Aninda Bhowmik

Sustainable Freight Analytics: Capacity and Emissions Visualization using Python

School of Technology and Innovations
Master's thesis in Industrial Systems Analytics

Vaasa 2025

ACKNOWLEDGEMENTS

I am truly grateful to Almighty First that I got too many opportunities to learn from the very basic during this Thesis.

I would like to express my sincere gratitude to **Professor Petri Helo** for his invaluable guidance, insightful feedback, and continuous encouragement throughout the development of this thesis. I am debted to his appraisal by granting me the opportunity to complete my Masters' Thesis under his direct supervision. His clear directions, critical questions, and emphasis on methodological rigor significantly shaped my understanding of the research process and helped me refine both the analytical approach and the overall structure of my work.

I am equally grateful to **Clara Rajalehto** for her exceptional support, patience, and practical guidance. Her detailed explanations, timely feedback, and willingness to clarify complex data and methodological issues were instrumental in completing this study. Her support not only improved my understanding of my basics but also strengthened my confidence in applying independent problem-solving and analytical skills.

My heartfelt thanks and gratitude go to **Nitish Singh,** my dear friend and owner of the YouTube channel **CampusX.** He personally assisted me to learn Python coding and helps by guiding me a lot to build the application from his huge experiences and knowledge. I also got best lesson for connecting Python with CSS, HTML and MySQL to run and share data smoothly to this application without error.

On second semester, I started learning Python under the supervision of **Jani Päivärinta.** An exceptional personal to whom I am grateful for his solid teaching style to learn coding. I tried to follow the logic mostly learnt in his classes.

Aninda Bhowmik
Vaasa, January 11, 2026

UNIVERSITY OF VAASA
School of Technology and Innovations
Author: Aninda Bhowmik
Title of the Thesis: Sustainable Freight Analytics: Capacity and Emissions Visualization using Python
Degree: Master of Science in Technology
Programme: Industrial Systems Analytics
Supervisor: Professor Petri Helo
Co-Supervisor: Clara Rajalehto
Year: 2025 **Sivumäärä:** 78

ABSTRACT :

In the present day, people are relying on to proceed with their business based on accurate data. The mission of this Thesis has clear goal to prepare an authentic dashboard which can calculate minimum path using the most popular GoogleMaps to compare human centric fetched data with the technology based accumulated data. Obviously, technology helps us to make our process easy to figure out the actual distances where multiple emission factors help to determine the actual WTW emission in CO_2e Kg. The visualization of components inside maps clearcuts the authenticity of location with prime impact. The Live Calculation table works in parallel with the movement of the component in each leg. The calculation comes automatically after touching each leg with load and unload data. This thesis creates a detailed CO2 emission calculator in accordance with the requirements of ISO 14083 for refrigerated road transport. Propulsion fuel, TRU fuel consumption and refrigerant leakage are combined in single WTW model. An integrated time-based (L/h) and distance-based (L/100 km) TRU energy model is proposed, utilizing parameter sets (a, b), which are dependent on driving environments like town, city, mixed and linehaul operations. Refrigerant leakage losses are apportioned on annual basis to individual transport chain components based on distance-related activity shares following guidelines established by ISO, GLEC. The model also includes dynamic vehicle weight, route-based division and average-speed class and TKM–specialized assignment for consistent per-leg emissions.

KEYWORDS: Python, CO_2e Kg, ISO 14083, GLEC, Longitude, Latitude, Current Weight, SFD

Table of Contents

List of Figures

List of Tables

1 Introduction

In many economies, transportation has been remarked for its far-reaching contribution to strengthening a sustainable effect, which also has serious environmental effects. The concurrent research indicates that the emissions from freight transport account for a significant 8% of GHG emissions and that the process is severe to continue growing, roughly extendable by 2050 due to the massive increment of global supply chains. In part of the information, the possibility of expanding tough GHG emissions has become obligatory to follow the limits of the Paris Agreement; the freight transport sector must be aware of the GHG emissions that need to be controlled (GLEC, 2025, p. 2).

Responding accurately, the ISO 14083:2023 standard, the Global Logistics Emissions Council (GLEC) has been constructed to reduce greenhouse gases and impart guidelines regarding the measurement of quantity and preserving the details of freight emissions. The GLEC Framework v3.1 comprises the ISO 14083 provisions and suggests an amalgamated system of methods to ease the calculation for the freight transport sector that comes with a plausible report to ensure proper well-to-wheel (WTW) GHG emissions (GLEC, 2025, p. 11).

A “well-to-wheel” (WTW) perspective is constructed inside the emissions framework in accounting for “well-to-tank” (WTT) emissions linked to the provision of energy and “tank-to-wheel” (TTW) emissions exposed during operational use (GLEC, 2025, pp. 18–19). This spacious boundary secures the complete fuel and energy life cycle emissions of freight journeys. Direct (operational) and indirect (energy provision) emissions are also included, thus enhancing comparability across supply transport modes and chains. Guidelines of GLEC and ISO 14083 provide companies with consistency for seamless accounting, verifying emissions, and cover reporting in logistics, maintaining all structures at a global equivalency.

Adhering to the glimpse of facts, the GLEC Framework also suggests hub operation categories (HOCs) and transport operation categories (TOCs), maintaining standardized

structural units for congregating units like transport and logistics activities (GLEC, 2025, p. 11). An authorization is scheduled for firms to establish representative emission intensities and allocate proper emissions within and between stages of the supply chain. The Framework specifically sets forth the categorization of those emissions within the Scope 1, 2, or 3 based on the reporting organization's whereabouts in the value chain, and not on direct ownership of assets (GLEC, 2025, p. 11).

The major focus remains on maintaining the quality of the data. GLEC section 3 supervises that emissions results have a strong dependency on the type of data stored as input, whether the data standard is modeled, primary, or default (GLEC, 2025, p. 26). The operations execute the data in a direct approach, that is, measuring the standard of fuel usage, distances (SFD), or various factors, which refers to primary data and the most precise and useful type. The estimation of modeled data, which uses predictive parameters, route or vehicle type, and default data all have average credibility, which is set on the entire industry, aimed mostly at first-cut analysis. Default values are statistically traditional and well-constructed to overestimate emissions to motivate firms to gather more primary data (GLEC, 2025, p. 92). However, the identification will be hindered by over-reliance on predefined factors of reduction that connect potentiality (GLEC, 2025, p. 92) and may bias benchmarking outcomes.

The credibility and assurance, the words are mostly emphasized in the GLEC framework, and transparency in the data sources is highly appreciated. The firms come into agreement to reveal the percentage of primary and secondary data used as ISO 14083 compels them to follow the guidelines and advises them to undergo scrutiny with a third party to ensure the validation of the assumptions and the consequences (GLEC, 2025, p. 59).

Although ISO 14083 and GLEC v3.1 have introduced the best level of procedures to standardize, there remains a theoretical void. In the query, there is a gap in verifiable proof that measures the impact of confirming data quality on the emissions enumerated. Organizations often must make an experimental judgment about the dealings between the costs and intricacy of collecting more detailed data versus the value of more detailed

data. Highlighting recent research states that “What level of perfection is fair enough?” best condenses the problem of effective emission accounting and the underlying policy.

My thesis will address this issue. Using a unique dataset from a renowned retailer's logistics network, my study will focus on analyzing emissions from point to point of the road, transport, to examine the different outcomes brought out by divergent detail levels (default, modeled, and primary) data. The research design follows ISO 14083 and GLEC 3.1, ensuring the WTW system is fully covered with implications of my own calculation. This thesis inquiries into the emphasis of data input granularity on emissions estimates. In doing so, it documents the trade-off between access to data and calculation accuracy and the relevance of data to the decision at hand.

1.1 Background

Logistics and transport contribute significantly to global CO_2 emissions, and the transportation of goods is a significant source. Measurement and emission reduction are becoming important as the sustainability goals are becoming ambitious. Most of the logistics companies are unlucky to have gaps in their data: they do not have accurate distance records, and vouchers or financial invoices do not include real routes. This paper addresses such issues by designing an interactive dashboard. State-of-the-art routing algorithms are used in the dashboard to give the best possible estimations of the shortest truck routes and thus give more accurate estimations of the emissions. When using the data of real roads with advanced algorithms, the dashboard provides a much better quality of emissions calculations when logistics work goes on.

Given the pace of progress in setting requirements for reporting emissions e.g.: GLEC (the Global Logistics Emissions Council) Framework and ISO 14083 - it needs to have data-driven tools that can calculate, show and allocate emissions across individual delivery legs transparently. This thesis stems from that requirement, driven by the need to solve the practical problem of translation of operational transport data to its standardized emission outputs.

1.2 Purpose of the Study

Creating an Interactive dashboard using Python was the first choice which demonstrates every calculation with best effort. Focusing on learning Python to develop the dashboard is helping continuously to learn HTML and CSS in parallel. Every research project starts with a moment of confusion when the tools at hand fail to solve the questions that arise in actual work. At the beginning, the journey seemed same as others. In the first few weeks of analyzing with different transport data in excel file named: biokasu and dieselkasu, it was important to understand for familiarizing with the data structure that includes numbers – distances, tonnes, durations, fuel amounts etc., but searching was going on for better understanding with studying. The spreadsheets had useful information which helps to analysis with both excel and Python. On every instruction through email, the emission of CO_2e is precisely calculated in every leg and finally in a total. The application was tried to develop would illustrate the total successful calculation of CO_2e emission with more specific ways.

First, focusing on how to calculate the distances from one place to another using longitude and latitude. Many sites on the internet are searched and finally learnt from a YouTuber on how to do this. Then, drawing polyline between the distance of the sources and destination which was an intrinsic one.

GLEC Framework and ISO 14083 documentation demonstrates on how to calculate some of these impact categories, industries know what they need to calculate but very few of the tools show at a part level how they link together. The standards relate to TCEs, SFD, WTW energy streams, allocation but there was no system to merge it all together in one interactive tool.

The planning for constructing a “Live Calculation” table that would be capable of tracking the journey of a truck from source point to every destination point:

- In the table, it must store the location of Source and destination based on Longitude and Latitude.

- While crossing one point to another, a new row will be generated with detailed information.
- It can compute the calculation of “Consumption car × km”, Share of Capacity, Share of Consumption, CO_2e emission for each leg.
- Table will also store load value and unload value and measure the current weight by subtracting from load and unload to measure the actual weight carried by truck.
- In table, the SFD and actual distance will be automatically stored based on the distance found by calling Google Map API.

In addition, the target is to calculate emissions and presented in this “Live Calculation” table; so that the logic has been implemented to focus on how emissions worked. For this purpose, the research was directed at:

- To analysis the shortest possible path, focusing on the algorithm helps to determine the actual route data while running the program.
- K-means algorithm, helicopter algorithm, travelling salesperson algorithm modeling helps to identify the best suit for calculating shortest possible path.
- Finally, searching in many ways, the “Greedy Algorithm” may way further helped to count distance more logically that also reduced the excessive distance in kilometers counted by Google Map data.
- At this point, the algorithmic approach has been established in such a way that if multiple different locations are put, it can calculate the shortest possible journey among the distances from the source point and get back to the source point following quickest possible way.

This is not only an emissions study, but also about rendering emissions as something intelligible, traceable and visual, so that any logistics manager, any researcher, or indeed auditor can follow the journey in the same way they would through a map.

1.3 Research Problem

In this study, it is highly observed with several analysis through internet that the distance data of logistics operations is many times overlooked and sometimes it becomes impossible to determine the CO_2e rate of a specific organization. The available data on vehicles, which are mostly contained in financial accounts, do not provide details of the routes that are taken. Such an inconsistency impedes routing that is environmentally effective and precise reporting of impact.

Two types of emissions are generated in the case of road transport under refrigeration: one from the automobile, due to vehicle mobilization and the other from the cooling unit that keeps a constant temperature in the cargo. However, most published methods for calculating emissions do not include refrigeration, are based on simplistic assumptions or do not comply with global standards such as ISO 14083, and the GLEC Framework. Additionally, the emission shares also differ by route segment because the operating time, speed profiles, load and unload activities, behavior of TRU fuel consumption and refrigerant leakage are different. This results in a serious gap: No company has developed a standardized, per route data supported approach to the figuration of well-to-wheel (WTW) activities of temperature-controlled transport chains.

Thus, the main question of this thesis is:

How accurately CO_2e emissions can be calculated and route-level might be fairly allocated for a multi-stop refrigerated road transport operation by combining primary diesel fuel consumption data with activity-based indicators such as tonne-kilometres, notional activity, share of capacity and shortest feasible distance (SFD) for each delivery?

1.4 Research Objectives

This study aims to:

1. Develop the multimodal dashboard that identifies the best path of the truck based on real-world road information to assist with logistics planning.
2. Add emission factors to the dashboard to be able to approximate the CO_2 by the optimized routes.
3. Compare the calculation (on the dashboard) to the method using the traditional techniques.
4. Allow logistics companies to employ a dashboard to reduce emissions.
5. Identify transportation emission will be enumerated with the guidelines published by ISO14083/GLEC which is properly followed and intended to minimize the real distance using algorithms to determine the shortest possible path.
6. Dedicatedly indicate the comparison among the real calculations and shortest distance identified using Google Map's mainstream analysis through API.

1.5 Structure of the Thesis

This thesis is organized as follows:

Chapter 1: Introduction - provides an outline of the background, problem, objectives, and structure.

Chapter 2: Literature Review - describes modern studies in the field of calculating emissions, routing, and an interactive dashboard.

Chapter 3: Methodology - describes research design, data collection, and analysis.

Chapter 4: Results - explains the building of the interactive dashboard.

Chapter 5: Conclusions - outlines results, standards, and compares with previous methodology, and discusses implications.

2 Literature Review

Standards foundation: ISO 14083 and GLEC V3

Studying core system boundaries, calculation flow, along with the concept of transportation chain elements (TCEs), all depicted in the ISO 14083, has enriched the knowledge on the segments that the freight is handled by a single vehicle or through a hub; TCEs are built against the summation of the emissions as a whole (Olivari et al., 2025). A common rule is given by ISO 14083 that how the transport emission can be calculated such as distance, weight, time and energy (International Organization for Standardization, 2023).Freight having temperature sensitivity are a must handle with specific intensities instead of ambient defaults. GLEC v3 says it is in line with ISO 14083. In this standard foundation, we found emissions have the link to real transport activity. The emission from vehicle operations, production of energy as well as the operation of different hubs must be clear and transparent(International Organization for Standardization, 2023 Clause 5.2). It sets the scene for the Well-to-Wheel (WTW) method, the Well-to-Tank (WTT) and Tank-to-Wheel (TTW) method (GLEC, 2025, pp. 19–21). It presents refrigerant factor tables with worked allocation examples (GLEC, 2025, pp. 113–115). Thus, it serves as the “industry-accessible” level for implementing ISO’s requirements in tools and dashboards. However, where GLEC V3 supports real-life implementation, ISO 14083 indicates the scientific structure.

2.1 Why should freight cooling have a treatment of its own?

The Road transport temperature-controlled is not the same as the ambient freight. The ISO 14083 guidance considers temperature-controlled operations as a special situation and implies that the special intensities (not generic ambient factors) should be used when determining the GHG emissions in the transport chain components (TCEs) raise question. This division is evident in the computing division regarding the standard and the sections that address terminal/hub energy in temperature-controlled flows, which, combined, drive the modeling of the refrigeration unit (TRU) as a specific energy consumer instead of integrating it into generic car variables. Emission does not come from

this movement only, hence it has real time relation with auxiliary system of the refrigeration units from the specific refrigerant leakage(International Organization for Standardization, 2023 Clause 5.2, Annex I). However, even also the cooling uses the energy if the vehicle is standing in some place without a bit movement and the energy required here has dependencies more on time and counting of stops way further than distance. For this, it is observed that the propulsion emissions have differential behavior from the cooling emissions. Energy usage related to transport operations has to be included as per the recommendation of ISO 14083. Refrigerant leakage is a separate emission source as well as the freight cooling which is modeled to avoid understanding total emissions in cold-chain transport. This framing is in line with the industry playbook exemplified by the GLEC Framework, which identifies itself as the operational methodology layer that is in line with the ISO. The chapters of the methodologies used by GLEC continuously emphasize the necessity to choose the correct intensity and factor in the operation uprise the question and to record the options that influence comparability (e.g., Well-to-Wheel vs. Tank-to-Wheel).

