Fleet Design for Last-Mile On-Demand Logistics

Thesis Report

Cilia Claij

TU Delft

Fleet Design for Last-Mile On-Demand Logistics

Thesis Report

by

Cilia Claij

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Thursday December 15, 2022 at 13:00 AM.

Student number:	4596633	
Project duration:	November 1, 2021 – December 15, 2022	
Thesis committee:	Dr. J. Alonso-Mora ,	TU Delft, supervisor
	Dr. S. Sharif Azadeh,	TU Delft
	Dr. C. Hernandez Corbato	TU Delft
	Ir. M. Kronmueller,	TU Delft, daily supervisor

TU Delft

Abstract

The simultaneous rapidly increasing demand for home delivery of goods and on-demand expectancy of customers over the past years leaves a tough challenge for the logistical branch. They have to keep up with this increasing demand and simultaneously they are obliged to satisfy consumer service level demands to preserve their customers. On the other hand, as economic goals drive these businesses, they are prompted to operate cost-effectively. As a result, the fleet, deployed to execute the last-mile delivery, should meet both the requirement of cost-efficiency as well as the requirement for meeting consumer service level demands. This raises the question of how to efficiently design a fleet for last-mile on-demand logistics. For a fleet to be able to operate cost-efficiently, the fleet design decisions are required to take both fixed and variable costs into account. As such, the fleet design decisions need to include the consideration of the size of the fleet as well as the distance the vehicles travel on daily basis. Therefore, the goal of this thesis is to develop a novel method for fleet design for last-mile on-demand logistics. This work contributes by being the first to investigate methods for doing fleet design specifically for last-mile on-demand logistics considering multiple depots and variable pick-up locations. The purpose of the method is to determine the operational plans of the individual vehicles, the number of vehicles needed throughout a certain time period, the pick-up locations for all orders and the total distance travelled by the full fleet of vehicles. The proposed method builds upon established fleet design methods for ride-sharing taxi problems. The optimization method is adapted for last-mile on-demand logistics, yielding the required number of vehicles and their individual operational plans. The input of the system is a set of trips, which represent a path of a single vehicle to deliver one or multiple orders from a depot. Connecting two trips, which is called chaining, has the benefit of reducing the number of vehicles used, as chained trips are served by a single vehicle. Additionally, from multiple available depots where orders can be picked up, the method determines the best depot per order. This part of the method is called depot re-assignment. Furthermore, the fleet design problem is modelled as a multi-objective optimisation problem to find the trade-off between fleet size and the total distance the vehicles travel. Three different modelled datasets, each containing 10.000 order requests in the city centre of Amsterdam, are used to prove the value of the given method. A comparison between the method with and without depot re-assignment is made, to prove the value of the given addition of depot re-assignment. It is proven that depot re-assignment is valuable as it decreases or retains the fleet size for all test cases. The experiments conducted show that a significant decrease of the required fleet size can be established by a minor increase in total travelled distance. Furthermore, the optimal trade-off between the fleet size and the total distance travelled can be determined for a specific operation with the knowledge of operational costs for that operation.

Cilia Claij
Delft, December 2022

Acknowledgements

This thesis has been a journey of over a year with really valuable learning experiences as well as tough moments. A few months after starting the literature part of this thesis I suffered a major concussion, from which I only fully recovered about a month before my graduation. With the brain being the major workhorse for this thesis, this has been proven difficult to deal with. However, although (and because) the tough circumstances I worked with during my thesis, I am pride with the results I have been able to produce.

First and foremost, I want to thank the person who guided me along this journey the most, my daily supervisor Maximilian Kronmueller. He encouraged me to bring my work to the next level and provided me with very extensive and thorough feedback on my work. At the same time, he was available at all reasonable times for my questions and our weekly meetings have always benefited me. Apart from guiding me on the topic, he has also been a great help with his workflow suggestions and his encouragement to take a step back at the right moments in my thesis. Without his help, patience and encouragement, this work would not have been in the same state it is in now. Next, I want to thank Javier Alonso-Mora for providing me with the opportunity of doing this project. His view of my work and valuable and honest feedback has been useful in professionalising my writing. I would also like to thank my fellow students and the employees at AIRLab for providing a good work environment and also nice downtime during breaks and lunches.

