elif len(distances_filtered_m) == 3: # 2 stops(payloads), 3 distances
        payload_no =2
        m_s1= mp*2+me # ! MASS TOTAL
        m_s2= m_s1 - mp
        m_s3= me
```

```
        Energy_total_J  =((energy_distance(mass_total=m_s1,  distance_m=dm_list[0],  mass_payloads  =  payload_no*mp))+  #  dist_1:  2 payloads
                    (energy_landing(mass_payloads=payload_no*mp))+ # int landing 1
                    (energy_distance(mass_total=m_s2, distance_m=dm_list[1], mass_payloads = (payload_no-1)*mp))+ #  dist_2: 1 payloads
                    (energy_landing(mass_payloads=(payload_no-1)*mp))+ # int landing 2
                    (energy_distance(mass_total=m_s3, distance_m=dm_list[2], mass_payloads = (payload_no-2)*mp)) #  dist_3: 0 payload
                    )
        # STRUCTURE:
        # = dist 1 + dist2 + dist3 + 2 stops (varying weights)

        EnergypertripWh=Energy_total_J/3600

        EnergypertripkWh= EnergypertripWh/1000

    else: # its equal to 2
        payload_no =1
        m_s1= mp+me # ! MASS TOTAL
        m_s2= me
        Energy_total_J  =((energy_distance(mass_total=m_s1,  distance_m=dm_list[0],  mass_payloads  =  payload_no*mp))+  #  dist_1:  1 payloads
                    (energy_landing(mass_payloads=payload_no*mp))+ # int landing 1
                    (energy_distance(mass_total=m_s2, distance_m=dm_list[1], mass_payloads = (payload_no-1)*mp)) #  dist_2: 1 payloads
                    )
        # STRUCTURE:
        # = dist 1 + dist2 + 1 stops (varying weights)

        EnergypertripWh=Energy_total_J/3600

        EnergypertripkWh= EnergypertripWh/1000

    return EnergypertripkWh

# %%
# ------------------------- CLASSIFICATION IN DELIVERY GROUPS ------------------------- #

#* this function classifies the coordinates within one route ID, into the groups of 1,2, or 3

def energy_groups (route_sequence):
    result =[]
    exception_counter = 0

    distance_sequence = []
    group_count = 1 # to know current group
    previous_energy = 0
    group_member_count = 0 # how many members within the current group (max 3)
    previous_point = None
    previous_distance_sequence = []
    previous_energy = 0
```

```
    for _, point in route_sequence.iterrows(): # for each point in each row, inside that section
        dc_longitude = route_sequence.dc_longitude[0]
        dc_latitude = route_sequence.dc_latitude[0]
        if group_member_count == 0:
            # * DISTANCES
            distance_sequence.append(geod.inv(dc_longitude, dc_latitude, point["longitude"], point["latitude"])[2]/1000)
            distance_sequence.append(geod.inv(dc_longitude, dc_latitude, point["longitude"], point["latitude"])[2]/1000)

            group_member_count+=1

        else:
            distance_sequence.pop() # eliminates last element
            # dist: A=>B
            distance_sequence.append(geod.inv(previous_point["longitude"], previous_point["latitude"], point["longitude"], point["latitude"])[2]/1000)
            # B => DC
            distance_sequence.append(geod.inv(point["longitude"], point["latitude"],dc_longitude, dc_latitude)[2]/1000)
            group_member_count+=1
        # * ENERGY
        energy = e_model_drone_triple_drop(me=20, mp=1.5, dist_list_km=distance_sequence)

        if energy >= 1.3024 : # if its more
            if group_member_count ==1: # !special case:  exception case if one stop would give us a higher energy
                result.extend([{"MD_group": group_count, "MD_energy":energy, "MD_distances_segments":distance_sequence, "MD_distance_per_trip": sum(distance_sequence), "group size": group_member_count }]*1)
                exception_counter +=1
                group_count+=1 # NOTE: we changed groups
                group_member_count = 0
                distance_sequence= []

            else: # you have 2 or 3 stops or coordinates
                result.extend([{"MD_group": group_count, "MD_energy":previous_energy, "MD_distances_segments":previous_distance_sequence, "MD_distance_per_trip": sum(previous_distance_sequence), "group size": previous_group_member_count }]*previous_group_member_count)
                distance_sequence= []
                # group_no
                group_count+=1 # NOTE: we changed groups
                group_member_count = 1
                distance_sequence.append(geod.inv(dc_longitude, dc_latitude, point["longitude"], point["latitude"])[2]/1000)
                distance_sequence.append(geod.inv(dc_longitude, dc_latitude, point["longitude"], point["latitude"])[2]/1000)
                energy = e_model_drone_triple_drop(me=20, mp=1.5, dist_list_km=distance_sequence)

        elif group_member_count == 3:
            result.extend([{"MD_group": group_count, "MD_energy":energy, "MD_distances_segments":distance_sequence, "MD_distance_per_trip": sum(distance_sequence), "group size": group_member_count }]*3)
            distance_sequence= []
            group_count+=1 # NOTE: we changed groups
```