2.2 Refrigerated-Transport Emission

Refrigerated land transport is considered as one of the most energy consuming processes in the cold chain. Maiorino et al. demonstrate that vapor compression refrigeration (VCR) units are the primary source of refrigerant leakage and can account for up to 40% of a vehicle's engine-related GHG emissions when direct refrigerant leaks and indirect fuel use are included. Inside the system boundary, the total emission related to transport are all relevant according to ISO 14083. The three main parts that is aligned with the guideline is depicted as(International Organization for Standardization, 2023 (Clause 5.2)):

- Fuel used to move the vehicle
- Electricity or fuel used by the colling machine
- Refrigerant leakage emissions

The standard has requirement of the summation of these parts all together to form total emissions(International Organization for Standardization, 2023 (Clause 12)). The review also mentions that the transport of refrigerated goods may be responsible for approximately 15% of global fossil-fuel energy use, underscoring its outsized role in contributing to climate impact compared with stages elsewhere in the supply chain (Maiorino et al., 2021). The cold-chain/refrigerated-transport model, in which it is considered that the transport leg is regarded as a separate subsystem having high impact for which fuel consumption and emissions shall be quantified accurately.

By measuring Consumption/100 km, Consumption car × km, and CO2e (kg) on a real distribution route, and which such figures can be dismantled on a customer level has supported the notion of refrigerated transport being an identifiable part of product climate impact rather than only being “overhead” (Maiorino et al., 2021).

2.2.1 Energy Expenditure, Weight loads and Work of Transport (TKM)

The review highlights that fuel consumption in refrigeration transport depends not only on driven kilometers, but also on the weight loads (wall transmission, air infiltration during door openings and product respiration heat), the COP of the refrigeration unit and the weight of goods loaded. Here ISO clearly indicates the transport activity as a parameter which can measure the way of service clearly is expressed as tonne-kilometers (TKM), it means the transported weight is naturally multiplied by distance to follow the standards(International Organization for Standardization, 2023 (Clause 10.2 and 3.1.27)). TKM means the physical “work” done by the transporting vehicle where emissions related to freight is quietly related as per how heavy the load refers and how far the loads are transported.

To make the connection between refrigeration behavior and services work, several vehicle features (K-value and SA), driving cycles that combine load/distance/operation profile are detailed which should be assumed as input for energy demand & emissions estimation.

This supports a second theory that is very similar to my calculations:

The energy/transport-work model, which links fuel use and emissions not only with distance (today's measure of work), but also with tonne-kilometers (TKM) and other measures of "activity.

The ideas are represented as the following exactly:

TKM = Current Weight × SFD / 1000 is the direct measure of transport work for individual legs (Maiorino et al., 2021).

Notional Activity may be applied in the same manner, but "activity" is ascribed to unloaded fraction for each customer rather than using "customer activity" index for the overall ship operation from one port of call to another. Fairness rule that might apply when several customers are involved on single trips (Maiorino et al., 2021).

The Share of Capacity and the Share of Consumption signify how much of the truck's technical capacity and fuel load each drop is contributing to.

Overall, with my model the rather vague idea of "transport work and thermal load" from literature is turned into a binding distribution for a concrete vehicle and route. Rather than attempting to model COP or the specifics of heat transfer you simply postulate that fuel use is proportional to **activity (distance * weight)** and divide measured consumption accordingly. This reflects the way tonne-km metrics are used in many of the studies, although these generally report at fleet or scenario level and not per-customer (Davydenko et al., 2022).

2.2.2 Emission Reduction and Optimal Strategy in Refrigerated Transport

One large section of the article is about optimization: reducing loads, making sure the system runs as efficiently as possible, and planning routes that require less energy and produce less emissions. The authors discuss:

Updated insulation and wall structure such as vacuum panels and PCM modified foams to reduce transmission loads. Air infiltration decreases through air curtains and improved door control, as distracting door openings add around 30% to the refrigeration load (Maiorino et al., 2021). Alternative heat sources (PV, fuel cells, LNG cold-energy, sorption systems, air cycles, CO_2 ejector cycles) to increase COP or uncouple cooling from the prime mover (FABRIS F et al., n.d.). The application of VRP models adapted to refrigerated distribution to minimize distances travelled and the opening of doors for loading/unloading, while respecting time–temperature constraints. If we look into the ISO 14083 standards which links the emission information that better routes can decline the travel time and better vehicle utilization can reduce propulsion also.

These provide a third theoretical framework:

"The optimization / sustainable-routing framework; endeavoring to minimize the aggregate fuel consumption and emissions via advancements in vehicle technology as well as improvements in the way a vehicle is operated and routed."

The Excel analysis fits into this framework by offering the measurement level to decide on the effectiveness of optimization. Because the calculation stands on:

The methodological approach by adding a quantitative measurement layer to assess the quality of route and load optimizations. By itself, it does not serve as a stand-alone calculation tool, but rather the excel file is employed to operationalize activity-based data from the routing dashboard and calculate fuel use and emission signs on a leg basis. The basis of the computations is the total amount of recorded fuel consumption related to a transport cycle completed at the latest, before which downstream allocation between operational responsibility is performed. This decomposition is done by assigning energy consumption and emissions to individual transport legs, which can be quantified in terms of measurable activity indicators like driven distance, time on the road, and vehicle load.

Generating the type of quantitative performance measure that optimization models require as a target or loss function. For example:

If someone tries a different routing or consolidation method, TKM and Notional-Activity based approach can indicate what the impact of such changes would be on total CO_2e and CO_2e per delivered kilogram (Davydenko et al., 2022).

The structure of spreadsheet can be reused to demonstrate against such changes whether Consumption/100 km and CO_2e per tonne-km fall.

2.3 Activity-Based Emissions / Tonne-Kilometre

The concept of activity-based emissions is also used by me to calculate and allocate emissions in this paper. In this model, freight emissions are calculated as a function of the real "transport work" performed (i.e., how much volume is moved and how far). The mass of cargo multiplied by the activity distance (in tonne-kilometers) is considered as freight transport activity, in line with ISO 14083 and the GLEC Framework. Instead of focusing on kilometers driven, the activity-based model ties greenhouse-gas emissions to tonne-kilometers – because transport work represents the key relationship for equitable and transparent allocation.

$$\boldsymbol{Emissions} = \boldsymbol{Actvty}(\boldsymbol{t-km}) \times \boldsymbol{Emssn\_intensity}(\boldsymbol{kg\ CO_2e/t-km}) \quad (1)$$

(Schmidt et al., 2025a) also introduced activity-based calculation as one of the two "core" methodologies for freight emission calculation—complementing it with energy-based (fuel-based) calculation. (Du Plessis et al., 2023) extend the logic to domestic refrigerated road transport and estimate g CO_2e/t-km following an estimation of fuel use for 147 real-world trips where I found match because my calculation for depicting real road distance using Google Map also represents the value as accurate based on algorithm.

2.3.1 Defining activity distance and transport mass

Activity distance and size of shipments defined. The GLEC Framework follows this requirement and allows for SFD, GCD or actual vehicle kilometers to be used as acceptable measures. By calling them "SFD" for the legs, and "Current Weight" in kilograms, the logic works as the distances and weight are explicitly quantified according to this activity-based layer where shortest path is easily calculable. These two input components serve as the starting point for determining activity. ISO 14083 also specifies that the mass of freight is to be declared as shipped mass in kilograms or tonnes (International Organization for Standardization, 2023). The representation of transport mass means the freight mass being transported but not the vehicle weight.

In my model, the columns "Load (kg)", "Unload (kg)" and "Current_Weight (kg)" correspond to this definition as they represent how much mass is effectively being carried over each link of the route. These two fundamental inputs for the activity-based model, mass and distance, are of course known.

2.3.2 Using primary fuel consumption to calculate emissions per leg

ISO 14083 and Schmidt suggest using original fuel data if it is available as this would provide the most accurate results. When estimated output indicates the emissions, then the fuel consumption primary data is considered from the excel sheet. Converting kilometers into liters of fuel (using the truck's consumption rate per 100 kms) and then apply a generic emission factor to calculate how much CO_2e is being emitted each leg (Schmidt et al., 2025b) and (International Organization for Standardization, 2023 (Clause 5.3)). Having both activity and emissions (in tonne-kilometers and kilograms of CO_2e) means it can calculate the implicit emission intensity for that leg in CO_2e per tonne-kilometers. This is like what happens in the activity-based framework, but at a more disaggregated level as it works for each leg of an activity rather than for the whole trip.

According to (Schmidt et al., 2025a) necessary equations are followed as

$$\boldsymbol{Enrgy - based\_emssn = fuel\ consumed \times emission\_factor} \quad (2)$$

And therefore another one precise formula indicates as :

$$\boldsymbol{Emission\ intensity} = \boldsymbol{total\ emissions} \div \boldsymbol{transport\ activity} \quad (3)$$

Emission intensity means how much greenhouse gas is generated for each unit of transport-work. It is calculated by dividing the **total emissions** from a trip or vehicle segment by the **total transport activity**, which is usuallly measured in tonne-kilometeres (t-km).

The excel model mirrors this sequence:

Calculation 1: $\text{Consumption}/100\ \text{km} \times \text{SFD} \rightarrow \text{Consumption car} \times \text{km}$

Calculation 2: $\text{CO}_2\text{e}\ (\text{kg}) = \text{fuel volume} \times \text{diesel emission factor}$

Because CO_2e (kg) and TKM can be derived to imply kg CO_2e per t-km, fully compatible with ISO and GLEC.

2.3.3 Allocating transport activity and emissions to individual consignments

The ISO 14083 makes a claim that under these standard emissions can be calculable on the transport-operation level as well as down to the consignment level, however it must be possible to allocate emissions across individual shipment (International Organization for Standardization, 2023). GLEC has the same guidance: calculate t-km per shipment and use it to assign emissions.

The following principle is used during calculation:

$$\boldsymbol{Notional\ Activity} = (\boldsymbol{Unload\_kg} \times \boldsymbol{SFD}) \,/\, \mathbf{1000} \quad (4)$$

Finally, the emitted quantities are assigned for individual customers by prolonging this tonne- kilometric reasoning to the level of customer. I then multiply the unloaded mass at each stop by this distance, resulting in "Notional Activity" value which reflects the

specific activity of a shipment. This allows me to assign emissions according to the transport work of each customer's delivery. Further allocation factors, capacity and consumption share, are applied if mass-based activity does not fully represent resource sharing used on the route (Maiorino et al., 2021). Primary fuel data is used along with leg-level tonne-kilometers and consignment level activity to put to work the theoretical foundations of the framework in response to the complicated nature of refrigerated multi-drop deliveries. This helps to calculate emissions in each TCE reliably.

2.4 Data-Quality, System-Boundary and Optimization / Routing

Learning throughout the topics in (International Organization for Standardization, 2023), (Schmidt et al., 2025a), (Maiorino et al., 2021) and (FABRIS F et al., n.d.) demonstrated that, refrigerated road-transport emissions are always discussed in a macro-level. The general premise of these studies is that it is appropriate to meaningfully aggregate emission performance over either the full driving route, the complete fleet or even a whole category of refrigeration technology. Therefore, theoretical concepts of tonne- kilometers, TRU energy intensity, system boundaries and data-quality tiers are explained in depth, but the everyday reality of where these emissions are sourced on an individual multi-stop refrigerated delivery route is all but untouched. This results in a practical methodological gap from norms to what industry can compete.

2.4.1 Absence of leg-level activity measurement based on real operational fuel data

The one of the main gaps is the "source of data" used for emissions modelling. Current studies often use standard driving cycles, default emission factors or average power evaluations. In most studies fuel consumption is either modelled by regression equations or it was taken a fixed percent increment over non-refrigerated operation. This is to say that dynamics at the level of a route varying length between stops, weight changing as deliveries get offloaded, or different driver speeds and cooling requirements are being largely abstracted out.

No study reviewed connected actual, measured fuel consumption to individual leg-level transport activity. They report fuel consumption for a trip or test cycle, but no per segment is modeled in a multi-stop route. Therefore, they are unable to measure what portion of the total fuel burn is caused by each individual leg, and so cannot determine which customers or flight legs are disproportionately contributing to emissions.

In this thesis the model designed prescribing with the application, the software takes care of this by using the vehicle's actual final reaching towards destination, the exact SFD distance for each segment and diminishing payload over route. When estimating CO_2e per leg, and when applying that to leg-level TKM, it is tried to make a precise connection between activity-based emissions and the highest possible level of data quality that followed the guidelines of (International Organization for Standardization, 2023). This method accurately models the real operation of multi-stop refrigerated transport and closes the most significant methodological knowledge gap identified in the literature.

The algorithm receives a source node (depot) and a set of candidate destination nodes (stoppers), together with their load-unload amounts. As a preliminary processing step, all destinations are consolidated which refer to the same physical location using either position coordinates or location name and loads / unloads aggregated. In addition, If any of those stops fall on the depot/origin itself, I processed them as well in a "return payload" to make sure that the algorithm works with clear list of distinct stops and logs simply all times a load/unload happened short before or after departure (Sar & Ghadimi, 2023). This is in line with the real-world operation where multiple customer orders may be delivered to the same address.

Then a complete distance/duration is calculated with a followed structured way that connects all these points via reading Google's Distance Matrix API: The matrix gives the driving distance and driving time, according to real-world conditions, between each pair of nodes. These values are then utilized as leg distance (km) and leg time (min), which

constitute critical inputs for the two activity-based (tkm) and fuel-based emissions calculations.

A scheduler which is a greedy priority based is at the heart of the algorithm. At each stage, it selects the 'next" stop from those yet unassigned according to three rules:

Priority order:

Case 1: First, prefer stops where only **load** in cargo operation executes (LOAD),

Case 2: Then stops in the zone where there is weight to **load and unload** (BOTH),

Case 3: Only then stop where there is weight purely to **unload** (UNLOAD).

Feasibility constraint: A stop is only considered if the resulting onboard weight never becomes negative. At BOTH and UNLOAD halts, it verifies whether the amount of onboard weight complies with the required volume to be unloaded.

Distance criterion: Among all feasible destinations according to the distance countdown in the highest-priority class, the algorithm selects the nearest stop (Li, 2022).

2.4.2 Identifying operational link to route Construction and Leg-Level Emission Allocation

In the cold-chain review article, the dominant operational connection between routing and performance continues to be distance or cost, and not detailed emissions. (Qiang et al., 2020) creates a capacitated VRP with refrigerated distribution per customer sequence whereby generating all possible permutations of sequences and using a greedy search algorithm to choose the best remaining option for each node based on cost and distance.

Their objective function is defined in monetary term and as a multiplying factor of kilometers stated - the optimization engine finds only kilometers (and capacities), refrigeration energy, TRU fuel use or CO_2 emissions never appear among decision variables or constraints (Sar & Ghadimi, 2023).

Once the greedy algorithm picked three best paths, it is easy for the authors to put those total route distances up into a transport-cost function to immediately have an unique and optimal cost value (Qiang et al., 2020).
Instead of treating routing as a continuously adaptive process driven by a cost metric, the route is determined in advance at a broader, macroscopic level. This route selection is based on predefined operational or “metabolic” criteria that define what is considered optimal before the journey begins. Once the route is fixed, the total distance travelled is calculated, and the associated cost is then derived afterwards using simple, standard calculation methods. In this way, cost is evaluated as a consequence of the chosen route rather than being used to dynamically adjust the route during execution. Specifically, there’s no mechanical link from route design to fuel usage or emissions (Dobers et al., 2025). There we concluded that while environmental and social goals are increasingly being incorporated into routing formulations, this is usually at a high level of aggregation (Sar & Ghadimi, 2023). This describe a large family of “green” VRP and pollution-routing models in which fuel consumption or CO_2 emissions are expressed as analytic functions of distance, vehicle load, speed and type of road using sometimes tools such the Comprehensive Modal Emissions Model.

Here carbon emissions come out as one term in a multi-objective function including cost and social criteria, and the objective of optimization is to find routes with minimum cumulative emissions over the entire planning horizon. The review, however, highlights that such work remains in early stage and even ‘green’ VRPs largely rely on generic emission factors and model-based equations as opposed to high-resolution operational fuel data or shipment level indicators Parameters: “black” Research Gap (Sar & Ghadimi, 2023). There is a lack of understanding regarding the environmental performance characteristics of green Vehicle Routing Problems (VRPs) (Yin et al., 2024).

None of the models they review contains an explicit, converged methodology for converting actual TRU fuel consumption and route telemetry data to CO_2e per leg and per customer. To construct a feasible route from a depot to several cold-chain customers and return, I focused on a constructive greedy algorithm.