Additionally, I want to thank Ilze for being my main support this year. Checking my plots, helping me format my work or giving feedback on my writing are a few of the many things she did for me. However, mainly just being able to talk to her during my thesis was the biggest help I could get. This helped me through some of the tougher moments I had during my thesis and kept me motivated to go on. I would also like to thank Maaike, Mariano, Roel, Daniel and Kitty for their help in reviewing my work and providing me with very useful feedback along the way. At last, I would like to thank my boyfriend, my family, other friends and my teammates from my futsal team for the massive support they gave me during my thesis, keeping me (mentally) fit and for providing me with the downtime I also needed.

Cilia Claij
Delft, December 2022

Contents

List of Figures

1 Introduction

Last-mile delivery is the term for describing the last step of the delivery of the order to the customer. With an increased demand for the delivery of goods, the demand for this last-mile delivery has been rapidly rising over the past few years. Accompanying the increasing demand for goods and widespread access to the internet, the online shopping market has become very prosperous. The convenience to save time and not having to go to the shops are among the reasons which justify the choice of customers to opt for online shopping instead of physically attending stores [2]. This has led to the described increased demand for goods delivery. Simultaneously, the demand for on-demand or same-day delivery has grown over the past years. A consumer survey showed an increase in interest in same-day delivery options from 33% in 2020 to 56% in 2022 [3]. The market is also predicted to have a compound annual growth rate of over 20% a year from 2021-2027 [4]. The increase of both the last-mile and on-demand delivery resulted in rapid combined growth over the last years and expected growth in the near future.

A particular branch benefiting from the increasing demand for last-mile on-demand deliveries is the grocery industry. Inherent to the daily need for edible products, the demand for the fast delivery of them exists. Within a short amount of time, fast delivery companies like Gorillas, Flink and Getir have established significant market share in the grocery market. As a result, larger companies partner with them to prevent major loss of market share to them [5, 6]. This market share is an important indicator of competitiveness and a higher market share reasonably leads to higher profit. It is therefore expected that companies prevent major loss of market share at all costs by the notion: "If you cannot beat them, join them".

Aiming for the best possible competitive positioning, companies thoroughly strive for the optimal cost-revenue balance of their products. For last-mile on-demand deliveries, the design of the fleet used to serve all order requests is essential for obtaining this cost-benefit balance. The design of the fleet determines both the delivery cost of the products, as well as the customer's satisfaction rate with the delivery: having a vehicle deficit causes the delivery times to be high. Because of this, promised service levels cannot be met, leading to unsatisfied customers. However, with a vehicle surplus, the asset use is inefficient. As a result, vehicles are idle too often, leading to higher costs than needed. These high costs cut the company's profit. Furthermore, the costs of operation do not only depend on the size of the fleet employed but depend on the variable costs of the fleet as well. With variable costs taking more than 1/3 of the total fleet-related costs [7], they cannot be disregarded. Variable costs such as fuel costs, maintenance costs and depreciation are directly related to the fleet mileage. The total mileage of the fleet is directly related to the fleet size, with a higher total mileage for a smaller fleet size [8]. As such, both fleet size and fleet mileage need to be taken into consideration for designing a fleet for cost-efficient operations. It can be concluded that prompted by the demand for cost-efficient operations and the requirement to uphold high customer service levels, the question of fleet design for last-mile on-demand delivery is important.

To accurately design a fleet for the last-mile on-demand logistics problem, it is expedient to take into account the individual vehicle plans. By considering the operational plans of individual vehicles over the course of an entire day and thus the capabilities of the individual vehicles to deliver the orders. Considering the information on individual vehicle plan level is important to operate cost-efficient, while simultaneously meeting promised service levels. Involving this information in the method ensures the orders are delivered within the promised delivery time without inefficient asset use.