```
            # when converting to dataframe if will have the dimensions of the coordinates
            group_member_count = 0

        # * back to the past
        previous_point= point # to save the last point (dataframe)
        previous_energy = energy
        previous_group_member_count = group_member_count
        previous_distance_sequence = distance_sequence.copy()

    if group_member_count < 3 and group_member_count > 0:
        result.extend([{"MD_group": group_count, "MD_energy":previous_energy, "MD_distances_segments":previous_distance_sequence,
"MD_distance_per_trip": sum(previous_distance_sequence), "group size": previous_group_member_count
}]*previous_group_member_count)


    return result

# * for loop
# get list of route IDS
routeID_list = merged["route ID"].unique().tolist()

list_dfs_energy =[]
total_routes = len(routeID_list)
route_counter = 1

for routeidx in routeID_list:
    #subset
    subset_route = merged[merged["route ID"]== routeidx].copy().reset_index(drop=True )
    dcx = subset_route["DC code"][0]

    # function
    list_energy_route = energy_groups(subset_route) # list of energy per route
    df_energy_route = pd.DataFrame.from_dict(list_energy_route).reset_index(drop=True)

    # merge energy  with subset_route
    df_concated = pd.concat(objs=[subset_route,df_energy_route], axis=1) # ignore_index=True
    list_dfs_energy.append(df_concated)
    print("route ", route_counter, " of ", total_routes)
    route_counter+=1

final_df_energy = pd.concat(list_dfs_energy) # with all route IDs in our list
display(final_df_energy)

# Save as csv
# with a variable name
list_of_DCs = str(sys.argv[1:])
final_df_energy.to_csv(f"{list_of_DCs}_energy_intermediate_multidrop.csv")
```

Appendix B.4. Contextualizing dataset processing

WorldPop data extraction

```python
# %%
import rasterio
import pandas
import geopandas as gpd
import numpy as np
from osgeo import gdal

# %%
# import the coordinate/stops dataframe
base_path= "/home/karlalp/"
path_stops = "/home/karlalp/1_WorldPop_slices/1_of_5_stops.csv"

stops_df= pandas.read_csv(path_stops)

# define .tiff path
path_tiff= (base_path + "/usa_pd_2020_1km.tiff")

# %%
# ------------------------open the tiff file-------------------------------------------------- #
tiff_file = rasterio.open(path_tiff)

# %%
# ------------------------------ get meta data ----------------------------- #

print(tiff_file.meta)
# {'driver': 'GTiff', 'dtype': 'float32', 'nodata': -99999.0, 'width': 43072, 'height': 6298, 'count': 1, 'crs': CRS.from_epsg(4326),
# 'transform': Affine(0.0083333333, 0.0, -179.15124930330558,
#        0.0, -0.0083333333, 71.39124988994655)}

print('No. of bands:',(tiff_file.count))
# No. of bands: 1

print('bounds: ',(tiff_file.bounds))

col = tiff_file.width # 43,072
rows= tiff_file.height # 6,298

# read tiff file
worldpop_raster = tiff_file.read(1)
worldpop_raster[worldpop_raster<0] = None


# # METHOD for extraction: direct from raster. Extract without polygonizing.
```

```
population_list= []
stop_counter = 1

for lat,lon in zip(stops_df.latitude, stops_df.longitude):
    #print(lat,lon)
    # for every coordinate / stop
    x = lon
    y = lat
    # get the row and column corresponding in pixel format
    row, col = tiff_file.index(x,y)
    # get the corresponding population value
    pop_value= tiff_file.read(1)[row,col]
    #print(pop_value)
    # append to the list
    population_list.append(pop_value)
    print("Coordinate: %d of 267,217"% stop_counter) # d is to print an integer
    stop_counter+=1


stops_df["population (km sq)"]=population_list

stops_df.to_csv("CORRECTED_RESULT_1_of_5_population_stop_level.csv",index=False, encoding='utf-8-sig')

# %%
tiff_file.close()
```

IPUMS NHGIS shapefile merging

```
"""
This script joins the csv and shapefile components from NHGIS, for further processing

STEPS:
1. import the shapefile contianing the geometries of the ZIP codes
2. join with the csv file of NHGIS (containing income and population data per ZIP code)
3. export the merged file

"""

# %%
import geopandas as gpd

# ----------------------------- import shapefile ----------------------------- #
us_shapefile_path= "/…/nhgis0003_shapefile_tl2021_us_zcta_2021/US_zcta_2021.shp"
# read ass a gpd df
us_shapefile = gpd.read_file(us_shapefile_path)
```

```
# know the CRS: coordinate reference system
us_shapefile.geometry.crs
   # <Derived Projected CRS: ESRI:102003>
   # Name: USA_Contiguous_Albers_Equal_Area_Conic

# reproject to our wished CRS
us_shapefile= us_shapefile.to_crs('EPSG:4326')
print(us_shapefile.head(2))
   # 0  POLYGON ((-66.66195 18.17755, -66.66191 18.177...