The function solve_phased_route begins at the depot, merges all identical locations together, keeps track of capacity with a running “on-board” weight and at each step steers towards the closest available next stop that is possible to service, preferring loading then combined load/unload services then pure unloading until all customers have been served once exactly and vehicle returns to source. This results in a specific sequence of legs, which can be readily compared with the greedy VRP routes of the cold-chain paper, but it is explicitly ensured to evolve with the payload and not allow negative load age, something necessary when calculating later emission allocation.

The literature review has shown that current research in transport optimization and cold chain logistics is more focused on integrated analysis of the routing efficiency, vehicle loading dynamics and auxiliary power consumption than an isolated one. Although there is valuable research that discusses the vehicle routing problems, temperature-controlled transport, emission accounting frameworks such as GLEC and ISO 14083, they have not been translated into practical tools to aid near- or real-time decisions at the company level. The literature shows that both time- and distance-based energy requirements must be considered for accurate truck behavior, but few operational tools combine these with dynamic routing.

Leveraging these findings, the method section that follows presents a system that puts reviewed concepts via a routing interface driven by dashboards and an accordingly Excel-based calculation framework. By this method the theoretical construction referred to in the literature will be operationalized and tested, quantified and assessed within their world of practice. The paper presents a proposal for a decision-support system that integrates the route optimization, monitoring of vehicle loading and emission estimation in temperature- controlled transport. Special attention is paid to how routing data, operating parameters and refrigeration unit behavior are combined in a dashboard. The accompanying Excel-based calculation tool, including its role in measurement of fuel consumption and CO_2e emissions on the trip and leg level is also discussed. Combined, these topics illustrate how theoretical guidelines and academic theories are used in a practical logistics environment.

3 Method

This method is designed to be used as an operational route-level emission 'accounting' tool for refrigerated road transport. It is intended to take real operational input (route geometry, per-stop loads/unloads, measured distances/times) and produce auditable, leg level activity/ emissions output suitable for reporting or optimization. Attention was given to transparency and traceability: every intermediate step (current onboard weight, leg distance, leg duration, shipment-level activity) is logged so that there is a clean audit trail from raw data to per-customer CO_2e. It supports refrigeration-specific functions like kilometric wise consumption estimation; allocation of refrigerant leakage should be counted or not and class-based operating modes and provides results to export for visualization on dashboards or for use as an objective signal in routing experiments. Finally, this method will be 'deployed' to support rapid scenario testing and sensitivity analysis, allowing comparisons of alternate routes, bundling options and TRU configurations with full lineage back to the original operational data.

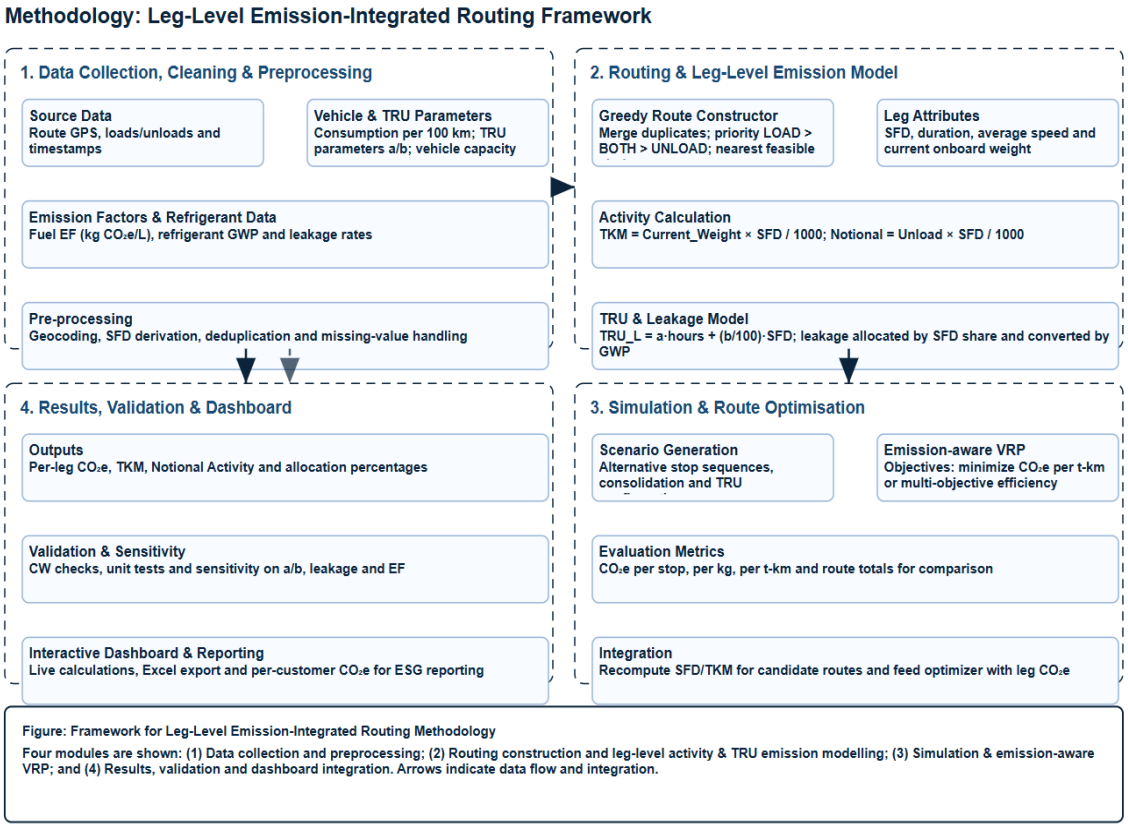

Figure 1: An illustration of Methodological Approach

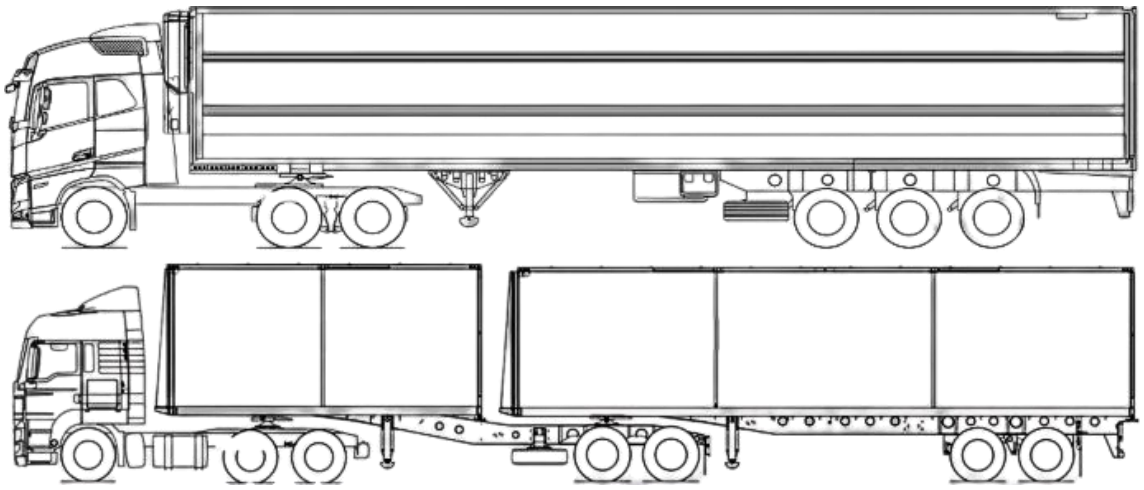

Figure 2: A tri-axle reefer semi-trailer (top) and a tandem-tandem interlink tautliner trailer (bottom) collected from (Sar & Ghadimi, 2023)

A trip case: How does it look based on my analytical case?

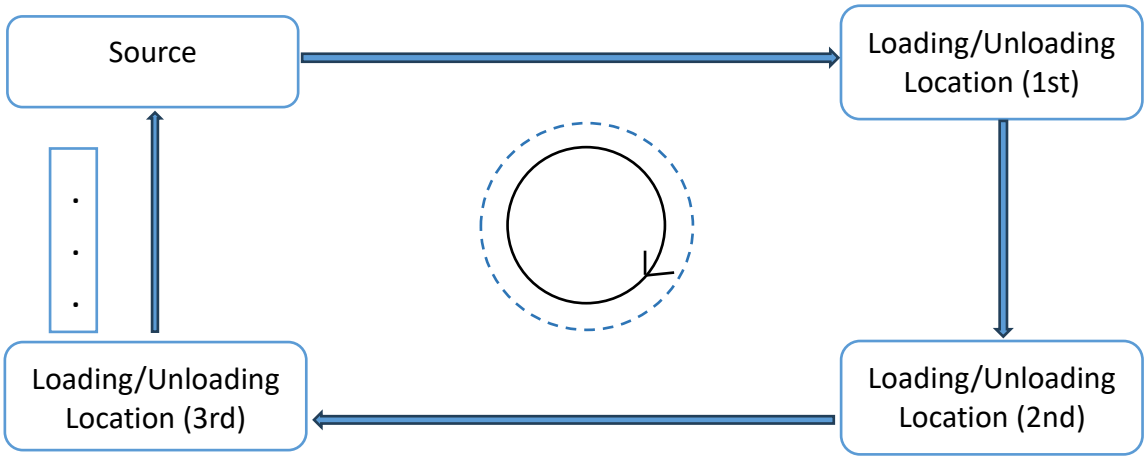

Figure 3: A single journey trip

The trip starts from the Source address, taking First_Load to reach it at the first location. The First Location has unloading value as well as the loaded peripherals. The trip continues to next location to unload values and take new values. Some places do not have loading value, where that position is only supposed to unload the value carefully. From the last location, the journey turns to the Sources again. Inside this journey, our prime target is to identify the TCE's emission per TCE.

3.1 Data Acquisition and Preprocessing

The data has been shared by an anonymous company who are intending to implement it's refrigerated road haulage services accurately. For confidentiality and data protection compliance reasons, all company and vehicle information, customer details and geographic locations have been anonymized. The names place had been obscured but no commercially sensitive information is revealed. The company voluntarily supplied the data, in belief that the findings would provide a better understanding of GHG calculation methods from cold chain transport viewpoint for academics.

The company was mainly motivated in sharing data to investigate improved and transparent alternative ways of estimating transportation related emissions that go beyond simple distance-based considerations. In this sense, the company was interested to know what impact (in terms of total well-to-wheel (WTW) emissions) do the actual operation features have such as: route segmentation, loading and unloading events, vehicle utilization or refrigeration energy use. We do not aim that our own software solution should be developed, but to analyze if standardized approach (as specified in ISO 14083 and the GLEC Framework) can be implemented practically by means of actual operation data.

Python is chosen as the main programming language for data analysis and preparation as it is flexible, transparent and particularly capable of dealing with raw operational data. The initial data provided by the company required analysis using SQL to connect with Excel to upload to the SQL database to pull accurately and push data to store permanently for future reliable queries. To pursue proper visualization by reading data from the stored dataset, Python libraries were employed to pre-process the data by cleaning, validating and structuring its format (e.g. timestamps, distances, route information) so that distances data contain no tracing errors or ubiquity and time stamps display no outliers of interest for further analysis. In addition, external driving information (e.g., driving distance and travel time) was programmatically queried through APIs so that route characterization could be done automatically and with reproducibility.

3.2 Decision of Implementating Greedy Algorithm

The greedy algorithm was employed due to the pragmatic tradeoff it offers with respect to operational realism, computational speed and methodological transparency. By design it yields deterministic, feasible multi-stop tours that adhere to loading/unloading restrictions and never violate non-negativity of onboard weight, producing phase-legs that are immediately suitable for leg-level TKM and TRU emission accounting; this ensures a transparent audit trail from route geometry and payload development through to per-customer CO_2e. While not aiming for a provably global optimum, its rule-based prioritization (LOAD → BOTH → UNLOAD), nearest-feasible unloads selection, and straightforward treatment of infeasible unloads all mirror real-world operational decision logic to produce routes which are credible for practitioners. The algorithm is computationally inexpensive and repeatable which allows for the rapid generation of scenarios, sensitivity analysis and producing other comparable candidate routes to be evaluated by an emissions-aware optimizer; it behaves deterministically with respect to a particular emission cost function where all important practical benefits in support of a thesis about bridging emissions measurement and routing practice.

3.2.1 Addressing Phase-Leg using this algorithm

Let the depot (source) be numbered 0. Let S= {1,.....,N} be the set of unique delivery stops after merging duplicates. For every stop i ∈ S, the following input fields are considered with precise assumption as:

Load: $load\_i$(kg) column *Load (kg)*.
Unload: $unload\_i$(kg) column *Unload (kg)*.
SFD: SFD_i(km) *SFD* for the leg from origin to destination.
Distance matrix: $\mathrm{dist}(a, b)(\mathrm{km})$shortest-feasible distances are precomputed between nodes which is derived from Google Distance Matrix for polyline calculation.
Current weight: CuW(kg) *Current_Weight* recorded before leg t.
An order phase-leg is constructed with the following equations:

$$\boldsymbol{M} = \{(\boldsymbol{tag\_t}, p\_\boldsymbol{t}, q\_\boldsymbol{t}) \backslash\}\_(\boldsymbol{t} = \boldsymbol{1})\hat{}\boldsymbol{S} \quad (5)$$

Which is established by the greedy route constructor where each tuple is stating leg type (tag), origin p_t and destination q_t. The next 5 positions contain notation, exact formulas and comments for the calculations used in the Excel/workflow.

Define phase-leg is challenging but it has been tackled with the creation of the transport activity which means useful to develop greedy routing considering all incremental manner by optimizing local decisions. To convey the above context, the idea of Phase comes with defining whether the movement is either loaded or unloaded, hence Leg is a term that puts the feasibility analysis with best comparison etiquette. These parts are taken as reference to this thesis to develop a extrinsic route where all the weights will be considered using proper implementation of algorithm. The effort of using greedy algorithm fixes the complexity by electing the nearest node locally which also helps PLR algorithm to maintain the features of the routing stability. In this format the major facilities are the distance and time both are preserved because the calculation process is quite solid to determine the distance of each leg where it is not important to assume on the average distances. Then comes the matter of determining the activity-based emission counting which is a distribution process that quite relates to the original movement of the Truck and relates to the operating systems. This process ensures the division of the chain into small identity that escalates emission research reporting tool way better than ever. In addition, it can be said that regarding greedy logic, it is way far better online and data related process that relates the research of the routes with detecting the faster and calculated prospects. In summary, the phase-leg data found with the determination of the Greedy algorithm is better presentation model to calculate easy operational ways of calculating emission phase data that relates the conjugation of activity with other formations.

In the research paper, ***" A systematic literature review of the vehicle routing problem in reverse logistics operations*. Computers & Industrial Engineering"** (Sar & Ghadimi,

2023)**,** here the phase is determined with the help of the routing based algorithm which is in reality differs the ideology of the planning with leg based data calculation. In logical nodes which are connected to each other where there is one in loading another one in unloading is the phase leg. The latitude and longitude both data is used basically to determine the actual distance where instead of the steady straight line it is whether connects reality routes that affects the emission. Thus, ensures clarity in nature with usage of the fuel and calculation of emission and it supports the countability of the following attributes. In this context, if the leg is considered as unit to determine uniquely then it helps to consider the load values, distance optimizations and usage of vehicles in routes is densely differed with proper presentation. The distribution data of the emission is since carrying complexity while the load is in the middle of the routes and there are more places to drop and the emission calculation becomes tough to identify while waiting. The way these rules are quite related to ensuring the calculation of the emission process to develop a sustainable research method to match everything fruitfully. In above all, this is a very crucial process where the method of phase-leg determination has become the base of the proper segmentation of emissions data in a solid structure.

3.2.2 Priority class determination to address LOAD, UNLOAD and BOTH

Onboard load is determined based on cumulative load-unload balance, such that it was always possible to attribute with certainty a specific mass carried at a given row with respect to which operations were recorded. This was realized as the cumulative sum of both Load values up to this row minus the cumulative sum of all Unload factors up to this row and then clamped to zero not to allow negative values. Where the Load /Unload column was blank this reflected to zero and the running total was such that each row represented a correct auditable vehicle payload at that stage of the tour.

$$\boldsymbol{priority}\,(\boldsymbol{i}) = \begin{cases} \boldsymbol{LOAD}, & \boldsymbol{if\ load} > \mathbf{0} \ \wedge\ \boldsymbol{unload} = \mathbf{0}, \\ \boldsymbol{BOTH}, & \boldsymbol{if\ load} > \mathbf{0} \ \wedge\ \boldsymbol{unload} > \mathbf{0}, \\ \boldsymbol{UNLOAD}, & \boldsymbol{if\ load} = \mathbf{0} \ \wedge\ \boldsymbol{unload} > \mathbf{0}, \end{cases} \tag{6}$$

This aggregate definition ensures mass is conserved for the tour, simplifies subsequent allocation and tonne–kilometer calculations, and facilitates easy validation (e.g. by verifying that Current_Weight (CW_n)becomes zero when all cargo has either been delivered or returned).

$$\mathbf{CuW}_{\boldsymbol{n}} = \mathbf{max}\left(\mathbf{0}, \sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{n}} \mathbf{Load}_{\boldsymbol{i}} - \sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{n}} \mathbf{Unload}_{\boldsymbol{i}}\right) \quad (7)$$

In the excel file, the The Current_Weight may be computed as the onboard_weight plus the load and minus the unload per row; missing empty load/unload values were treated as zeros and clamped at a minimum to be 0 so that negative payloads would not result.