This work proposes an optimization method for fleet design for last-mile on-demand logistics, yielding the required number of vehicles and the individual operational plans for each vehicle in the fleet. From multiple available depots where orders can be picked up, the method determines the optimal depot per order. The fleet design problem is modelled as a multi-objective optimisation problem to find the trade-off between fleet size and the total distance the vehicles travel.

1.1. Related Work

This section outlines a concise overview of the related works regarding fleet design for last-mile on-demand logistics. A more extensive overview of the related literature can be found in the Literature Survey constructed to support this thesis, which can be obtained from the author of this work upon request. Section 1.1.1 states the categorization and background of the Last-Mile On-Demand logistics problem. It is important to consider the nature of the problem to be able to incorporate the decision on individual vehicle level into the method. In Section 1.1.2, the fleet design problem is contextualized and the relationship between fleet size and fleet mileage is discussed. Section 1.1.3 elaborates on different methods which can be used for doing fleet design for the Last-mile on-demand logistics problem.

1.1.1. Last-Mile On-Demand Delivery Problem

The last-mile on-demand delivery problem can be categorized into the class of dynamic vehicle routing problems (DVRP). A dynamic vehicle routing problem is a variant of the vehicle routing problem (VRP), on which an extensive overview can be found in [9, 10]. The VRP is designed to find the set optimal/least-cost vehicle routes for a fleet of vehicles to service a set of customers, given a set of constraints. Each customer is visited exactly once by one vehicle, which starts and ends its route at a depot. In this original VRP, the vehicle routes are determined with all customer information available. In the DVRP, however, the routes are determined with the initial customer information available, and new requests can be placed during execution. To service these new requests, the vehicle routes need to be recalculated.

An example, made by Pillac et al. [1], of one vehicle in a dynamic vehicle routing problem (DVRP) can be found in Figure 1.1. When the vehicle departs at t_0 requests A, B, C, D and E are known. An algorithm calculates a solution to visit all initial requests. At t_1, when the vehicle has already visited A and B and is on its way to serve customer C, new requests X and Y enter the system. The current route plan is recalculated to include these new requests in the route plan. Instead of going from D to E to the depot, the new route plan now consists of going from D to Y to E to X to the depot. This real-time adaptation requires the recalculation of route plans for new requests. Additionally, in the context of our logistical problem, the vehicle needs to return to the depot before being able to serve customer Y. As the content of the order of customer Y is not known in advance, the vehicle must pick up the order before being able to serve the customer.

Considering city-scale operations, logically, multiple pick-up locations should be considered. Combining this with the dynamic nature of the on-demand delivery problem, the problem can be categorized as a dynamic multi-depot vehicle routing problem (DMDVRP). Due to the fact that every problem classified as a VRP is NP-Hard, in some cases, exact methods are mostly incapable of solving such a problem in finite time. However, different heuristic algorithms are used to approximate solutions for these kinds of problems. For DMDVRPs algorithms, such as ant colony optimization [11, 12], and particle swarm optimization [13] are used to generate these approximate solutions. Although they use different methods to solve the problem, the different works use a similar general strategy to obtain the final solution. By clustering the orders in regions with one depot, the DMDVRP problem is split into different DVRPs, which are solved using the heuristic methods described above. A more recent