# clean shapefile
us_shapefile = us_shapefile[["GISJOIN", "Shape_Area","geometry", "ZCTA5CE20"]]
print(us_shapefile.head(2))

# -------------------------------- import csv income -------------------------------- #
import pandas as pd

# * income csv NHGIS
path_income = (".../nhgis0003_ds254_20215_zcta.csv")
df_income = pd.read_csv(path_income,sep=",")
display(df_income)

# clean dataframe
df_income = df_income[["GISJOIN", "ZCTA5A", "NAME_E", "AON4E001", "AORME001"]]
df_income.rename(
   columns={"ZCTA5A":"ZIP Code",'NAME_E': 'Geographic Area', 'AON4E001': 'Total population', "AORME001": "Per Capita Income in the Past 12 Months (in 2021 Inflation-Adjusted Dollars)"}, inplace=True)
display(df_income)

# -------------------------- merge shapefile and csv ------------------------- #

# Perform the join on the "GISJOIN" column
merged_df = us_shapefile.merge(df_income, on='GISJOIN')
display(merged_df)

# Save the merged GeoDataFrame to a new shapefile
merged_df.to_file('.../income_merged_shapefile.shp')
```

IPUMS NHGIS data extraction

```
"""
This script gets the corresponding income values per ZIP code, given the coordinate list

STEPS:
1. import the income information with geometries
2. join with the coordinate dataframe, uses an sindex ()
```

```
"""

# %%
import geopandas as gpd
import pandas as pd
from shapely.geometry import Point
from tqdm import tqdm # progress bar
from IPython.display import display

# %%
# ---------------------------- import dataframes ---------------------------- #
# * shapefile with income data
base_path= "/home/karlalp/2 income and ZIP codes"
path_shapefile = (base_path+
                "/income_merged_shapefile.shp")
income_shapefile = gpd.read_file(path_shapefile)

print(income_shapefile.head(2))

# * 9000 subset cooridnates import
path_stops = "/home/karlalp/0_corrected coordinates/CORRECTED90001_stops_combined_csv_excluding_exceptions.csv"
df_stops = pd.read_csv(path_stops)
display (df_stops)
# 1336872 rows × 10 columns

# ------------------------ geometry from lat & long ------------------------ #
# convert lat and long to points
geometry = [Point(xy) for xy in zip(df_stops['longitude'], df_stops['latitude'])]

# Convert the Pandas dataframe to a GeoDataFrame by creating a new column called 'geometry' with Point objects
df_stops = gpd.GeoDataFrame(df_stops, geometry=geometry, crs=income_shapefile.crs)
display(df_stops)

# ---------------------------- merge with sjoin ---------------------------- #

# Create a spatial index for the merged GeoDataFrame
# R-tree algorithm: to improve the performance of the spatial join for large datasets
income_shapefile.sindex

# Perform the spatial join using the sjoin() function with 'op='intersects'' to use the spatial index
joined = gpd.sjoin(df_stops, income_shapefile, predicate='intersects', how="left") # TODO try how="left"
display(joined)

# Create a progress bar for the sjoin operation
tqdm.pandas(desc="Joining data")

# Apply the progress bar to the sjoin operation
joined.progress_apply(lambda x: x, axis=1)
```

```
# rename columns
clean_joined = joined.drop(["index_right", "geometry"], axis=1)
clean_joined.rename(
   columns={"Per Capita":"Per Capita Income in ZIP code", "Total popu":"Population in ZIP code"}, inplace=True)
display(clean_joined)
clean_joined.to_csv("LEFT_CORRECTED_1RESULTS_income_stop_level.csv",index=False, encoding='utf-8-sig')