To identify the Load State, Distance and Stop priority, we got the information from various research to focus on the operation path extraction by considering the iterative explicit movement with structured sequence (Davydenko et al., 2022). In this thesis, we need to be consistent with ISO 14083 and the GLEC framework to determine operational path, transport activities and individual “Transport Chain Element” calculation all goes to secure the performance of enabling emissions based on aggregated routes. LOAD, UNLOAD and BOTH are terms that safely indicate the state of carrying method between the legs where in some places the Truck has BOTH weight and need to UNLOAD in some places and also chances to LOAD in some places. So, the iteration of this journey continues maintaining the category. Hence, the direct usage of the fuel faces the impact on using the energy consumption as the refrigeration demand also the usage of the fuel along with the movement of Truck cost fuel.

As an example, the initial phase may have weight and Truck is moving with that weight as it is known there are place to drop some loads, so the changes of the loads in multiple place has direct relation to optimize logistic situation and but it does not indicate that the shortest path identification is the only solution. According to (Schmidt et al., 2025b), the model stands to determine activity based emissions makes the process easy on the environment and transportation activity also because the legs inside every shipment carries the resolution to ensure the clarity of the data that enhances the transparency. I

relate the calculation for LOAD, UNLOAD and BOTH while studying ***"Calculating fuel usage and emissions for refrigerated road transport using real-world data",*** (Du Plessis et al., 2023), where it is explained clearly that in between the trip level, it is possible to detect the amount of emission by guess or at accurate. From this research, it comes as reflection with the clear notation that the vehicle must move with partial LOAD or take LOAD in different places or even scope to UNLOAD multiple times. Even the case of BOTH is tackled very technically matched with the idea received from this research paper that is included and discussed with formula. In future, there is possibilities to implement the heuristic algorithm which can tackle the LOAD issue based on the capacity but must focus on the distance parameter also.

3.2.3 Identifying nearest feasible destination

At each construction step the residue set of stops RRR was split into three ordered candidate classes originating from fields Load (kg), Unload (kg) and current onboard mass Current_Weight:

L= {j ∈ R | $Load_j > 0$, $Unload_j=0$},

B= {j ∈ R | $Load_j > 0$, Unload j> 0, CW + $Load_j \geq Unload_j$} ("BOTH" feasible), and

U= {j ∈R | $Load_j = 0$, Unloading⇏⇍CW≥ Unload=j} ("UNLOAD" feasible).

The highest-priority non-empty class among $\mathcal{L}, \mathcal{B}, \mathcal{U}$ is selected as the candidate set C, and the next stop $p$was chosen by the nearest-neighbour rule

$$\boldsymbol{p = arg\ min}_{j \in C}\ \boldsymbol{dist}(\boldsymbol{curr}, \boldsymbol{j}) \tag{8}$$

using the precomputed distance matrix (the *Distance_km* / SFD values). Deterministic tie-breaking (by index or canonical name) is applied so that identical inputs always produced identical phase-legs; feasibility tests were evaluated against the logic of *Current_Weight* so that selection reflected on the true payload state before the visit.

3.2.4 Operational path determination for Iterative explicit movement

The onboard weight was recalculated iteratively in the spreadsheet, with an explicit capping of each unload to the physical maximum (the full wet load). On every row the actual_unload was thus given by the min of Requested Unload and Sum from previous Current_Weight plus this Load at that stop, and next Current_Weight computed as previous Current_Weight minus requested unload (with result clipped to zero) plus Load. It is cast in formula although no effect can be observed.

Define the actual unload performed at row *n* as

$$\boldsymbol{actual\_unload_n} = \boldsymbol{min}(\boldsymbol{Unload_n}, \boldsymbol{CuWn} - \mathbf{1} + \boldsymbol{Load_n}) \quad (9)$$

The blank load/unload cells were processed as zeros such that the iterative update was stable, and their capping behavior was done at route-constructor to ensure CW≥0 invariant. This future-proofs physical reality gives you a great audit trail in the Current_Weight column for relative inefficiency and will mean that downstream tonne-kilometre and "per customer how much do we use" calculations are based on consistent real-world state.

$$\boldsymbol{CuW_n} = \boldsymbol{CuW_{n-1}} + \boldsymbol{Load_n} - \boldsymbol{actual\_unload_n} = \boldsymbol{max}\,(\mathbf{0},\ \boldsymbol{CuW_{n-1}} + \boldsymbol{Load_n} - \boldsymbol{Unload_r} \quad (10)$$

These dispatches are pinged in the Current_Weight column of spreadsheet and set about by a valid running sum **= SUM (loads up to this row) - SUM (unloads up to this row)**. The capping condition maintains the invariant CW≥ 0 and hence discards infeasible negative payloads. The capping behavior is purposefully conservative, it ensure to visit each stop while preserving mass accounting in a physically meaningful way.

Once route sequence L is generated, we merge adjacent identical phase-legs to represent the repeated same phase-leg only once: specifically, two consecutive entries (tag_t, p_t, q_t) and (tag_{t+1}, p_{t+1}, q_{t+1}) are combined if (p_{t+1}, q_{t+1}) = (p_t, q_t). This processing removes redundancy and simplifies subsequent CO_2e calculations with no impact on the payload evolution.

In this thesis, the route is not defined as fixed and steady, ratherr multiple divisible Leg is connected to each portiion from last one to the newest one and each leg has its own very real time operational states. Here comes the method VRP (Vehicle Routing Problem) mentioned in (Davydenko et al., 2022) where the route is not considered as actual solution, hence it has been developed step by step with positional progress. In each step, the distance, time and usefulness is determined as for the condition of routing is activated in real time. For this, the identification of routing depends upon the real data but no on the asumption. The heuristic and metaheuristic appproach stands on the same methodologies where the step by step engagement in the route creates the transperancy, computational traceability, practical applicability by optimizing the total transport format with improvement of the iterative segments.

For detail learning, the journal ***"Does a detailed freight emissions calculation really matter?***", (Schmidt et al., 2025a) where it is even discussed the real route does not follow the linear distance, hence, the road network, the condition of the placement and the real time route determination has the significant role in the total operation. So that, the closest and useful distance at first is considered and sequentially the movement is determined from one node to another by calculating the shortest distance among the nodes. Distance between two nodes is said leg and it is at first analyzed so that the status of the loads, distance and operational differences is vividly visible to take further decision. In addition, the emission is, however, not calculated as an average, either connecting the movement in the reality. In that sense, the empty movement, partial load and the dependencies on multiple stops are well expressed in terms of gaining maximum output.

3.3 Computation of Shortest-Feasible Distance (SFD)

A local geocode cache is introduced to optimize address-to-coordinate resolution for the route and emission pipeline: responses should be fast, deterministic and auditable. In actual practice, each geocoding request is checked against the local cache first and only if there's no cached result we go to an external geocoding service; however, when an external lookup is made, the entire human-readable address, resolved lat-long pair and raw provider response are all stored so that subsequent runs can re-use them (which in itself would provide a rudimentary form of "historical" data), as well as allows us to inspect the original info when desired. This design is a significant improvement by limiting the amount of external API client call, latency and preventing problems due to rate-limiting and making geocoding an idempotent operation. The cache is equivalent to a computational shortcut and gives both computational efficiency and reproducibility: SFDs in addition to every per-leg emission metric become repeatable and auditable across runs even when external services have been down or given back slightly different answers.

If an address is in cache memory, the cached coordinates and formatted address are returned.

$$(\boldsymbol{lati}, \boldsymbol{long}, \boldsymbol{fmt}) = \begin{cases} \boldsymbol{Cache}(\boldsymbol{addrs}), \boldsymbol{if Cache}(\boldsymbol{addrs}) \neq \boldsymbol{\phi} \\ \boldsymbol{let}\ (\boldsymbol{lati}, \boldsymbol{long}, \boldsymbol{fmt}, \boldsymbol{raw}) = \boldsymbol{GeoAPI}(\boldsymbol{addrs}); \\ \boldsymbol{Store}(\boldsymbol{addr}, \boldsymbol{lati}, \boldsymbol{long}, \boldsymbol{fmt}, \boldsymbol{raw}); (\boldsymbol{lati}, \boldsymbol{long}, \boldsymbol{fmt}); \end{cases} \quad (11)$$

Failing to do so, an external geocoding request is made, full provider response stored and returned coordinates are taken from it for the above requests. This rule ensures idempotent address → coordinate mapping as identical addrs strings have only subsequent, equivalent (non-identity increasing) branches. Let Cache(addrs) be the cache lookup that results in a tuple (lati, long, fmt) or const NULL if the address is not in the cache. Let GeoAPI(addrs) be an external reverse geocoding API return (lati, long, fmt, raw).

$$\boldsymbol{Store}(\boldsymbol{addr}, \boldsymbol{lat}, \boldsymbol{lon}, \boldsymbol{fmt}, \boldsymbol{raw}) \Rightarrow \boldsymbol{Cache}(\boldsymbol{addr}) \leftarrow (\boldsymbol{lat}, \boldsymbol{lon}, \boldsymbol{fmt}, \boldsymbol{raw}) \quad (12)$$

There are two distance calculations based on existence of road geometry. If one has a road-distance RoadDistance(a,b) (in the Distance Matrix, or route polyline), the shortest-possible distance between nodes i and j is

$$SFDi,j=RoadDist(i,j) \tag{13}$$

Road miles are preferred as they account for realistic drivable distance. The fallback on Haversine makes the distance stable and deterministic, even when no external route geometry is applied. Since each SFD is computed based on cached positions, downstream dynamics and emissions are determined by the stored coordinates. Formally, for any leg *r* connecting the nodes *i* to *j* :

$$TKM_k = \frac{CW_k \; X \; SFD_{ij}}{1000} \quad , \tag{14}$$

$$Notional_r = SFD_{ij} \; X \; \frac{Unload_r}{1000} \tag{15}$$

The geocode cache policy was formalized as: return the cached coordinates, if they exist; else fetch the response from external geocoding API, store it and use the returned latitude and longitude. Leg distances were the road distance from the Distance Matrix where available, otherwise by the geodesic (Haversine) formula.

3.4 How “activity” is measured

Regarding each leg of the run as a TCE, it was tried to choose up hours, weights, and distance, so that it would be able to describe the activity in tkm per platform. In GLEC, it is quite clear that tkm is the common denominator, and it has been done by shipment/TCE. In this context, Shortest Feasible Distance (SFD) determination is recommended according to GLEC in most instances. The path determination, based on Google Maps, restrictions of vehicles, and actual roads, is taken cautiously.

Table 1: Addresses of Loading and Unloading Locations

Loading_Location	Unloading_location	Actual Distance in Km (manually determined)
AX (Source (S))	BY (Destination 1 (D1))	10.4
BY (Destination 1 (D1))	CZ (Destination 2 (D2))	191
CZ (Destination 2 (D2))	DX (Destination 3 (D3))	65.9
DX (Destination 3 (D3))	EY (Destination 4 (D4))	6.1
EY (Destination 4 (D4))	FZ (Destination 5 (D5))	2.9
FZ (Destination 5 (D5))	GX (Destination 6 (D6))	11.1
GX (Destination 6 (D6))	HX (Destination 7 (D7))	7.9
HX (Destination 7 (D7))	IY (Destination 8 (D8))	59.5
IY (Destination 8 (D8))	AX **(Source (S))**	197

Relevance: SFD will give similar results across days, times of the day, and drivers. It does not over-punish a late drop in a round trip, and its process can be repeated by any individual, provided the driver adheres to the same rules. In Table 1, the route name is given to specify the distances from the sources to the destination. As it is a loop consists with the information of multiple routes, it is easy to demonstrate calculation of other parameters based on each TCE individually.

3.4.1 Determination of longitude and latitude by Python Coding

In the Python code, I emphasize first fetching the Longitude and Latitude of the location by putting the location address in the terminal after running the program. The program is designed in such that the longitude and Latitude would come from the input until it

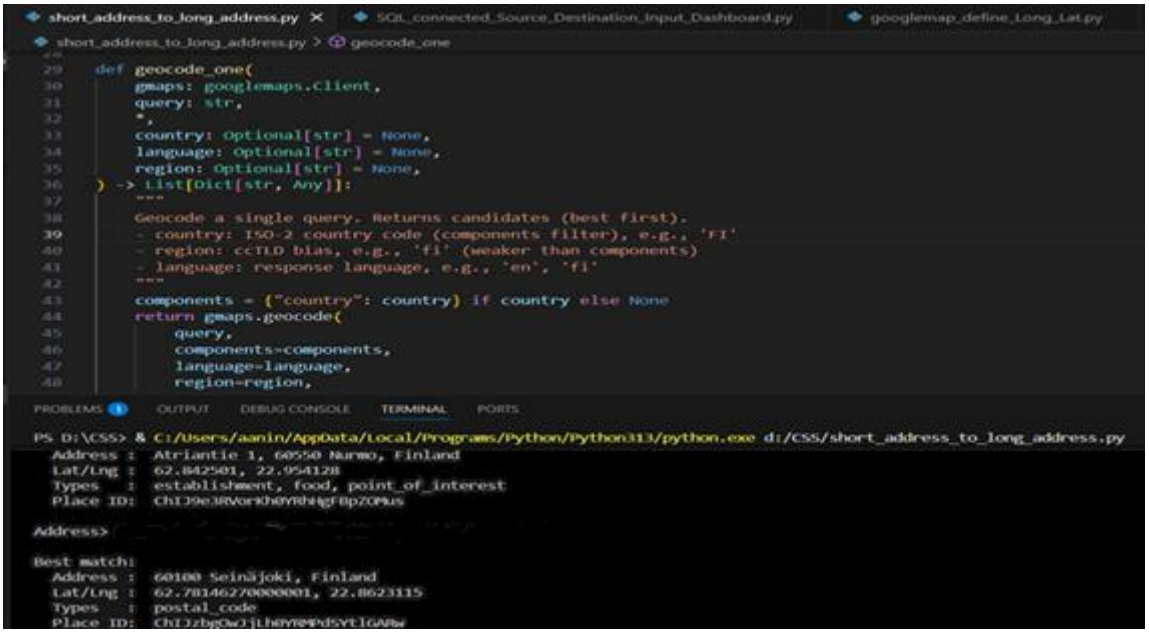

Figure 4: Python code to fetch the Longitude and Latitude of the location

stops. Here in Table 1, the Loading Location: "**AX**" for which the **lat_long** is derived as: 62.842501,22.954128, and the Unloading Location: "**BY**" for which the **lat_long** is derived as 62.78146270000001,22.8623115 using this programming code. In the same way, other longitudes and latitudes are also derived from the python coding. These steps are taken into highest consideration to ensure the deterministic distance between each source and destination much more precisely. In addition, here the best approach is observed to name every leg with D1, D2, D3…etc. In the coding, the distances are counted first to observe the difference between source and destination and measure the actual route value. The code follows Greedy Method to measure every distance individually and decide to follow the shortest path, hence the usages of names come with big impact to understand the actual route map. The time is also calculated along with the distance in minutes and hour.