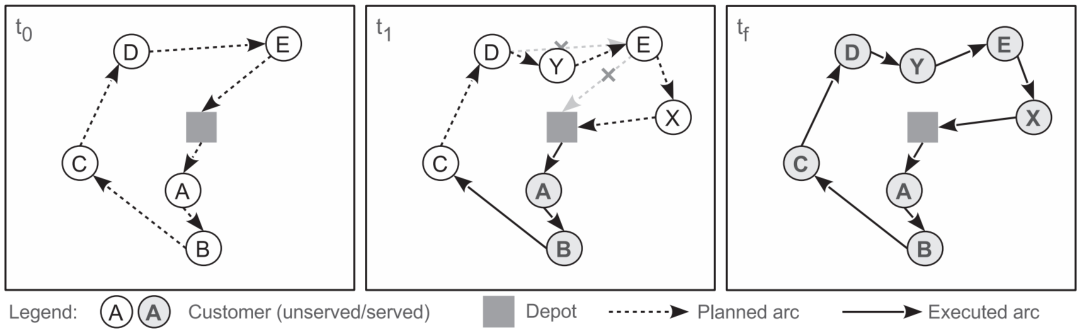


Figure 1.1: Diagram of one vehicle in a dynamic vehicle routing problem [1]

work by Kronmueller et al. [14] does not use this division into different DVRPs, but directly solves the DMDVRP problem at hand. The problem is solved by adapting the Vehicle-Group Assignment (VGA) method [15], which is originally designed to solve a dial-a-ride problem in the logistical context. The authors claim to be the first to consider multiple depots for a dynamic vehicle routing problem without decomposing it into sub-problems.

1.1.2. Fleet Design Background

According to Jones et al. [16], realistic fleet design problems include the choice from a variety of different vehicle types, purchase of new vehicles, maintenance and replacement. To make the problem more complex, facts about changing demand for transportation services over time and developments in vehicle technology influence the operating characteristics and economics regarding the fleet and the fleet design. Klosterhalfen et al. [17], designing a rail car fleet, split the fleet design problem into a fleet size problem and a fleet structure problem, focusing on the determination of the size of the fleet and the type of vehicles used, respectively. Different formulations are used to describe the fleet size problem and the fleet structure problem, such as fleet dimensioning problem [18] is used for fleet size and fleet capacity problem [19] is used as a synonym for the fleet structure problem. The combined problem is described as a fleet composition problem [18] or, in combination with the vehicle routing, as the Fleet Size and Mix Vehicle Routing Problem (FSMVRP) [20]. The essence of this problem is obtaining the number of vehicles and their size to accommodate demand at a minimal cost. It differs from the normal VRP as it chooses the number and capacities of the vehicles in the fleet, whereas a normal VRP assumes a fixed number of vehicles. By incorporating the minimization of the number of vehicles and their corresponding capacity into the objective function, the problem is solved [21]. All the different terminologies for a fleet design problem are not used consistently for the same problem and thus add to the complexity of defining the fleet design problem. The interpretation of fleet design used in this work mainly includes the decisions on fleet size and fleet mileage.

A few works consider the relationship between the fleet size and the amount of traffic/total distance travelled. Levin et al. [22] show this dependency for shared autonomous vehicles with and without dynamic ride-sharing. Dynamic ride-sharing enables the use of a car simultaneously with other users, whereas in the case without dynamic ride-sharing the vehicles would only allow one user at a time. Without dynamic ride-sharing, the relationship between the fleet size and the number of Vehicle Miles Travelled (VMT) was found: the more vehicles available to serve the demand, the less VMT, as fewer connection miles between customers have to be made. With the use of dynamic ride-sharing, an increase in VMT was found with a higher fleet size. The authors argue this occurred because less ride-sharing was used. For high fleet sizes, more significant than 14,500 vehicles, the same behaviour as without dynamic ride-sharing was found, however decreasing more rapidly with the number of vehicles. This is because dynamic ride-sharing is generally more efficient, which results in fewer miles travelled compared to not dynamically sharing rides. A similar conclusion is found in the work of Fagnant and Kockelman [23]. Sharing of vehicles induces excess VMT because of unoccupied vehicle re-locations, but ride-sharing also lowers the VMT. Therefore, with a high number of customers sharing their rides, the VMT is lower than without ride-sharing. It is important to note that the above-discussed works do either not include dynamic routing in their solution method, or do not include routing at all. As stated earlier in this chapter, incorporating the routing decision into the fleet design method is important to ensure that the orders are delivered within the promised delivery time without inefficient asset use.