# find NaN values
clean_joined['Per Capita Income'].isnull().values.any()
clean_joined[clean_joined['Per Capita Income'].isna()]
# 69 rows dont have income values
# small ZIP codes
# comes from the dataset
clean_joined[clean_joined['Per Capita Income'].isnull()]
clean_joined[clean_joined['Population in ZIP code'].isnull()]
# 7 coordinates that were not found a corresponding ZIP code
# in SEATTLE (DSE2)
```

Appendix B.5. Packages utilized in conda environment

name: GISenv
channels:
  - conda-forge
  - defaults
dependencies:
  - affine=2.4.0=pyhd8ed1ab_0
  - anyio=3.6.2=pyhd8ed1ab_0
  - appdirs=1.4.4=pyh9f0ad1d_0
  - appnope=0.1.3=pyhd8ed1ab_0
  - argon2-cffi=21.3.0=pyhd8ed1ab_0
  - argon2-cffi-bindings=21.2.0=py38hef030d1_3
  - asttokens=2.2.1=pyhd8ed1ab_0
  - attrs=22.2.0=pyh71513ae_0
  - babel=2.11.0=pyhd8ed1ab_0
  - backcall=0.2.0=pyh9f0ad1d_0
  - backports=1.0=pyhd8ed1ab_3
  -backports.functools_lru_cache=1.6.4
=pyhd8ed1ab_0
  - beautifulsoup4=4.11.1=pyha770c72_0
  - bleach=6.0.0=pyhd8ed1ab_0
  - blosc=1.21.2=hebb52c4_0
  - boost-cpp=1.78.0=h8b082ac_1
  - branca=0.6.0=pyhd8ed1ab_0
  - brotli=1.0.9=hb7f2c08_8
  - brotli-bin=1.0.9=hb7f2c08_8
  - brotlipy=0.7.0=py38hef030d1_1005
  - bzip2=1.0.8=h0d85af4_4
  - c-ares=1.18.1=h0d85af4_0
  - ca-certificates=2022.12.7=h033912b_0
  - cairo=1.16.0=h904041c_1014
  - certifi=2022.12.7=pyhd8ed1ab_0
  - cffi=1.15.1=py38hb368cf1_3
  - cfitsio=4.2.0=hd56cc12_0
  - charset-normalizer=2.1.1=pyhd8ed1ab_0
  - click=8.1.3=unix_pyhd8ed1ab_2
  - click-plugins=1.1.1=py_0
  - cligj=0.7.2=pyhd8ed1ab_1
  - colorama=0.4.6=pyhd8ed1ab_0
  - country_converter=0.7.7=pyhd8ed1ab_0
  - cryptography=39.0.0=py38h4257468_0
  - curl=7.87.0=h6df9250_0
  - cycler=0.11.0=pyhd8ed1ab_0
  - debugpy=1.6.6=py38h4cd09af_0
  - decorator=5.1.1=pyhd8ed1ab_0
  - defusedxml=0.7.1=pyhd8ed1ab_0
  - entrypoints=0.4=pyhd8ed1ab_0
  - et_xmlfile=1.1.0=pyhd8ed1ab_0
  - executing=1.2.0=pyhd8ed1ab_0
  - expat=2.5.0=hf0c8a7f_0
  - fiona=1.9.0=py38h4a1972a_0
  - flit-core=3.8.0=pyhd8ed1ab_0
  - folium=0.14.0=pyhd8ed1ab_0
  - font-ttf-dejavu-sans-mono=2.37=hab24e00_0
  - font-ttf-inconsolata=3.000=h77eed37_0
  - font-ttf-source-code-pro=2.038=h77eed37_0
  - font-ttf-ubuntu=0.83=hab24e00_0
  - fontconfig=2.14.2=h5bb23bf_0
  - fonts-conda-ecosystem=1=0
  - fonts-conda-forge=1=0
  - fonttools=4.38.0=py38hef030d1_1
  - freetype=2.12.1=h3f81eb7_1
  - freexl=1.0.6=hb7f2c08_1
  - gdal=3.6.2=py38h86a93f0_3
  - geopandas=0.11.0=pyhd8ed1ab_0
  - geopandas-base=0.11.0=pyha770c72_0
  - geos=3.11.1=hf0c8a7f_0
  - geotiff=1.7.1=h2039a76_5
  - gettext=0.21.1=h8a4c099_0
  - giflib=5.2.1=hbcb3906_2