Table 1: Addresses of Loading and Unloading Locations

From Loading Location's Lat_Long	To Unloading location's Lat_Long	Time (Minute)
62.842501,22.954128	62.78146270000001,22.8623115	12
62.78146270000001,22.8623115	61.8640918,25.1929862	154
61.8640918,25.1929862	62.2970797,25.8124431	55
62.2970797,25.8124431	62.2574189,25.7691865	10
62.2574189,25.7691865	62.2744337,25.7961759	7
62.2744337,25.7961759	62.2044625,25.7207254	14
62.2044625,25.7207254	62.2478984,25.7664288	12
62.2478984,25.7664288	61.8619823,25.1802074	46
61.8619823,25.1802074	62.842501,22.954128	150

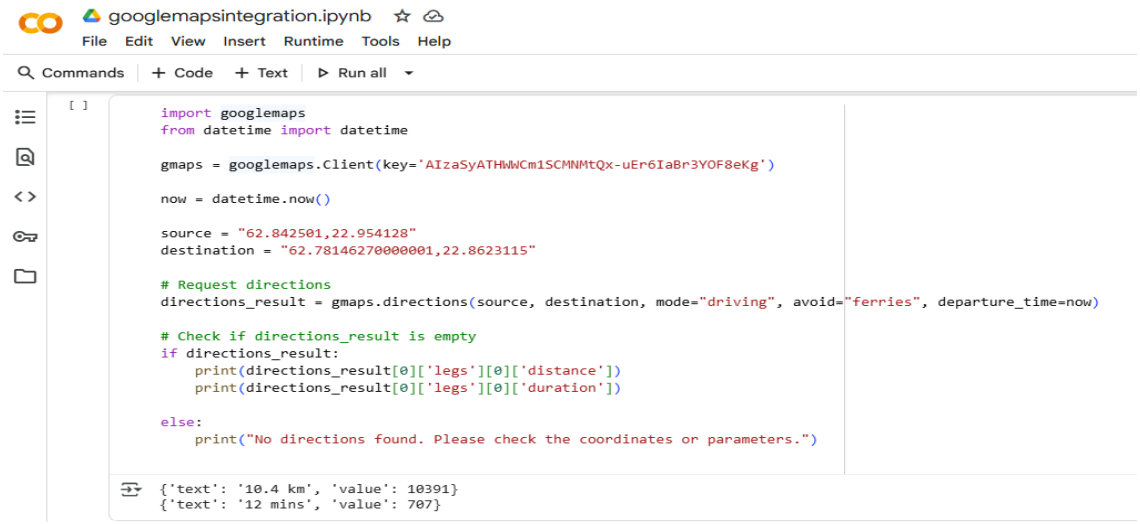

Figure 5: Code to fetch the distance with Time

The above code generates the distances with stipulated time required to reach to every distances between source and destinations are calculated manually using this code and put it in the table that helps to make further process become easier.

Python helps to establish the accuracy of the spatial references using its own geocoding techniques where the geographical longitude and latitude were secured based on this routing structure. The LOAD, UNLOAD and BOTH all locations were historically found and fetched using the API of Google geocoding platform with the manual entry, or clicking on the map or by the own QR based method for this application where reverse geocoding methods ensured the origin and destination points which represents the GIS interface. But the embedded technique of scanning the location address through QR does not require any manual intervention of copy and paste on the application, hence QR has accurate location coordinate that automatically minimized error which is more often commonly arise from manual coordinate collection(Hou & Liu, 2024).

The technique of optimizing the route through algorithms, the foundational inputs are served to determine the coordinate to generate distance matrix and asses leg-level performance, here the application that is designed for this thesis is following this structure. In the application, the same task is performed with the pointer based on the location address given to it precisely doing the repeated task consistently and ensuring the smooth reach towards geographic position requires more accuracy to take next smooth stoppage to perform journey.

Another paper ***"Quantifying carbon emissions in cold chain transport: A real-world data-driven approach. Transportation Research Part D: Transport and Environment.",*** (Lin et al., 2025) has depicted cold chain transport in the real world, the idea is added to the thesis from this paper which highlights the importance of using the real spatial data to accurately measure the transport emissions. The author has indicated the reconstruction of vehicle movement that is using the GPS data contains the time-stamped latitude and longitude information can be processed by Python-based programming. The CSS, HTML has been used to draw the polyline to precisely track pointer trajectories. The ideas are described in the paper very shortly but creating the application to determine the paths movement with the relation of calculating data and inserting values to the table based on the calculation ensured with formula.

3.5 How shared emission is allocated to the round trip for the vehicle using fuel: Diesel

The joint emission in the round trip is distributed proportionately based on SFD of each stop followed by GLEC framework to ensure each delivery's notional point-to-point activity (mass x SFD).

Table 2: Identifying Load, Unload and calculating Current weight

Load_Value	Unload_Value	Current_Weight
31617	20574	11043
24705	10800	24948
5382	7713	22617
	306	22311
	36	22275
	486	21789
	9000	12789
4500	360	16929
	16929	0

The updated Current Weight at each leg on the route is calculated based on the previous leg's weight and new loading/unloading. The vehicle brings whatever weight was left over from the previous **leg *i-1*** at the start of **leg *i***. While at the stop, a second Q number of items are loaded on to the vehicle, and another R number of items may be unloaded. Mathematically, we can represent this as:

$$\boldsymbol{CuW_n = CuW_{i-1} + Load_i - Unload_i} \quad (16)$$

The weight for ***leg i*** is then determined by obtaining the current weight as the last weight plus the amount that was loaded on that stop less what was unloaded.

Instead of recomputing from scratch the weight at each stop, the model maintains the previous weight and modifies it only by net effect of new operations. The mass increases with the load term, as more goods are stowed in the vehicle and decreases with the unload term when some goods are delivered.

Table 3: SFD, Consumption/100 Km, Calculating LVP, Consumption car × km

SL	SFD (Given)	LVP (Load Value Parameter)	Consumption/100 Km	Consumption car × km
1	10.4	12.27	**40.12**	4.172
2	205	27.72		82.246
3	202	25.13		81.042
4	202	24.79		81.042
5	198	24.75		79.438
6	197	24.21		79.036
7	200	14.21		80.240
8	142	18.81		56.970
9	142	18.81		56.970

The shortest feasible distance (SFD) value was given, which is the distance calculated from the source to each destination.

3.5.1 LVP (Load Value Parameter)

The partial factor consumed (litres per 100 km) for the truck was assumed to be one of its vehicle attributes. This parameter was estimated from operational data provided by the transit agency and considered constant for all segments of the line.

$$LVP_i = \frac{CuW_n}{Pallete\_Weight(kg)} \quad (17)$$

When maintaining this parameter constant, changes in total fuel consumption were guaranteed to only stem from differences in traveled distance plus the subsequent assignment rules

3.5.2 Consumption per 100 km

Total fuel consumption due to propulsion and flight control for each route leg was determined by taking the leg distance, multiplying it by specific mass flow of (lbs-l)/hr and dividing by 100.

$$\boldsymbol{Con_{vh} = 40.12\ L/100\ Km} \tag{18}$$

Using this method, we determined how much fuel a vehicle would need to travel along a specified segment. The output gave a fuel estimate at the route level based only on physical distance traveled, to obtain equivalent energy calculations between routes of different length.

3.5.3 Consumption Car × Km

Total fuel consumption due to propulsion and flight control for each route leg was determined by taking the leg distance, multiplying it by specific mass flow of (lbs-l)/hr and dividing by 100. Using this method, we determined how much fuel a vehicle would need to travel along a specified segment.

$$\boldsymbol{Fuel_i = \frac{Con_{vh}\ X\ d_i}{100}} \tag{19}$$

The output gave a fuel estimate at the route level based only on physical distance traveled, to obtain equivalent energy calculations between routes of different length.

3.5.4 Share of Capacity

A performance-based weighting factor was calculated for each leg, to represent how much of the vehicle's design capability has been usefully deployed. This variable was obtained by the ratio between the LVP of the leg and the nominal carrying capacity of truck.

$$\boldsymbol{Share\_Capct_i = \frac{LVP_i}{Con_{vh}} = \frac{CuW_n/Pallete\_Weight(kg)}{Con_{vh}}} \tag{20}$$

A proportional share of the payload is obtained, and, in turn, an allocation was established of the fuel use and emissions according to how much available capacity had been utilized.

3.5.5 Share of Consumption

A portion of the total propulsion fuel consumed on each leg was allocated to the types of cargo studied by multiplying leg-level fuel use with capacity share. In this process, the fuel necessary to carry only the pertinent cargo section was determined.

$$\boldsymbol{Consumtion\_Share_i} = \boldsymbol{Fuel_i} \text{ x } \boldsymbol{Share\_Capct_i} \quad (21)$$

This assignment enabled the emissions and energy to be distributed equitably among the transported loads using well recognized activity based costing and environmental accounting principles.

3.5.6 Why is Emission Factor used here?

If not, we shall end up again with the fuel consumed and an emission factor is necessary to be defined. The amount of fuel does not necessarily tell us anything about emissions. Fuel quantity is physical; emissions must be expressed in kg CO_2-equivalent for environmental impact assessment. It is a conversion factor from fuel to CO_2e in which each volume of one liter of the fuel yield how much weight of CO_2 emitted well-to-wheel.

$$\boldsymbol{EF} = \boldsymbol{Density\ kg/l\ x\ GHG\ Emission}\ (\boldsymbol{total/WTW\ kgCO_2/kg}) \quad (22)$$

The model then multiplies the assigned fuel volume by this factor to convert the energy use into a standardized emission metric that can be compared between legs, fuels and modes of transport. This method approach also allows for alignment with international reporting standards, ISO 14083 or the GLEC Framework, that specify that GHG emissions must be calculated using validated lifecycle emission factors and not raw quantities of fuel. Thus, the emission factor is key for translating measured physical fuel consumption into meaningful, reportable values of climate impact.

3.5.7 The emission calculation: CO_2

The figure on the CO_2e (kg) shows how much climate gas has been emitted by the fraction of fuel which is allocated to a leg. First a discrete quantity of fuel is allocated to that leg (the "Share of Consumption"), which varies depending on what proportion the amount of goods being considered occupies in relation to the total capacity of the vehicle. This fuel volume on its own is just a physical quantity in liters and does not yet tell you anything about greenhouse gas emissions. To make it CO_2-equivalent, the model multiplies the apportioned fuel (in liter) with a well-to-wheel emission factor which gives how many kg of CO_2-equivalents are emitted per liter when both combustion (tank-to-wheel) and upstream processes (extraction, refinery, distribution) are considered.

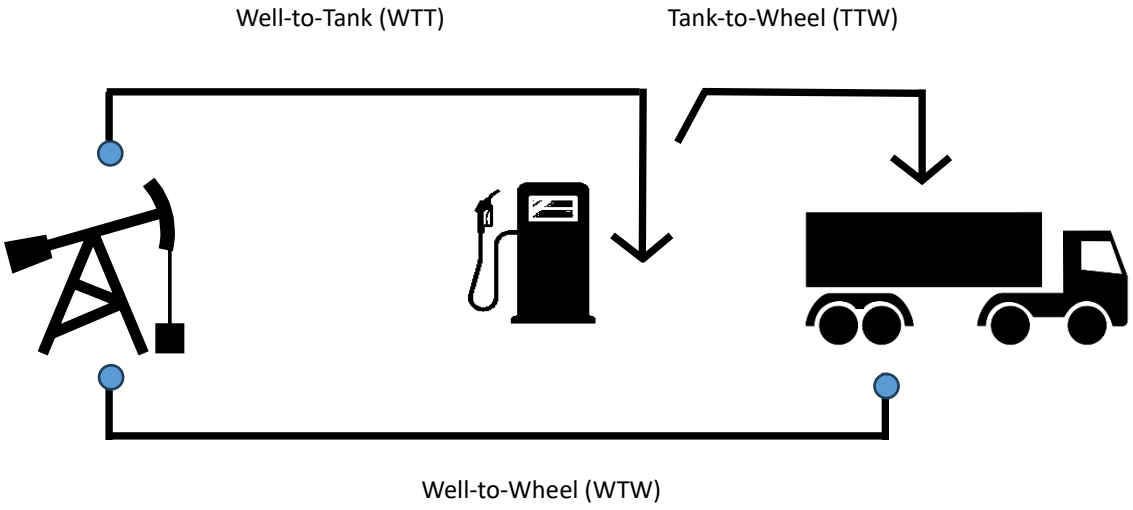

Figure 6: Illustration of process of oil tranfered untilmately to truck

The emission factor here is constructed from two variables: mass of fuel (kg per litre) and its life-cycle greenhouse-gas intensity (in $kgCO_2e$ per kg of fuel). This factor is used to get a standardized emission value from the energy use by multiplying the assigned fuel volume on leg i with it.

We combine on the sheet a density of 0.836 kg/L with a WTW intensity of 3.96 $kgCO_2e/kg$, so that each liter of this diesel, when considered over its full life cycle, generates 0.836 × 3.96 kilograms of CO_2-equivalent.

$$CO2e_i = ConsShare_i \times EF_{diesel} \quad (23)$$

This is in line with competing leg fair comparison, summation of emissions over the trip and reporting of results in a manner consistent with ISO 14083, GLEC and other green-house-gas accounting systems.

3.5.8 Enumerating TKM (tonne-kilometer)

The task of carrying goods on each leg was measured in tonne-kilometres by multiplying the mass load per vehicle (in metric tonnes) by distance.

$$TKM_i = \frac{CuW_n \; x \; SFD}{1000} \quad (24)$$

This measure offered a common industry standard for expressing the performance of freight transport and was used as reference for energy efficiency and allocation of emissions. Through using TKM, the model was consistent with popular reporting conventions adopted in logistics and carbon accounting literature.

3.5.9 Identifying Notional activity

The termed Notional Activity has been derived by multiplying the mass of goods unloade d on each leg (in tonnes) with the travel distances on the TCE.

$$Notional\_Activity = \frac{Unloaded\_Kg}{1000} \text{ x } SFD \quad (25)$$

This measurement represented the actual yield output attributable only to the unloaded material. It was a supplementary depiction of transported work and subsequently used to allocate accumulated environmental loads over delivery locations.

3.5.10 Sharing Allocation (%)

A percentage allocation was calculated for each leg by dividing the notional amount of that leg by the aggregate notional activity across all legs. In so doing, an even amount of total transport was allocated to each section.

$$\boldsymbol{Allocation}\ (\%) = \frac{\boldsymbol{Notional\_Activity}}{\sum \boldsymbol{Notional\_activity}} \text{ on each leg} \qquad (26)$$

3.6 Description of the scenario for the Two Distances Determination

Two different methods were applied for calculating transport distances for the identical delivery route.

Scenario A (Manual distance determination):

Distances were determined manually between successive loading–unloading pairs using a set of assumed routes and external map detection. The order of visits was a predetermined operational sequence that we received from **Table 1.**

(AX → BY → CZ → DX → EY → FZ → GX → HX → IY → AX).

Scenario B (Software-based shortest-path calculation):

Distances were calculated automatically by the software with a routing algorithm. The path was determined using a greedy shortest-path heuristic over Google Maps routing information based on minimum incremental distance rather than predefined operational order.

(AX → BY → IY → CZ → FZ → EY → HX → GX → FZ → AX)

Both cases refer to the same physical transmission problem, however, with different routing logic and distance estimation models.

The function of the application is to determine the difference between each distance which is collected using Google Map API. The distance from CZ -> DX is far longer than the distance from IY -> CZ. And the other sequence for each leg is calculated in the same way based on the distance calculation. Overall, the Greedy algorithm in the background

will identify the location first and calculate the distance of each destination in the background to identify the shortest feasible way.

Table 2: Qualitative comparison between two scenarios

Method Name	Determinant	Distance in Km
Scenario A Manual Method	AX → BY → CZ → DX → EY → FZ → GX → HX → IY → AX 10.4->191->65.9->6.1->2.9->11.1->7.9->59.5->197	551.80
Scenario B Greedy Method	AX → BY → IY → CZ → FZ → EY → HX → GX → FZ → AX 9.07->194.97->0.81->65.96->3.76->2.69->1.82->7.58->197.71	483.59

Absolute difference: 551.80 – 483.59 = 68.21 Km

Relative difference: 68.21 / 551.80 = **12.4%**

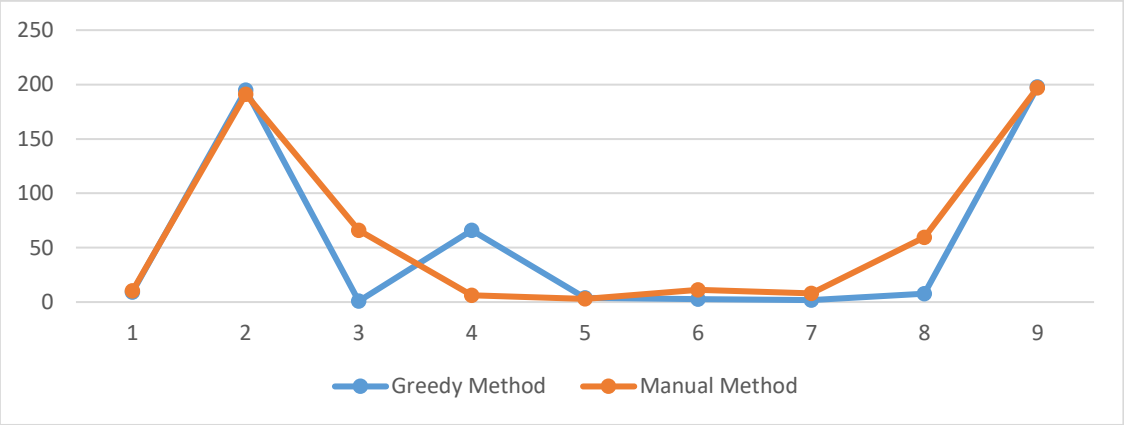

Figure 7: Distinguish between Two Scenarios

3.6.1 Explanation of the Underlying Phenomenon

The discrepancy observed is due to the routing mechanism, and not the result of measurements error. In the manual case, route order is predetermined and corresponds to an existing or historical way of working. This method tacitly incorporates restrictions such as driver experience, operational priorities or historical planning choices. The net

result is the car can be used to travel further (over a larger radius) if distant places are near.