1.1.3. Fleet Design Solution Methods

To accurately design a fleet for the last-mile on-demand logistics problem, as explained, it is expedient to take into account the individual vehicle plans. Considering the information on individual vehicle plan level is important to operate cost-efficient, while simultaneously meeting promised service levels. Involving this information in the method ensures that the orders are delivered within the promised delivery time without inefficient asset use. Therefore only solution methods incorporating the individual vehicle operation plans are considered suitable for fleet design for last-mile on-demand logistics.
The first work designs a Shared Automated Mobility-On-Demand system, focusing on two main performance metrics: quality of service (fast delivery to their destination) and operation cost (minimization of fleet size and total energy consumption) [24]. These metrics are stated to be in conflict, which means they cannot be minimized simultaneously. They propose a way to combine both objectives into an optimization problem to create trade-off curves, which can be used to make decisions about the fleet size. To solve this multi-objective problem, they use the Vehicle-Group Assignment method (VGA) as introduced by Alonso-Mora et al. [15]. The multi-objective replaces the objective function of the original VGA method, to apply the method to this particular problem. It finds the solution by generating all possible groups of requests that each vehicle can serve and then by finding an optimal assignment of such groups to individual vehicles.

Using the way of combining multiple objectives, Wallar et al. propose a better scalable method. The set of requests is divided into batches with a fixed batch size, which are iterated over backwards in time. For each batch, the VGA method is used to compute the set of travel schedules to pick up and drop off requests. The objective minimizes the sum of the delays of the passengers, the transition times between the current batch and the set of initial travel schedules (to connect schedules between different batches), and additive costs for ignoring requests and schedules from the previous batch. After the iteration of all requests is done, a maximum bipartite matching between travel schedules without outgoing transitions and the set of initial travel schedules is computed. This time a larger maximum idling time is used while ensuring the delay and waiting time constraints are still satisfied for all requests. This so-called long-term rebalancing reduces the total number of vehicles needed in the fleet. The objective of the Integer Linear Program (ILP) is to maximize the number of transitions, by which it minimizes the total number of vehicles needed in the fleet.
A continuation of this work does not only optimize the fleet size, but also the composition of the fleet with two capacity classes [25].

Similar to the previous work, the minimum trip cover, the smallest subset of all trips to serve all requests, is calculated using the method described in [15]. To determine the fleet size and composition, this minimum trip cover is used, by determining which trips can be executed in sequence by the same vehicle, without violating the quality of service constraints or letting the vehicle idle for too long between trips, and then the total vehicles needed is minimized. This sequencing problem is formulated as an ILP, with the objective function minimizing the weighted sum of the fleet size and the unused capacity of the vehicles.

Using the concept of trips, two works [26, 27] determine the fleet size slightly different. By splitting the construction of the trips and the determination of the fleet size, the fleet size is determined offline. Vazifeh et al. assume single orders as trips as input of their system and do not assume ride-sharing. Qu et al., on the other hand, focus mainly on the creation of the trips and do consider ride-sharing, allowing multiple orders per trip. While the interpretation of the trips themselves is different, the way of determining the fleet size is similar. They create a trip graph in which edges in the graph exist if both trips can be served by the same vehicle without violating constraints. From this graph, a maximum matching, corresponding to the minimum number of vehicles, is found by using the Hopcroft-Karp algorithm. This algorithm is focused on only finding the maximum matching, not taking into account other objectives.

Combining the notion of creating a multi-objective optimisation problem with the trip connection as proposed by Vazifeh et al. a method for fleet design for last-mile on-demand logistics is established. Instead of using the Hopcroft-Karp algorithm, a multi-objective optimisation is used to take both the fleet size and the fleet mileage into account. By considering trips as input of the method, the customer service levels are met using the routing method of Kronmueller et al. [14]. The ambiguity of the use of depots is covered with a novel method.