  - hdf4=4.2.15=h7aa5921_5
  - hdf5=1.12.2=nompi_h48135f9_101
  - icu=70.1=h96cf925_0
  - idna=3.4=pyhd8ed1ab_0
  - importlib-metadata=6.0.0=pyha770c72_0
  - importlib_metadata=6.0.0=hd8ed1ab_0
  - importlib_resources=5.10.2=pyhd8ed1ab_0
  - ipykernel=6.13.0=py38h60dac5d_0
  - ipython=8.4.0=pyhd1c38e8_1
  - ipython_genutils=0.2.0=py_1
  - jedi=0.18.2=pyhd8ed1ab_0
  - jinja2=3.1.2=pyhd8ed1ab_1
  - joblib=1.2.0=pyhd8ed1ab_0
  - jpeg=9e=hac89ed1_2
  - json-c=0.16=h01d06f9_0
  - json5=0.9.5=pyh9f0ad1d_0
  - jsonschema=4.17.3=pyhd8ed1ab_0
  - jupyter_client=8.0.2=pyhd8ed1ab_0
  - jupyter_core=5.2.0=py38h50d1736_0
  - jupyter_server=1.23.5=pyhd8ed1ab_0
  - jupyterlab=3.3.4=pyhd8ed1ab_0

- jupyterlab_pygments=0.2.2=pyhd8ed1ab_0
- jupyterlab_server=2.19.0=pyhd8ed1ab_0
- kaleido-core=0.2.1=h0d85af4_0
- kealib=1.5.0=h5c1f988_0
- kiwisolver=1.4.4=py38h98b9b1b_1
- krb5=1.20.1=h049b76e_0
- lcms2=2.14=h29502cd_1
- lerc=4.0.0=hb486fe8_0
- libaec=1.0.6=hf0c8a7f_1
- libblas=3.9.0=16_osx64_openblas
- libbrotlicommon=1.0.9=hb7f2c08_8
- libbrotlidec=1.0.9=hb7f2c08_8
- libbrotlienc=1.0.9=hb7f2c08_8
- libcblas=3.9.0=16_osx64_openblas
- libcurl=7.87.0=h6df9250_0
- libcxx=14.0.6=hccf4f1f_0
- libdeflate=1.17=hac1461d_0
- libedit=3.1.20191231=h0678c8f_2
- libev=4.33=haf1e3a3_1
- libffi=3.4.2=h0d85af4_5
- libgdal=3.6.2=h623d8b8_3
- libgfortran=5.0.0=11_3_0_h97931a8_27
- libgfortran5=11.3.0=h082f757_27
- libglib=2.74.1=h4c723e1_1
- libiconv=1.17=hac89ed1_0
- libjpeg-turbo=2.1.4=hb7f2c08_0
- libkml=1.3.0=haeb80ef_1015
- liblapack=3.9.0=16_osx64_openblas
- libnetcdf=4.8.1=nompi_hc61b76e_106
- libnghttp2=1.51.0=he2ab024_0
- libopenblas=0.3.21=openmp_h429af6e_3
- libpng=1.6.39=ha978bb4_0
- libpq=15.1=h3640bf0_3
- librttopo=1.1.0=h9461dca_12
- libsodium=1.0.18=hbcb3906_1
- libspatialindex=1.9.3=he49afe7_4
- libspatialite=5.0.1=hc1c2c66_22
- libsqlite=3.40.0=ha978bb4_0
- libssh2=1.10.0=h47af595_3
- libtiff=4.5.0=hee9004a_2
- libwebp-base=1.2.4=h775f41a_0
- libxcb=1.13=h0d85af4_1004
- libxml2=2.10.3=hb9e07b5_0
- libzip=1.9.2=h6db710c_1
- libzlib=1.2.13=hfd90126_4
- llvm-openmp=15.0.7=h61d9ccf_0
- lz4-c=1.9.4=hf0c8a7f_0
- mapclassify=2.5.0=pyhd8ed1ab_1
- markupsafe=2.1.2=py38hef030d1_0
- mathjax=2.7.7=h694c41f_3
- matplotlib=3.5.3=py38h50d1736_2
- matplotlib-base=3.5.3=py38hae485fc_2
- matplotlib-inline=0.1.6=pyhd8ed1ab_0
- mistune=2.0.4=pyhd8ed1ab_0
- munch=2.5.0=py_0
- munkres=1.1.4=pyh9f0ad1d_0
- nbclassic=0.4.8=pyhd8ed1ab_0
- nbclient=0.7.2=pyhd8ed1ab_0
- nbconvert=7.2.9=pyhd8ed1ab_0
- nbconvert-core=7.2.9=pyhd8ed1ab_0
- nbconvert-pandoc=7.2.9=pyhd8ed1ab_0
- nbformat=5.7.3=pyhd8ed1ab_0
- ncurses=6.3=h96cf925_1