The software-based method on the other hand uses a shortest-path greedy algorithm in which at each stage it chooses the stop that results in smallest amount of incremental distance. This logic detours between destinations based on their spatial proximity and orders the visits to them dynamically to minimize total travel distance. The resultant route is mathematically efficient but does not necessarily consider hard constraints from business operation, such as delivery time windows, contractual priorities, and unloading orders.

3.6.2 Implications for Emission Calculation

Comparison indicates that the choice of route has a major, direct effect on emission estimates. An over **68.21** km discrepancy in a single distribution loop can dramatically change well-to-wheel emission results. Thus, models of emissions that are purely distance-based and historical route–based might underestimate true emissions or leave available efficiency gains on the table.

This finding supports the methodological approach I have followed in this thesis where emissions are calculated either under a real-route or under a shortest-path assumption, making it possible to transparently compare operational practice with theoretical optimisation potential.

3.7 Activity-Based Modelling of Transport Refrigeration Unit (TRU) Fuel Consumption and Emissions (considering biogas truck)

In this paper, fuel use of Transport Refrigeration Units (TRUs) is modelled with a hybrid, activity-based method that explicitly considers both time-varying and distance-varying operational behavior of temperature-controlled road freight. In contrast to propulsion fuel, the energy requirements for TRU are not only dependent on distance travel but

also influenced by stationary operation, traffic conditions, loading/unloading activities and compressor cycling during door openings.

Energy required by TRU is not just distance dependent, but also affected by:

Operation (compressor cycle cooling, standby cooling, door openings, idle).
Vehicle dynamics (air change, fan duty, vibration induced thermal losses),
Driving situation (urban traffic vs. long-haul steady movement).

The ISO 14083 and GLEC Framework do not specify a single, static TRU fuel equation. Instead, they will rely on relevant and documented intensity factors for temperature-controlled operations and transparent conversion of energy usage to Well-to-Wheel (WTW) CO_2e emissions based on accepted emission. This is reflected in the TRU fuel consumption per transport leg, which is determined as the sum of a time-dependent and a distance-related component according to:

$$\boldsymbol{TRU_L = p \text{ x } hours + \frac{q}{100} \text{ x } SFD_{km}} \quad (27)$$

where ***p*** is the average TRU fuel consumption rate in L/h, ***q*** is travel related intensity (in liters/100 km), hours are the transport leg duration time and $\boldsymbol{SFD_{km}}$, the driven distance of the leg in kilometres. Parameter accounts for fuel usage in conjunction with time-based vehicular refrigeration loads, such as compressor running during wait times at rest, congestion or queuing and loading/unloading events, while q addresses about any extra fuel demand from airflow exchange, fan duty and auxiliary power associated with vehicle motion.

To realistically represent enough use conditions, each transport leg is assigned one of four driving cycles (TOWN, CITY, MIXED or LINEHAUL) as a function of the distance travelled for different average speeds. Each regime shares one user-editable (p, q) pair of parameters, permitting the model to account for different TRU duty cycles in urban, regional and long-haul operations.

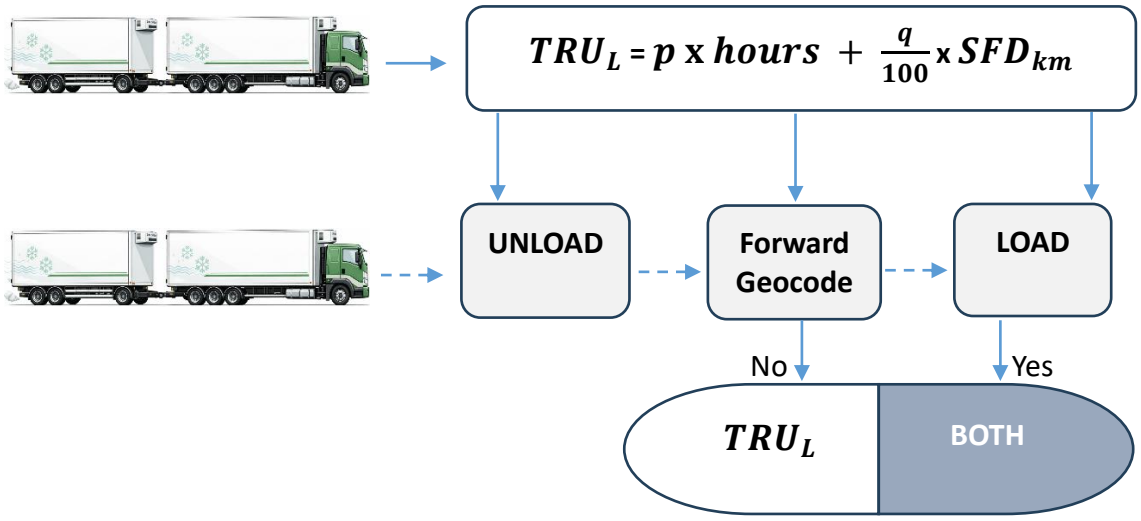

Figure 8: Illustration of Actual definitiion of formula

Table 3: Defining Classes based L/h and L/Km

	Liter/Hour	Liter/KM
Class	p (L/h)	q (L/100km)
TOWN	35.46	45
CITY	27.58	35
MIXED	22.06	28
LINEHAUL	17.33	22

The table defines operating classes with a hierarchy that connects flow rates in hourly (L/h) and distance-specific terms (L/100 km), so that different drive cycles can be consistently integrated into the model. The highest fuel intensity is observed in the TOWN class which represents several stops, low average speeds, and elevated idling time. CITY operation show somewhat lower values, that is, smoother traffic but still a significant congestion influence. The MIXED class represents the progression of both urban and inter-urban driving, during which an average consumption is obtained among those for city and extraurbain driving. Finally, the LINEHAUL class is for highway operation with constant speed and few engine interruptions, then it has the lowest fuel consumption in terms per hour (per kilometre). These classes can then be used to account for the realistic operational conditions when such measures as fuel consumption and emissions are scaled rather than using a single average factor. This categorization thus enables more appropriate leg-level emissions estimation by situating the fuel intensity under actual drive conditions on the route.

This above discussion allows explicit assumptions to be documented which may align with the ISO 14083 and GLEC Framework guideline to model efficient, activity-specific energy intensities for temperature-controlled transport rather than applying a single fixed factor. In this hybrid formulation, resulting TRU fuel consumption is then converted to WTW (well-to-wheel) CO_2e emissions via established emission factors, thereby attaining alignment with life-cycle accounting principles as intended by both standards.

At the end of the Method section, the leg-level transport output stands based on real route distances, vehicle properties and load dynamics which is obtained carefully and dynamically with several formulas. The upcoming Results section is designed to express these thought processes with detailed illustrations. Through the operationalization of geocoded inputs, QR-based address identification and map-based destination selection it is guaranteed that all route plans are established against real spatial coordinates rather than estimated driving distances. The iterative shortest-path and greedy assignment logic additionally organizes every trip in separate transport legs which correspond to the sequence of realistic load, unload and mixed operations. These stages enable us to standardly associate vehicle behavior and dynamic load variations with route-level distances. This implies that the produced leg-wise outputs build a consistent sample in emission language itself for interpretations. The Results section thus extends the groundwork, showing how the distances, routing selection and loading behavior in turn are translated into fuel use, transport work and CO_2e emissions. It analyses how impacts are spread among individual legs and loops, indicating where in the transport chain emissions concentrate. In the process, the results not only show numeric outputs but also display how methodological designs impact on observed emission profiles.

4 Results

Initially, a dashboard idea was prototyped with the aim of delivering operational value to company users. The dashboard is primarily aimed at finding the optimal route to support supervisory decision as well drivers instructing capability to follow an optimized path defined a priori. In view of this requirement, Google Maps was chosen as the routing engine because it can route the path continually from a source to multiple destination points based on reliable load amount generated with pre-set source location and loading amount at first enabling all unloading from all destinations prior to returning for closing an operational cycle.

In practice, the dashboard was built with two interactive maps to visualize this concept. The source location and all destinations can be chosen through one of the three alternative methods using the first map.

- Preparing a QR of the location based on longitude and latitude and scanning the QR using the device camera.
- Click on the specific point/location/building on the map to select the place.
- Copy the location address from some other document and paste it into the application.

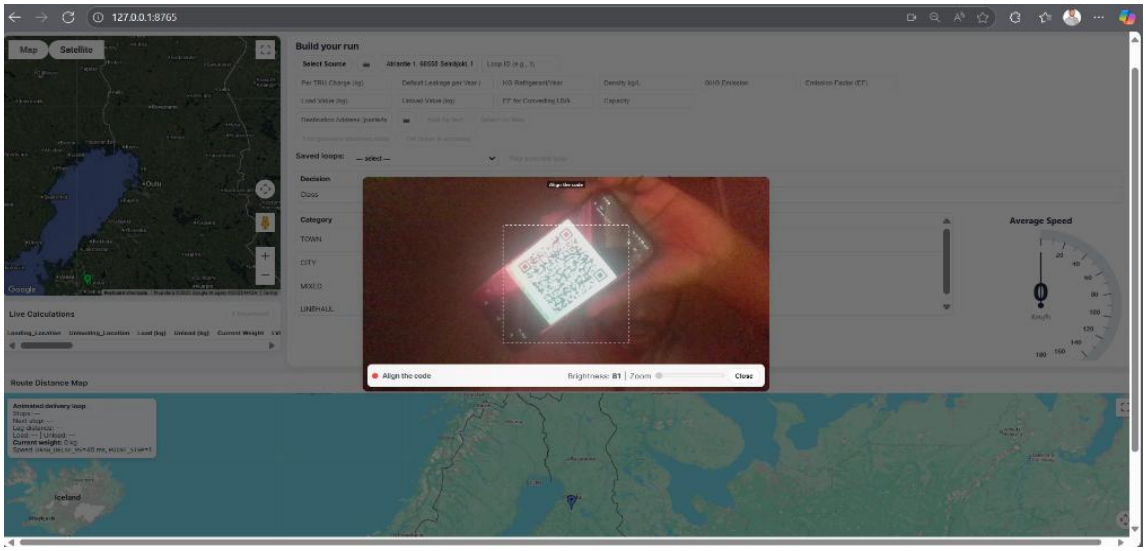

Figure 9: Implementing QR fetching of location address in the application based on Lat-Long

In practice, the dashboard was built with two interactive maps to visualize this concept. The source location and all destinations can be chosen through one of the three alternative methods using the first map. For one, a QR code of geographic coordinates can be created and read directly from the device's camera. Its second, you can just select a goddamn location or building or what have you on the map interface. Thirdly, a place address may cut and paste from another document into the application for geocoding. When picking the source and each destination separately the form also has an input field to enter loading/unloading for this pickup/drop-off.

First, it requires putting the Load Value and Unload Value into the application. The next phase is to open the camera to scan the QR. While the camera can fetch the QR successfully, it takes the position in the application with a line of specification of the values, with the exact location name as it is embedded in the QR code. In addition, by the quick motive, the position in the left map has been spotted with an orange/red mark.

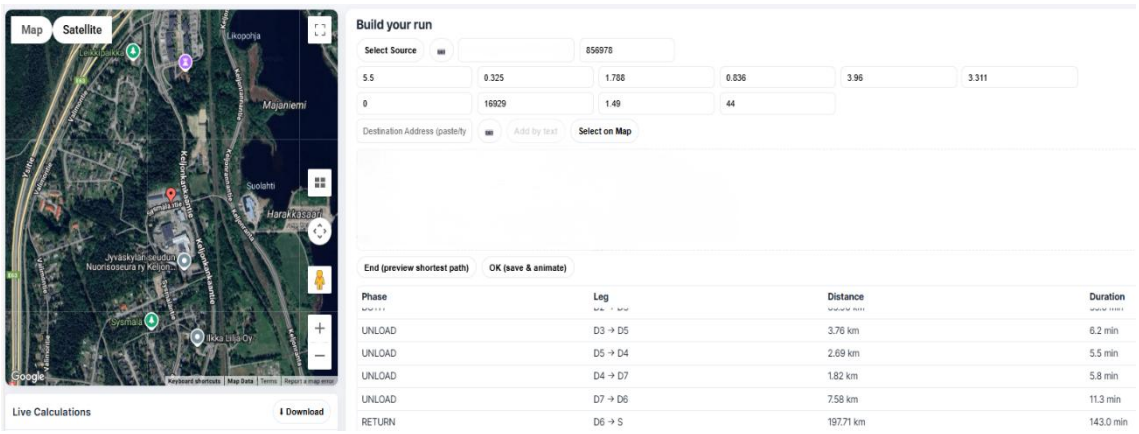

Figure 10: Inserting input into the necessary box

Another option that the user can use is to copy the location from another document and paste it into the Destination Address box. While clicking on the" Add to text" box, the location will be added as a destination. The third one is clicking exact location on the map by pointing to the mouse smoothly that helps fetching the location address to the screen according to the longitude and latitude. These processes help to address the point on the screen with proper mark as "Maroon" pointer as destination.

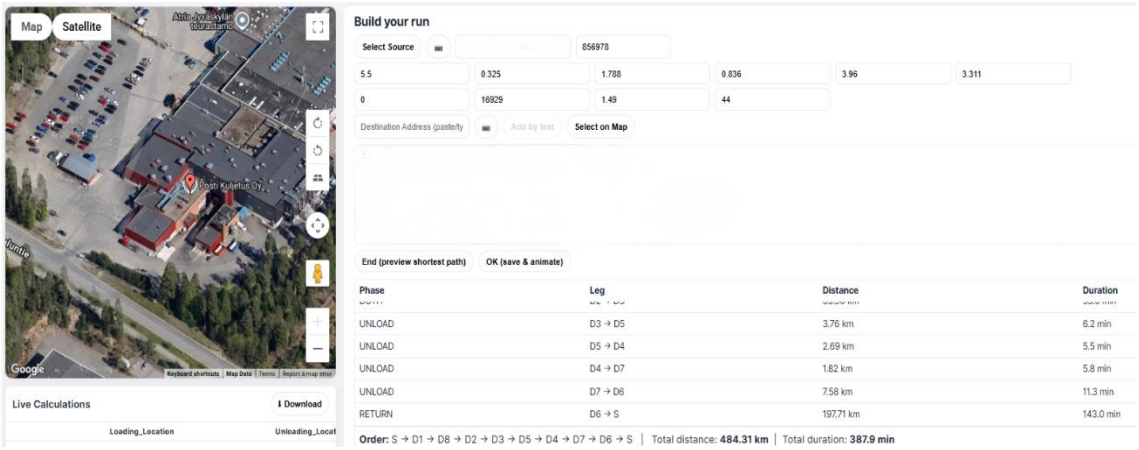

Figure 11: Focusing the Location's destination by hovering the mouse

4.1 Map Container Code Snippet

A simple idea that convinced me to prepare a route builder, which is more prudent to provide a natural vibe, where I attempted to put all sources to active, putting information of source and destination location in three ways as discussed earlier. Setting up HTML containers following plain blocks that are used to hold two maps: a small builder map, which I tried to fit in the top, and a large map, which will showcase the animation has been set below. Google's JavaScript library is used to call the <script> tag that enables the geometry tools to define distance and draw polylines with placeholders to define the specific address inside the map by putting the map logo on it.

Table 4: Map building, Positioning, Live calculation initiation

```
<!-- Builder row: LEFT = (mini map + metrics), RIGHT = controls -->
 <div class="panel">
  <div class="left-col">
   <div class="mini-wrap"><div id="finland"></div></div>
   <!-- Live calculations card -->
   <div class="metrics card">
    <div class="card-header">
     <span>Live Calculations</span>
      <button id="dlExcel" class="icon" disabled title="Download Excel">⬇ Down-
load</button>
    </div>
```

Table 7 shows the arrangement of the screen to indicate clearly the map, position and live calculation results. The interface is divided into 2 halves, the left one shows a mini-map and live calculations numbers with controls shown at right now; this was to keep data on space always together making the base regenerating visibility. The map area is utilized to display the chosen location and beneath, the live calculation card is then up-dated in real-time as the backend processes. There is a download button added, and it has been disabled until there is results to show the.