1.2. Contribution Statement

The contributions of this work regard the method designed to achieve fleet design for last-mile on-demand logistics. The main contribution regards the fact that this work is the first to investigate methods for doing fleet design specifically for last-mile on-demand logistics considering multiple depots and variable pick-up locations. By allowing the method to assign another depot to a trip, the fleet size can be decreased significantly. The method allows for operator decisions on the trade-off between the use of more vehicles or more driven kilometres. In addition to that, the method is scalable for scenarios with thousands of orders/trips. The method is evaluated on different datasets to prove value under varied conditions.

1.3. Thesis Structure

This thesis report is split into two main parts. Part I contains a scientific paper, proposing a condensed outline of the thesis work. The main concepts and results are considered in this first part. The second part of the report, Part II, serves as an extension of the paper in the first part. As the extended report builds upon concepts as discussed in Part I, it is advised against reading Part II before or without having read Part I. This extended report composes a more thorough explanation of some of the concepts addressed in the method section of the paper in Section 2. A more thorough background on the input data used is illustrated in Section 3. Section 4 reports and discusses an extended version of the experiments and results section of the paper. Last, a conclusion of the whole thesis and suggestions for future research are given in Section 5.

I

Paper: Fleet Design for Last-Mile On-Demand Logistics

Fleet Design for Last-Mile On-Demand Logistics*

Cilia Claij[1], Maximilian Kronmueller[1] and Javier Alonso-Mora[1]

***Abstract*— Keeping up with the rising demand for last-mile on-demand goods delivery, companies employ fleets for this purpose. Simultaneously, these companies are prompted to operate cost-efficient and satisfy consumer service level demands. Therefore, the goal of this paper is to develop a novel method for fleet design for last-mile on-demand logistics. This work contributes by being the first to investigate methods for doing fleet design specifically for last-mile on-demand logistics considering multiple depots and variable pick-up locations. The purpose of the method is to determine the operational plans of the individual vehicles, the number of vehicles needed throughout a certain time period, the pick-up locations for all orders and the total distance travelled by the full fleet of vehicles. The proposed method builds upon established fleet design methods for ride-sharing taxi problems. The input of the system is a set of trips, which represent a path of a single vehicle to deliver one or multiple orders from a depot. Connecting two trips, which is called chaining, has the benefit of reducing the number of vehicles used, as chained trips are served by a single vehicle. Additionally, from multiple available depots where orders can be picked up, the method determines the best depot per order. This method is referred to as depot re-assignment. Furthermore, the fleet design problem is modelled as a multi-objective optimisation problem to find the trade-off between fleet size and the total distance the vehicles travel. Three different modelled datasets containing 10.000 order requests in the city centre of Amsterdam are used to analyse the results of the given method. It is proven that depot re-assignment is valuable as it decreases or retains the fleet size for all test cases. A significant decrease of the required fleet size can be established by a minor increase in total travelled distance.**

I. Introduction

Last-mile on-demand delivery has been a rapidly growing branch over the last few years. Last-mile delivery, the last step in the delivery of the order to the customer, has been rising due to increased demand for the delivery of goods. Next to that, the demand for on-demand or same-day delivery has grown simultaneously, resulting in rapid combined growth. Fast delivery companies like Gorillas, Flink and Getir have established significant market share in the grocery market. As a result, larger companies partner with them to prevent major loss of market share [1], [2]. This market share is an important indicator of competitiveness and a higher market share reasonably leads to higher profit. Similarly, profit drives the demand for cost-efficient operations. For last-mile on-demand deliveries, the design of the fleet used to serve all order requests is essential for obtaining an optimal cost-revenue balance. Therefore, the research on designing efficient fleets for last-mile on-demand delivery is crucial. A vehicle deficit causes the delivery times to be high, and as a result, promised service levels cannot be met, leading to unsatisfied customers. With a vehicle surplus, the asset use is inefficient. As a result, vehicles are idle too often, leading to higher costs than needed which cut the company's profit. Additionally, the costs of operation do not only depend on the size of the fleet employed but depend on the variable costs of the fleet as well. With variable costs taking more than 1/3 of the total fleet-related costs [3], they cannot be disregarded. Variable costs such as fuel costs, maintenance costs and depreciation are directly related to the fleet mileage. The total mileage of the fleet is directly related to the fleet size, with a higher total mileage for a smaller fleet size [4]. As such, both fleet size and fleet mileage need to be taken into consideration for cost-efficient operations.