- nest-asyncio=1.5.6=pyhd8ed1ab_0
- networkx=3.0=pyhd8ed1ab_0
- notebook-shim=0.2.2=pyhd8ed1ab_0
- nspr=4.35=hea0b92c_0
- nss=3.78=ha8197d3_0
- numpy=1.22.3=py38h7eae8df_2
- openjpeg=2.5.0=h13ac156_2
- openpyxl=3.0.9=pyhd8ed1ab_0
- openssl=3.0.7=hfd90126_2
- packaging=23.0=pyhd8ed1ab_0
- pandas=1.4.2=py38h2b30649_2
- pandoc=2.19.2=h694c41f_1
- pandocfilters=1.5.0=pyhd8ed1ab_0
- parso=0.8.3=pyhd8ed1ab_0
- patsy=0.5.3=pyhd8ed1ab_0
- pcre2=10.40=h1c4e4bc_0
- pexpect=4.8.0=pyh1a96a4e_2
- pickleshare=0.7.5=py_1003
- pillow=9.4.0=py38h48932a6_0
- pip=22.0.4=pyhd8ed1ab_0
- pixman=0.40.0=hbcb3906_0
- pkgutil-resolve-name=1.3.10=pyhd8ed1ab_0
- platformdirs=2.6.2=pyhd8ed1ab_0
- plotly=5.9.0=pyhd8ed1ab_0
- pooch=1.6.0=pyhd8ed1ab_0
- poppler=22.12.0=h6e9091c_1
- poppler-data=0.4.11=hd8ed1ab_0
- postgresql=15.1=hbea33b9_3
- proj=9.1.0=hf909084_1
- prometheus_client=0.16.0=pyhd8ed1ab_0
- prompt-toolkit=3.0.36=pyha770c72_0
- psutil=5.9.4=py38hef030d1_0
- pthread-stubs=0.4=hc929b4f_1001

- ptyprocess=0.7.0=pyhd3deb0d_0
- pure_eval=0.2.2=pyhd8ed1ab_0
- pycparser=2.21=pyhd8ed1ab_0
- pygments=2.14.0=pyhd8ed1ab_0
- pyopenssl=23.0.0=pyhd8ed1ab_0
- pyparsing=3.0.9=pyhd8ed1ab_0
- pyproj=3.4.1=py38hb2f5155_0
- pyrsistent=0.19.3=py38hef030d1_0
- pysocks=1.7.1=pyha2e5f31_6
- python=3.8.15=hf9b03c3_1_cpython
- python-dateutil=2.8.2=pyhd8ed1ab_0
- python-fastjsonschema=2.16.2=pyhd8ed1ab_0
- python_abi=3.8=3_cp38
- pytz=2022.7.1=pyhd8ed1ab_0
- pyxlsb=1.0.9=pyhd8ed1ab_0
- pyzmq=25.0.0=py38h0b711fd_0
- rasterio=1.3.4=py38h028a342_0
- rasterstats=0.17.0=pyhd8ed1ab_0
- readline=8.1.2=h3899abd_0
- requests=2.28.2=pyhd8ed1ab_0
- rioxarray=0.13.3=pyhd8ed1ab_0
- rtree=1.0.1=py38hc59ffc2_1
- scikit-learn=1.2.1=py38h573ff9c_0
- scipy=1.10.0=py38hfb8b963_0
- seaborn=0.12.0=hd8ed1ab_0
- seaborn-base=0.12.0=pyhd8ed1ab_0
- send2trash=1.8.0=pyhd8ed1ab_0
- setuptools=66.1.1=pyhd8ed1ab_0
- shapely=1.8.5=py38h4d28eb3_2
- simplejson=3.18.1=py38hef030d1_0
- six=1.16.0=pyh6c4a22f_0
- snappy=1.1.9=h225ccf5_2
- sniffio=1.3.0=pyhd8ed1ab_0
- snuggs=1.4.7=py_0
- soupsieve=2.3.2.post1=pyhd8ed1ab_0
- sqlite=3.40.0=h9ae0607_0
- stack_data=0.6.2=pyhd8ed1ab_0
- statsmodels=0.13.5=py38hbd87e4b_2
- tenacity=8.1.0=pyhd8ed1ab_0
- terminado=0.17.1=pyhd1c38e8_0
- threadpoolctl=3.1.0=pyh8a188c0_0
- tiledb=2.13.2=h8b9cbf0_0
- tinycss2=1.2.1=pyhd8ed1ab_0
- tk=8.6.12=h5dbffcc_0
- tornado=6.2=py38hef030d1_1
- tqdm=4.64.0=pyhd8ed1ab_0
- traitlets=5.9.0=pyhd8ed1ab_0
- typing-extensions=4.4.0=hd8ed1ab_0
- typing_extensions=4.4.0=pyha770c72_0
- tzcode=2022g=hb7f2c08_0
- tzdata=2022g=h191b570_0
- unicodedata2=15.0.0=py38hef030d1_0
- urllib3=1.26.14=pyhd8ed1ab_0