4.2 Config: Finland Bounce Padding

Then, comes the scope to identify Finland's map to be merged inside the first map within the boundary box to keep the users oriented within the limit of the surroundings, as if they try to drift to other places like the Atlantic, it may not happen. So, the restriction process is used by calling a specific function by defining FINLAND_BOUNDS and writing fitFinland() to clarify my justification. Camera function is now set to remain activated inside Finland, also in the same condition while clamping the zoom by mouse hovering. So, the consequences are that a map is always open while work opens.

Table 5: Binding Top Map of Finland inside limit Box to fit it

```
<script   src>="https://maps.googleapis.com/maps/api/js?key={{API_KEY}}&lib>rar-
ies=geometry,places"></script>
const FINLAND_BOUNDS = new google.maps.Lat>LngBounds(
  <new google.maps.LatLng(59.3, 19.1),
  new google.maps.LatLng(70.31, 31.6)>
);
const FIT_PADDING = { top: 12, right: 12, bottom: 12, left: 12 };

function fitFinland()<{
  if(!finMap) return;
  finMap.fitBounds(FINLAND_BOUNDS, FIT_PADDING);>
  const z = finMap.getZoom();
  if(z > 5) finMap.setZoom(5);
}
```

In Table 8 it is focused on how the upper map over Finland is clamped in a limited display box so that the whole country fits into it. Finland is geolocated, with a map and the potentially visible coordinates of where it may be spread out on a plot, this means that it has been strongly indicated latitude and longitude limits for how things can potentially influence one another. I used this feature to make sure that Finland (latitude 60 N == half Europe height) is always in the center of the map AND completely visible and zoom level does not go over some max limit. This may be a consistent and readable map view without unwanted panning or zooming on interaction.

4.3 Top Builder Map: initFinland(), GeoCoding, Point-Peak

Special attention was given to the construction of the initial “builder” map using the **initFinland()** function, where we set up features such as map center, zoom level and user interface controls that react to requests made by users. WHILE SELECTING LOCATION VIA MOUSE On selecting a location through the mouse, a button “Select Source” will now be made available to trigger the automatic fetching and loading of details of selected location right into input field near camera icon.

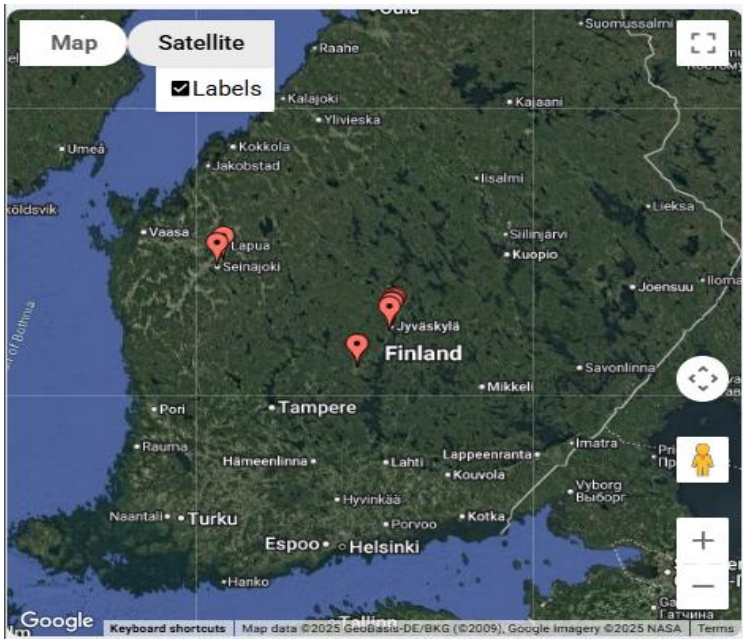

Figure 12: Adjusting Top Map inside the box by identifying the location by QR scanning/directly inputting the location address

Another fundamental feature of the system is to integrate a phone's camera into the system for QR scanning to obtain discretely embedded geographic locations. This method shows how the longitude and latitude encoded in the QR code can be decoded and used to further calibrate the zooming+positioning of the map with better accuracy. A similar process is repeated when choosing a destination, the camera-based functionality scanning QR codes in the same way to obtain and interpret a location based on that data.

There is an issue that may clutter the plan, so it is focused on the duplicating stops to be avoided using the function **upsertDest()**. This function works to identify any newly added place that is matched with an existing one, depending on the normalized name.

In the case of matching, the new load/unload values are merged into the old stop instead of creating any new second point. It sustains the cleanliness of the route by ensuring data accuracy, and a red pointer has been added as marked on the location to picturize the site.

Table 6: Fetching position by clicking on the location, calling greedy process

```
function initFinland(){
  finMap = new google.maps.Map(document.getElementById('finland'), {
    center: {lat: 64.5, lng: 26.0},
    zoom: 5, minZoom: 4, maxZoom: 18,
    restriction: { latLngBounds: FINLAND_BOUNDS, strictBounds: true },
    zoomControl: true, mapTypeControl: true, streetViewControl: true, fullscreenCon-
trol: true,
    gestureHandling: 'greedy', draggableCursor: 'pointer', draggingCursor: 'grabbing'
  });
...
...
//Taking the source and Destination by clicking on the map
  finMap.addListener('click', (e) => {
    if(!picking) return;
    reverseGeocode(e.latLng).then(name => {
      if(picking === 'src'){
        source = { name: name || 'Source', lat: e.latLng.lat(), lng: e.latLng.lng() };
        if(srcMarker) srcMarker.setMap(null);
        srcMarker = new google.maps.Marker({
          position: e.latLng, map: finMap,
```

```
        title: 'Source',
        icon: 'http://maps.google.com/mapfiles/ms/icons/green-dot.png'
       });
       picking = null; fitFinland();
     } else if(picking === 'dest'){
       const dest = { name: name || 'Destination', lat: e.latLng.lat(), lng: e.latLng.lng(),
load: 0, unload: 0 };
       upsertDest(dest);
       picking = null; fitFinland();
     }

   });
```

In the above code, my plan is to determine user selection of locations via clicks on the map which is is initialized with limiting zoom levels and geographic area to restrict navigation within Finland, while greedily processing user gesture are processed smoothly. When a location on the map is clicked, system captures latitude and longitude of the point clicked and then it converts into human readable location name without any user input using reverse geocoding. I try to use that method to make source and dest points selectable easily, without having to hand-enter addresses. If the user is selecting two points, then a marker on map will be put accordingly, and reset location or put new marker if exist in previous sate while it's being done. In this manner, locations are placed incrementally and the map is refitted after each one to ensure all relevant points are visible for future route requests.

Next time, for the second map to focus on using the builder as solid, I transformed to the" show me" part- that picturize bottom animation map in initRouteMap(). A big canvas is created for smooth movement that is viewed where the addition of a blue" mover" marker and a DirectionService is asked to Google for making the routes on its path. The getRoutePoints() function is heavily lifted to catch the route polyline for decoding it to the actual coordinates. If it happens, as the service cannot originate the polyline, then a fallback plan is ready to draw a straight line up to the destination from the origin. In that case, it ensures no dead ends, while the animation will always run actively.

Table 7: GoogleMap's position status checking, polyline drawing method

```
function getRoutePoints(origin, destination){
  return new Promise(resolve => {
    dirService.route({
      origin, destination, travelMode: google.maps.TravelMode.DRIVING,
      provideRouteAlternatives: false, avoidFerries: true
    }, (result, status) => {
      if(status !== google.maps.DirectionsStatus.OK || !result.routes?.length){
        resolve([new google.maps.LatLng(origin.lat,origin.lng),
              new google.maps.LatLng(destination.lat,destination.lng)]);
        return;
      }
      const route = result.routes[0];
      if(route.overview_polyline?.points){
        resolve(google.maps.geometry.encoding.decodePath(route.overview_pol-
yline.points));
      } else if(route.overview_path?.length){
        resolve(route.overview_path);
      } else {
        const pts = [];
        for(const leg of route.legs){
          for(const step of leg.steps){
            if(step.polyline?.points){
              constdec          =          google.maps.geometry.encoding.decodePath(step.pol-
yline.points);
              for(const p of dec) pts.push(p);
            }
          }
        }
        resolve(pts.length ? pts : [
          new google.maps.LatLng(origin.lat,origin.lng),
          new google.maps.LatLng(destination.lat,destination.lng)
        ]);
      }
    });
  });
}
```

The process of the system checking route status in Google Map and Getting points for Polylines where Table 10 depicts how the system checks the route status in google map and extracts the points required to draw polyline on map. It calls a function to get driving directions from the selected origin and destination, then checks the response to ensure that a valid route is available. Once a valid path is located, decoded path points

from the Google Maps polyline of type are extracted and transformed into sequence of latitude–longitude. A direct line between start and destination, whether route data is missing or not complete. I applied this principle to ensure a route is displayed on the map even when we don't have detailed routing data. Thus, route visualization can be done in a robust manner and the polyline is always rendered based on the most accurate path info.

The aftermath that stands, the route is exposed lively while focusing on the runLegs() and drawPoints() as the runLegs() works on walking leg-by-leg through the itinerary aptitude searching for the points and plotting them. These points are pushed to draw the polyline incrementally, while the drawPoint() function works smoothly, which indicates the smooth 'driving' animation that ensured the advancement of the mover marker. The addition of a few vertices at a time is illustrated as a little illusion, which turns the static data into a story that our eyes can follow.

4.4 The functionalities of boot ()

A single function: boot() is switching all the boots, whereas on the page load," window.onload=boot" makes the initialization of the builder map and the animation map altogether. The UI started its work from that point to pick a source and by adding the destination with previewing and functioning play all together. Each function features its weight under common agendas: initFinland() focuses on the top map, binds the click parameters with attaching Places Autocomplete to ensure the input for the smooth entry to the addresses. Human text and the following coordinates are deeply investigated with reverseGeocode() and forwardGeocode() functions so that they can be displayed in friendly names and enumerate the exact position carefully. Duplication and aggregation of load/unloads are enforced to ensure proper data integrity upon calling the function upsertDest().

Table 8: Route initiation call, setting position of pointer animation

```
 function drawPoints(polyline, points, token){
    return new Promise((resolve)=>{ let idx=0; function tick(){ if(token!==animTo-
ken){ resolve(); return; } if(idx>=points.length){ resolve(); return; }
              for(let c=0;c<POINT_STEP && idx<points.length;c++,idx++){ pol-
yline.getPath().push(points[idx]); }
      const pos = points[Math.min(idx-1, points.length-1)]; mover.setPosition(pos);
routeMap.panTo(pos); setTimeout(tick, DRAW_DELAY_MS); } tick(); });
 }
  async function boot(){ initFinland(); initRouteMap(); updateButtons(); await re-
freshLoops(); }
 window.onload = boot;
```

Preparing initRouteMap() for animation is emphasized, whereas the route is always following the polyline and straight line under the calling of the function getRoutePoints(). The orchestration of the movement is managed by runLegs() and drawPoints() so that the path feels lively without any dumping.

4.5 The thought process for mapping destinations for the table “Phase / Leg / Distance / Duration”

That preview card is only viewed after the server has created the route and the entire matrix showing the distances, times, and stops of travel. When End Button is selected, the browser sends and receives data to its parallel transfer server to find the shortest route. The request goes to the server, where Handler.do_POST, a routine, figures out where to send the request by planning the route. A planner constructs a mapping that indicates a certain order between stops for the intended vehicle route. Finally, the requesting vehicle asks for the O×O matrix, which is a simple square matrix where O is the whole set of axes.

Table 9: Source Position’s Latitude and Longitude declaration

```
points = [(src['lat']src['lng'])] + [(d['lat']d['lng']) for d in dests]
full_distfull_dur = batched_distance_matrix(pointspoints)
```

4.5.1 Greedy Loop

When the phase matrix is loaded up, the code will assign a phase tag to each phase, for example, load, both, or return, and then determine which route it is going to take to get from one index to the other. After the server places the information into a table, it passes that data to anything, which converts the numbers into their corresponding text, the source being: S, the destinations being: D1, D2, etc. The statistics from the searched database are returned to the requesting source as a JSON data record. This is how the preview of the picture is built into HTML.

The formula works to find the distance of every individual destination from the source address first. The reason behind this is to identify the actual distance from the source to every destination, and to calculate which destination is preferable to move first. Then the program works to identify the distances among the destinations to identify the possible shortest path to go next destination after completing the first move from the source to the first destination.

Table 10: The order path from source to destination and other destination

```
while remaining:
   loads   = [i for i in remaining if kind(i) == "LOAD"]
   boths   = [i for i in remaining if kind(i) == "BOTH"
            and (onboard + idx_to_stop[i]["load"] >= idx_to_stop[i]["unload"])]
   unloads = [i for i in remaining if kind(i) == "UNLOAD"
            and (onboard >= idx_to_stop[i]["unload"])]
   if loads:   pick = nearest(curr, loads)      # priority 1
   elif boths: pick = nearest(curr, boths)      # priority 2
   elif unloads: pick = nearest(curr, unloads)  # priority 3
   else: pick = nearest(curr, list(remaining))  # fallback (cap unload)
   # update onboard safely (cap unload)
   if di["load"] > 0: onboard += di["load"]
   if di["unload"] > 0:
      feasible_unload = min(di["unload"], onboard)
      di["unload"] = feasible_unload
      onboard -= feasible_unload
```

Inside Table 13, the order of routes that the traveler visits from source to destination, is sequentially maintained while traveling in a valid path and visiting all other stations. For

that, the algorithm continuously examines the remaining sites and determines which ones are LOAD/UNLOAD/BOTH for each of them where a load/unload is possible at that stop having the given onboard cargo. First preference is a LOAD location, second a BOTH operation and third an UNLOAD location based on the nearest stop if available. I employed this type of priority-based greedy logic to ensure that cargo handling is done reasonably and discharging is never attempted when there's not enough load! The quantity is safely updated after every stop so that unloading is capped. Thus, the path is established and routing efficiency and load can be scheduled over the path.

So, the concept is like the K-nearest algorithm that works fruitfully. But the major calculation lies in the Load and Unload balances. If there is no adequate Load, it must move to the Unloading Location until the truck has a proper Loading Amount. There is another factor, which is Capacity, which is accountable because the Truck may not proceed if there is more weight carried in the Truck.

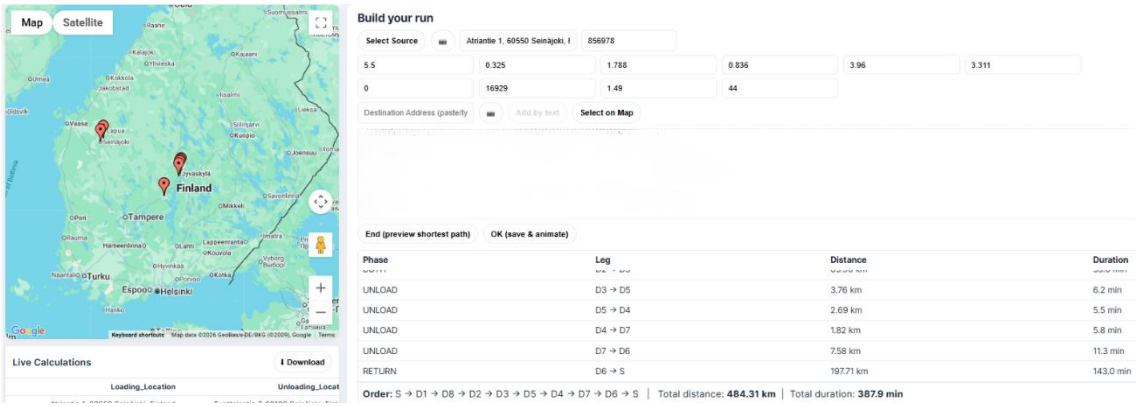

Figure 13: Shortest Path calculation, Greedy approach, Distant calculation

The preview of the items is rendered by the formatting handler, which builds the HTML table and the text by string [list of items]. In other words, the "destination mapping" you see (S, D1, D2…) on the screen is arranged so you can easily see where the different stages of the route are holding up.

4.6 The planning algorithm used for the calculation, and how the Current_Weight is calculated inside the coding

The construction of the route applies to a greedy scheduler that considers distance constraints ever to start off, the function will first remove duplicate stops that have the same place and similar names. This means that all the loads/unloads at the same place will be merged, such that in the end, only one unique stop at a place will be retained with all the quantities added up. The points array is then built, and batched_distance_matrix is called to fetch a full pairwise distance/time matrix from Google's Distance Matrix API (batched 10×10 to respect API limits and speed). Kilometers and minutes are stored in full_dist/full_dur.

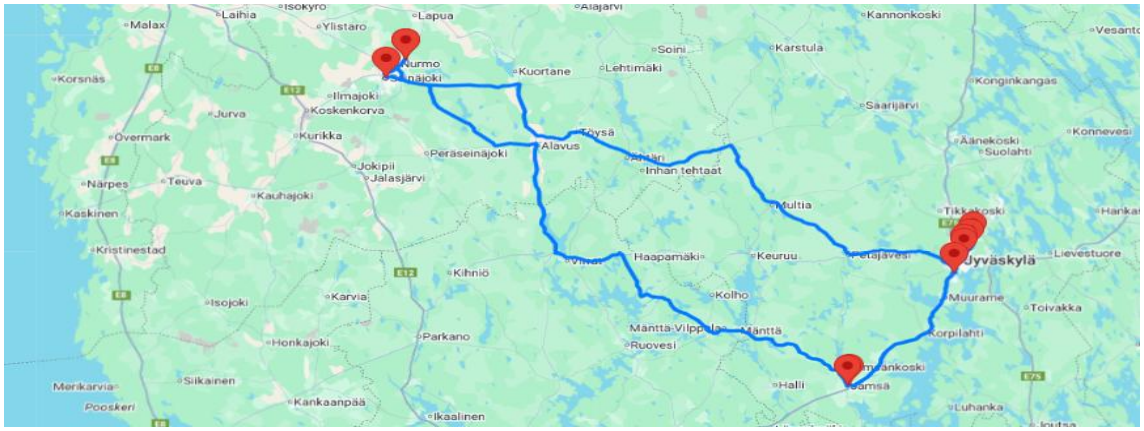

Figure 14: Polyline drawing through CSS, HTML and JavaScript backed using the advice of Google Map

The core loop leaves the collection of unvisited stops and current "onboard" weight, so the plan never goes negative. At every move, it creates three qualified tiers sorted by priority. The first includes all remaining LOAD stops. The next includes BOTH stops whose unload we can make feasible if we first add their load. The last includes UNLOAD stops whose unload is already feasible. So, we do this at every move. If no stop can be made, it still picks the nearest remaining one and restricts the unload to whatever weight is on board. Each destination is included exactly once.

Priorities: LOAD > BOTH > UNLOAD.

Feasibility states that the onboard weight must never go negative, if an UNLOAD would cause this.

Always take the next closest stop (using full_dist matrix) to a priority tier.

It goes to every destination once before returning to the source.

Table 11: The function works on leg identification to find short distance

```
if route_global[-1] != 0:
    phase_legs.append(("RETURN", route_global[-1], 0))
    route_global.append(0)
```

After each pick, update onboard safe (add load, deduct min(unload, onboard)), log leg (phase_tag, from, to), and continue until all are visited, then append a RETURN leg back to the source. The values in the preview are not recalculated heuristics; they come from the matrix: for each leg (a,b), full_dist[a][b] (in km) contains the distance, and full_dur[a][b] (in min) the duration. The batched_distance_matrix populates that matrix, which contacts Google's API for each batch, converts meters to kilometers and seconds to minutes, fills the correct slices in that O×O arrays, retries on transient failures, and leaves cells at infinity when no route is returned; those infinities are displayed as "∞" in preview. Essentially, this algorithm is a constrained, priority-aware nearest neighbor heuristic. It is simple, explainable, and fast. Also, it harnesses the power of Google's authoritative travel metrics rather than geometric shortcuts.

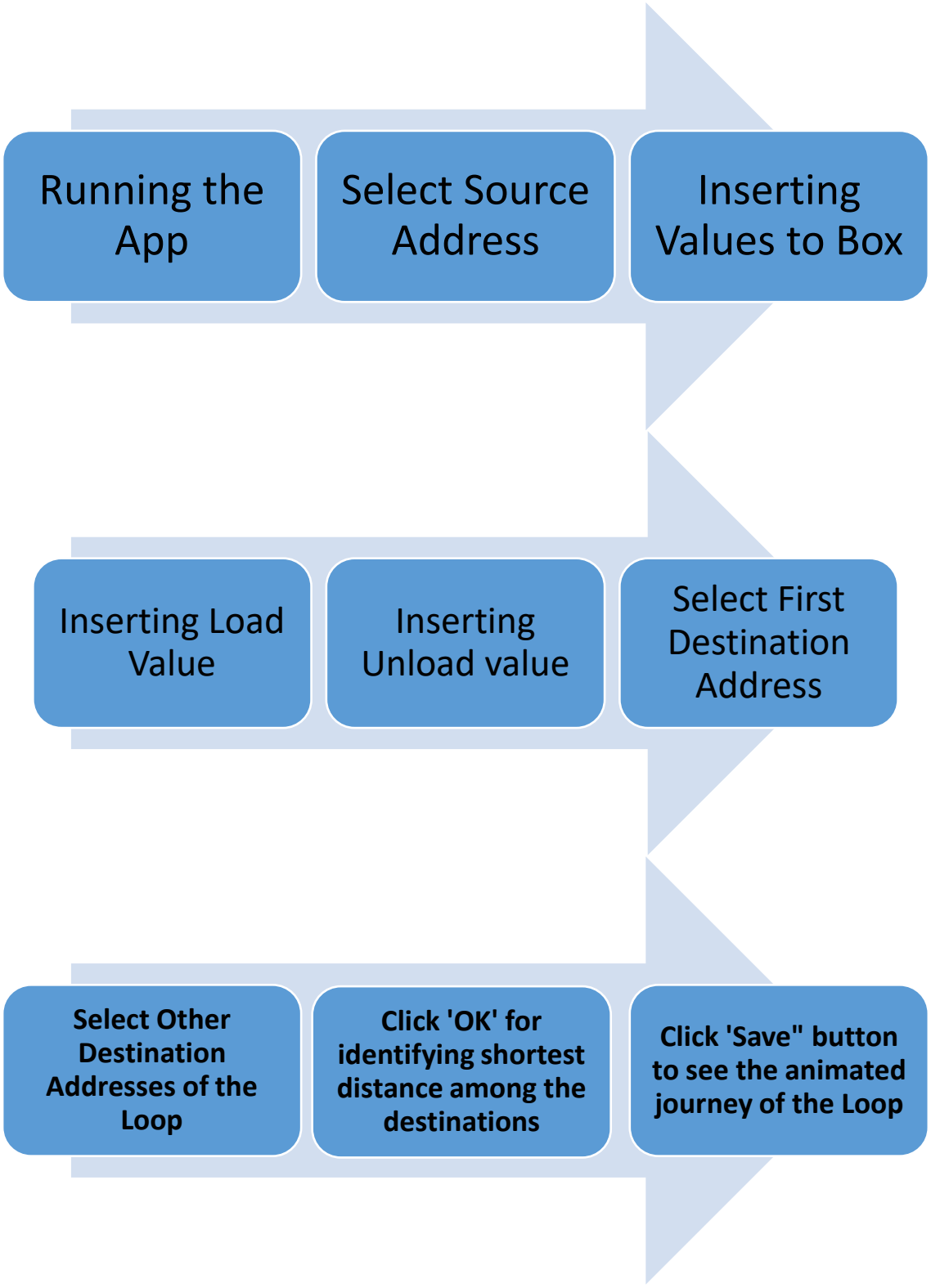

Figure 15: A quick direction to run the application

5 Conclusion

The study is extensively focused on delving into the deep of the route mapping for the anonymous company, deeply concerned about the emission of CO_2 as they are moving their truck inside Finland in different route every day. The focus point of this thesis is to estimate the shortest possible route for the truck movement with additional calculation related to the journey that has the direct impact on finding the actual status of the emission of CO_2.

This study developed and validated an operational, transparent methodology to estimate route-level emissions for temperature-controlled road freight by integrating routing logic, activity data and standards-based emission accounting within a unified process. The key insight is associating the actual structure of a journey (legs, distances, hours, speeds), with behavior of the cooling unit and refrigerant leakages instead using coarse-grain trip-averaged factors.

5.1 Summary and Key findings

A significant result of this work is the introduction of a dual map dashboard design that allows flexible and accurate location input of different types, including QR code scanning, dynamically mapping selection by dragging or clicking on map, and searching addresses by manually entering address name. This layout enhances user-friendliness, minimizes data-entry errors and guarantees the legitimacy of geocoding with latitude and longitude latitude/longitude extraction. The routing engine (based on Google maps) can find reasonable routes to do all loading and unloading dock-stops and return the truck back to source, thus reflecting logistics cycles in real life.

Operationally, this dynamic computation of current vehicle weight and LVP for route sections allows measuring the percentage of capacity filled. This eliminates any sudden situation where the vehicle capacity is violated at one of the trip legs, promoting better

safety and compliance. The analysis also goes beyond classical routing to consider transport activity dynamics including distance travel, duration, and ton-kilometers (TKM) and refrigerating energy requirements.

An important contribution to this study includes the creation of a transparent activity-based TRU (Transport Refrigeration) fuel and emissions model. When fuel consumption (liters/hour) and distance-based relative fuel consumption (liters/100 km) components are combined, the model accurately represents refrigeration duty cycles over a range of drive cycle types (TOWN, CITY, MIXED and LINEHAUL). The transformation of fuel consumption to well-to-wheel (WTW) CO_2e emissions, together with the apportionment of refrigerant leakage emissions over route legs, guarantee adherence to ISO 14083 and GLEC Framework standards. This method proves that discrete leg-specific emission accounting can be accomplished without the use of over-simplified or constant emission factors.

5.2 Future Directions

Though the system achieves significant performance, there are many potential improvements that could be made. One possible direction is to broaden the list of criteria for route optimization considering other alternatives like fastest, more fuel-efficient or less emission routes and allowing users to specify constraints such as avoiding highways or certain maneuvers. In addition, the inclusion of additional mapping and routing services (eg., Mapbox, or open-source routing modules) would probably improve flexibility as well as reduce chances of API failures.

As a future work, from the perspective of sustainability, emissions model can be further expanded to consider multi-fuel type, electrified Refrigeration Unit (e-RU) and hybrid vehicle scenarios and make it applicable in transitioning fleets. Adding real time vehicle telemetry and IOT sensor data, then we can dynamically adjust fuel and emission calculations based on actual operating conditions instead of planned estimates.

Finally, including even more operational limitations such as legal limits on driver working hours, rest periods and safety rules into routing logic to support compliance-driven planning. By gradually adding these dimensions, the dashboard was going to transform itself from a planning tool into a fully-fledged platform of logistics management.

Finally, this study shows that well-designed electronic systems could both enhance logistics efficiency and environmental responsibility. Through the pursuit of specific scope with future expandability, the proposed framework provides near-term practical relevance as well as long-term innovation space in sustainable freight transportation.

The Observation Link:

https://drive.google.com/file/d/1sTPGZ6z7s5yAbO2a5G4XM_Au18NifYEQ/view?usp=sharing

References

Davydenko, I., Hopman, M., Fransen, R., & Harmsen, J. (2022). Mass-Balance Method for Provision of Net Zero Emission Transport Services. *Sustainability*, *14*(10), 6125. https://doi.org/10.3390/su14106125

Dobers, K., Jarmer, J.-P., & :unav. (2025). *Guide for greenhouse gas emissions accounting at logistics hubs*. Fraunhofer-Gesellschaft. https://doi.org/10.24406/PUBLICA-4159

Du Plessis, M. J., Van Eeden, J., Goedhals-Gerber, L., & Else, J. (2023). Calculating Fuel Usage and Emissions for Refrigerated Road Transport Using Real-World Data. *Transportation Research Part D: Transport and Environment*, *117*, 103623. https://doi.org/10.1016/j.trd.2023.103623

FABRIS F, FABRIZIO M, MARINETTI S, ROSSETTI A, & MINETTO S. (n.d.). *Comparison of the environmental impact of HFC and natural refrigerant transport refrigeration units from a life-cycle perspective.* [Dataset]. International Institute of Refrigeration (IIR). https://doi.org/10.18462/IIR.NH3-CO2.2023.0033

GLEC, S. F. C. (2025). *Global Logistics Emissions Council Framework for Logistics Emissions Accounting and Reporting* (p. 171). Smart Freight Centre. https://www.smartfreightcentre.org/en/our-programs/emissions-accounting/global-logistics-emissions-council/

Hou, D. N., & Liu, S. C. (2024). Optimization of cold chain multimodal transportation routes considering carbon emissions under hybrid uncertainties. *Advances in Production Engineering & Management*, *19*(3), 315–332. https://doi.org/10.14743/apem2024.3.509

International Organization for Standardization, S. (2023). *Greenhouse gases. Quantification and reporting of greenhouse gas emissions arising from transport chain operations (ISO 14083:2023)*. ISO. https://www.iso.org/standard/78864.html

Li, X. (2022). Multiparty Coordinated Logistics Distribution Route Optimization Based on Data Analysis and Intelligent Algorithm. *Journal of Sensors*, *2022*, 1–9. https://doi.org/10.1155/2022/6053332

Lin, H.-J., Chen, P.-C., Lin, H.-P., & Hsieh, I.-Y. L. (2025). Quantifying carbon emissions in cold chain transport: A real-world data-driven approach. *Transportation Research Part D: Transport and Environment*, *142*, 104679. https://doi.org/10.1016/j.trd.2025.104679

Maiorino, A., Petruzziello, F., & Aprea, C. (2021). Refrigerated Transport: State of the Art, Technical Issues, Innovations and Challenges for Sustainability. *Energies*, *14*(21), 7237. https://doi.org/10.3390/en14217237

Olivari, E., Caballini, C., & Lluch, X. (2025). How to calculate GHG emissions in freight transport? A review of the main existing online tools. *Case Studies on Transport Policy*, *19*, 101343. https://doi.org/10.1016/j.cstp.2024.101343

Qiang, X., Appiah, M. Y., Boateng, K., & Appiah, F. V. (2020). Route optimization cold chain logistic distribution using greedy search method. *OPSEARCH*, *57*(4), 1115–1130. https://doi.org/10.1007/s12597-020-00459-4

Sar, K., & Ghadimi, P. (2023). A systematic literature review of the vehicle routing problem in reverse logistics operations. *Computers & Industrial Engineering*, *177*, 109011. https://doi.org/10.1016/j.cie.2023.109011

Schmidt, V., Aschauer, F., Scott, A., & Gerschberger, M. (2025a). Does a detailed freight emissions calculation really matter? *Transportation Research Part D: Transport and Environment*, *148*, 104761. https://doi.org/10.1016/j.trd.2025.104761

Schmidt, V., Aschauer, F., Scott, A., & Gerschberger, M. (2025b). Does a detailed freight emissions calculation really matter? *Transportation Research Part D: Transport and Environment*, *148*, 104761. https://doi.org/10.1016/j.trd.2025.104761

Yin, Y., Wang, H., & Deng, X. (2024). Real-time logistics transport emission monitoring-Integrating artificial intelligence and internet of things. *Transportation Research Part D: Transport and Environment*, *136*, 104426. https://doi.org/10.1016/j.trd.2024.104426

Appendices

Disclaimer

This Master's Thesis has been suggested with the referrences of Artificial Intelligence specially for Python Coding section implementation. But, the tools used here are well known and trying to maintain the instruction of the University of Vaasa.

An overview of the AI tools that are deployed, i.e.:

The AI applications that are in use, however:

- ChatGPT: To talk about progress on the development of coding, discuss ideas to iterate and use chats generated from Chatgpt to further develop greedy algorithm.
- Grammarly: For a grammar and spell check as well as writing Decent error free sentences.
- Perplexity AI: We speak of coding for our CSS, HTML or Javascript right here in this development to bet out even more realistic lines of code set up precisely the same way and no script table code is completely verified audited ussing Perplexity AI.

The use of such tools in this thesis is good to enhance the writing progress and, moreover, keep the coding structure as accurate as possible; however, it cannot always replace the author's own analysis (creative) thinking, process of thought and how he argues. Thus, all AI-generated text is either written by the thesis's author or modified/merged to create novel, correct and personalized content. Obeying the ethical guideline at University of Vaasa the writer/Author figures/drives on being responsible for submitted work which is found right concerning the origin information included and sources named. Mistakes and all inaccuracies, of course, are my own.