- wcwidth=0.2.6=pyhd8ed1ab_0
- webencodings=0.5.1=py_1
- websocket-client=1.5.0=pyhd8ed1ab_0
- wheel=0.38.4=pyhd8ed1ab_0
- xarray=2023.1.0=pyhd8ed1ab_0
- xerces-c=3.2.4=h2007e90_1
- xlrd=2.0.1=pyhd8ed1ab_3
- xorg-libxau=1.0.9=h35c211d_0
- xorg-libxdmcp=1.1.3=h35c211d_0
- xyzservices=2022.9.0=pyhd8ed1ab_0
- xz=5.2.6=h775f41a_0
- zeromq=4.3.4=he49afe7_1
- zipp=3.12.0=pyhd8ed1ab_0
- zlib=1.2.13=hfd90126_4
- zstd=1.5.2=hbc0c0cd_6

Appendix C

Appendix C.1. Header of the final generated dataframe provided in the supplementary material

Column name	Exemplary value
DC code	DAU1
route ID	RouteID_00ae3f5e-a708-4c37-b9c4-ebd3964dbdac
Drone (orig DC) [km]	3891.815505
Drone (alt DC) [km]	1388.187933
Dist. ratio (orig DC) [km drone: km truck]	49.15447647
stops in route	134
Drone (orig DC) [kWh per route ID]	81.5966064
Drone (orig DC) [kWh per package]	0.608929898
EV (orig DC) [kWh per route ID]	31.77696652
EV (orig DC) [kWh per package]	0.237141541
Diesel (orig DC) [kWh per route ID]	67.8135588
Diesel (orig DC) [kWh per package]	0.506071334
Drone (orig DC) [kg of CO2 per route ID]	35.33133057
Drone (orig DC) [kg of CO2 per package]	0.263666646
EV (orig DC) [kg of CO2 per route ID]	13.7594265
EV (orig DC) [kg of CO2 per package]	0.102682287
Diesel (orig DC) [kg of CO2 per route ID]	22.08685969
Diesel (orig DC) [kg of CO2 per package]	0.164827311
Dist ratio (alt DC) [km drone: km truck]	17.53311558
Drone (alt DC) [kWh per route ID]	49.0855922
Drone (alt DC) [kWh per package]	0.36631039
Drone (alt DC) [kg of CO2 per route ID]	21.25406142
Drone (alt DC) [kg of CO2 per package]	0.158612399
Drone (multi-delivery) [km]	1545.759924
Drone (multi-delivery) [kWh per route ID]	58.59804128
Drone (multi-delivery) [kg of CO2 per route ID]	25.37295187
Drone (multi-delivery) [kWh per package]	0.437298816
Drone (multi-delivery) [kg of CO2 per package]	0.189350387
Dist ratio (multi-delivery) [km drone: km truck]	19.52328411
Drone (alt DC multi-delivery) [km]	697.3707318

Drone (alt DC multi-delivery) [kWh per route ID]	47.8683626
Drone (alt DC multi-delivery) [kg of CO2 per route ID]	20.72700101
Drone (alt DC multi-delivery) [kWh per package]	0.357226587
Drone (alt DC multi-delivery) [kg of CO2 per package]	0.154679112
Dist ratio (alt DC multi-delivery) [km drone: km truck]	8.807944051
EV (orig DC) [km]	79.1752
Diesel (orig DC) [km]	79.1752
population (km sq)	2575.800111
ZIP Code	78751.05224
Population in ZIP code	15633.99254
Per Capita Income in ZIP code	51746.79851
population (1 arc-second)	2.645743506
City	Austin
Drone (orig DC)-EV [kWh per route ID]	49.81963988
Drone (multi-delivery)-EV [kWh per route ID]	26.82107476
Drone (alt DC)-EV [kWh per route ID]	17.30862568
Drone (alt DC multi-delivery)-EV [kWh per route ID]	16.09139608
Drone (orig DC)-Diesel [kg of CO2 per route ID]	13.24447088
Drone (multi-delivery)-Diesel [kg of CO2 per route ID]	3.286092183
Drone (alt DC)-Diesel [kg of CO2 per route ID]	-0.832798268
Drone (alt DC multi-delivery)-Diesel [kg of CO2 per route ID]	-1.359858683
Drone (alt DC multi-delivery)-EV [kg of CO2 per route ID]	6.967574504
Drone (alt DC multi-delivery) better than diesel	TRUE
EV (orig DC) [km per package]	0.590859701
Diesel (orig DC) [km per package]	0.590859701
avg group size (alt DC multi-delivery)	2.955555556
Drone (alt DC multi-delivery) better than EV	FALSE
Drone (orig DC) [kg of CO2 per route ID] (alt elec mix)	3.533133057
Drone (orig DC) [kg of CO2 per package] (alt elec mix)	0.026366665
EV (orig DC) [kg of CO2 per route ID] (alt elec mix)	1.37594265
EV (orig DC) [kg of CO2 per package] (alt elec mix)	0.010268229
Diesel (orig DC) [kg of CO2 per route ID] (alt elec mix)	2.208685969
Diesel (orig DC) [kg of CO2 per package] (alt elec mix)	0.016482731
Drone (alt DC) [kg of CO2 per route ID] (alt elec mix)	2.125406142
Drone (alt DC) [kg of CO2 per package] (alt elec mix)	0.01586124
Drone (multi-delivery) [kg of CO2 per route ID] (alt elec mix)	2.537295187

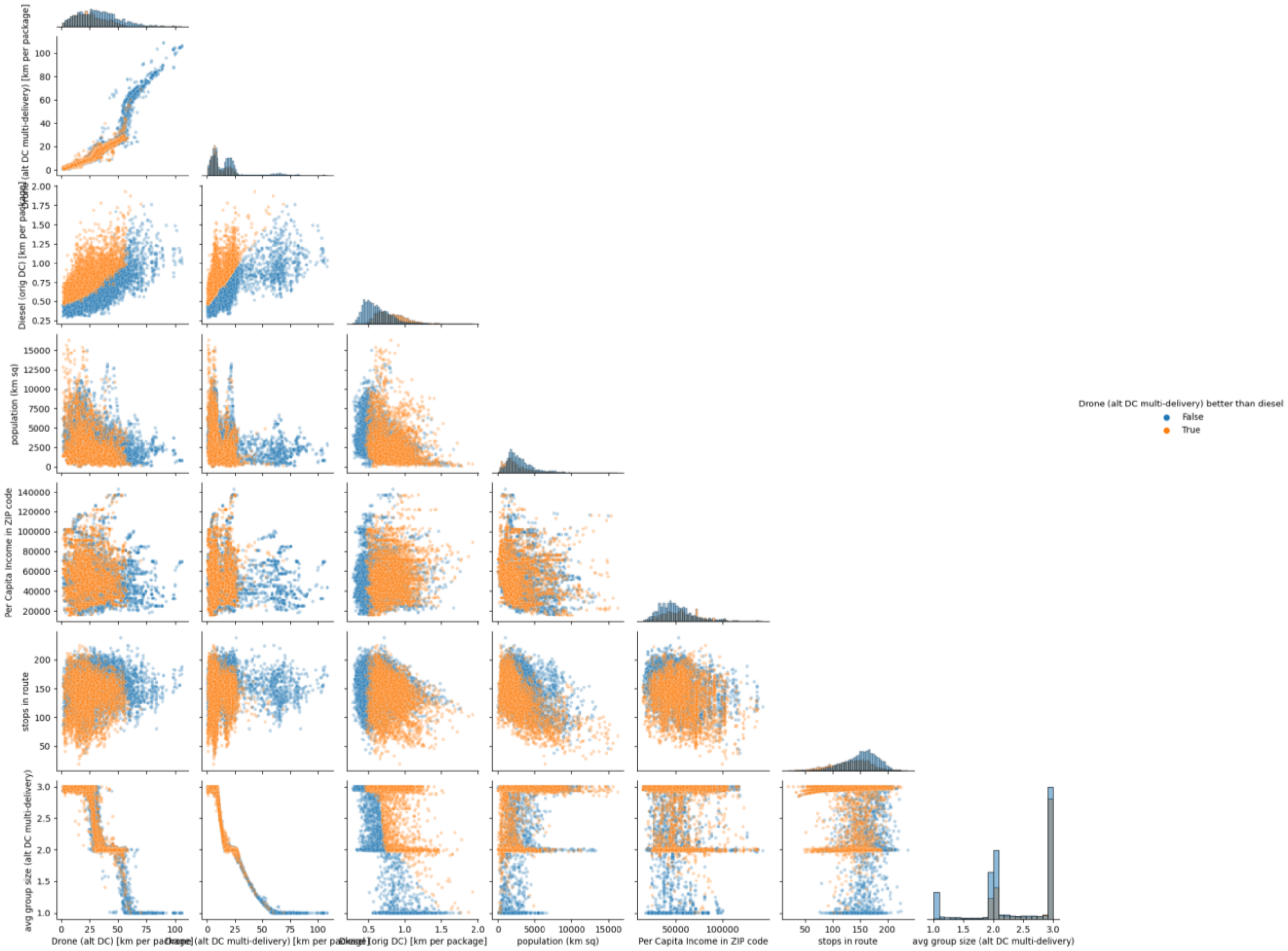
Drone (multi-delivery) [kg of CO2 per package] (alt elec mix)	0.018935039
Drone (alt DC multi-delivery) [kg of CO2 per route ID] (alt elec mix)	2.072700101
Drone (alt DC multi-delivery) [kg of CO2 per package] (alt elec mix)	0.015467911

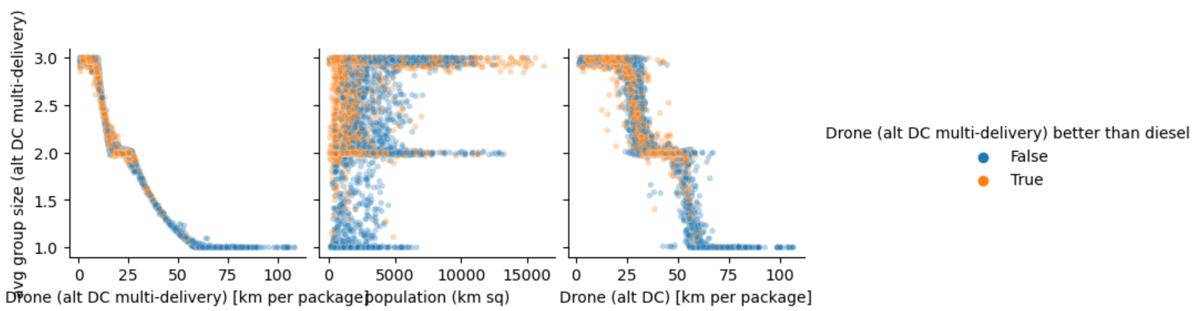
Appendix C.2. Equivalent route comparison summary statistics

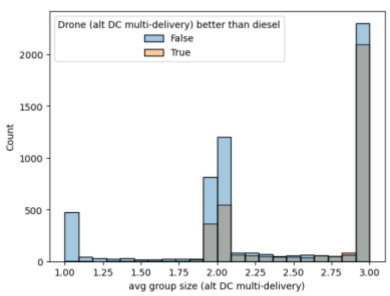
D4 and G2 equivalent-route comparison: set of summary statistics comparing the better and worse performing routes

Drone (alt DC multi-delivery) & Diesel (orig DC)		**Worse performing**	**Better performing**
Drone (alt DC multi-delivery) [km per package]	mean	10.93	18.79
	std	7.64	18.33
	median	8.09	14.86
Avg. group size (alt DC multi-delivery)	mean	2.64	2.34
	std	0.45	0.64
	median	2.95	2.17
Drone (alt DC) [km per package]	mean	25.08	33.03
	std	12.87	17.67
	median	23.55	31.48s
Dist. ratio (alt DC multi-delivery) [km drone: km truck]	mean	12.55	26.58
	std	6.75	18.60
	median	10.54	24.17
Drone (alt DC multi-delivery) [kg of CO2 per package]	mean	0.18	0.23
	std	0.04	0.10
	median	0.17	0.20
Drone (alt DC multi-delivery) [kWh per package]	mean	0.43	0.52
	std	0.09	0.22
	median	0.40	0.47

Appendix C.3. Complete pairplot sets







NTNU

Norwegian University of
Science and Technology