This work proposes an optimization method for fleet design for last-mile on-demand logistics, yielding the required number of vehicles. From multiple available depots where orders can be picked up, the method determines the optimal depot per order. The fleet design problem is modelled as a multi-objective optimisation problem to find the trade-off between fleet size and the total distance the vehicles travel. As can be seen in Figure 1, the choice of depot determines the distance the vehicle has to travel. It is assumed that every order will get delivered and therefore no orders are rejected.

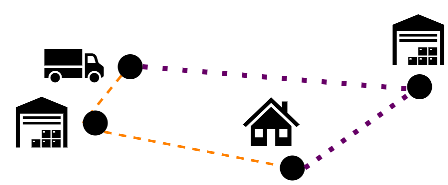


Fig. 1: Example of the importance of the choice of the depot. A vehicle in the top left has to deliver an order at the location at the bottom right. Even though the depot on the top right is closer to the request, following the purple dotted path is longer than using the depot on the bottom right and following the dashed orange path.

A. Related Work

The last-mile on-demand delivery problem is categorized as a dynamic vehicle routing problem (DVRP), as reviewed in [5], [6]. Finding the fleet size (and vehicle capacity) for a vehicle routing problem is called the fleet size and mix vehicle routing problem (FSMVRP) [7], which includes both finding the fleet size and vehicle capacity as well as the routes for the individual vehicles. To the knowledge of the authors of this article, no solution methods for finding the fleet size for a last-mile on-demand delivery problem exist.

*This work was supported by the AI for Retail Lab

[1]The authors are at the Department of Cognitive Robotics, Delft University of Technology, Mekelweg 2, 2628CD Delft, Netherlands {C.Claij, M.Kronmuller, J.AlonsoMora} @tudelft.nl

However, allowing ridesharing for the Dial-a-Ride problem [8], the problem becomes similar to a Last-Mile On-Demand Delivery problem. The Vehicle-Group Assignment Method (VGA) of Alonso-Mora et al. [9] allows for ridesharing of more than two passengers and finds the optimal assignment to requests for a given vehicle fleet. Using the VGA method, Kronmueller et al. [10] solve a last-mile on-demand logistics problem with multiple depots and elaborate on the main difference between the two problems (Dial-a-Ride vs Last-Mile On-Demand Logistics): for a logistical context, the pick-up locations are undefined.

For the Dial-a-Ride problem, different methods to determine the fleet size exist. Focusing on obtaining the minimum fleet size, Vazifeh et al. [11] developed a method to find the minimum fleet size by connecting taxi trips, and picking and dropping off individuals. This method, however, does not assume ride sharing, making it inapplicable to logistics on its own. Cap and Alonso-Mora [12] model a multi-objective fleet routing problem to compute Pareto-optimal fleet operation plans which resemble the trade-off between quality of service and operation cost (fleet size). The method, however, due to computational heaviness, could only be evaluated with one minute of taxi demand data. Wallar et al. [13] overcome this problem by presenting a method to determine the fleet size with a month of demand data while allowing ridesharing. In a later work [14] they improved upon this method by allowing different vehicle capacities, where the method determines the capacity of the vehicles and fleet size to meet the demand. The latest work of Cap et al. [15] introduces a method using the concepts of passengers travelling with a vehicle simultaneously, called pooling, and of passengers sharing the same car over the course of the day. This enhances the method over the method by Vazifeh et al. [11], by allowing ridesharing.

This article builds upon the method for the Dial-a-Ride problem of Cap et al. [15], using their chaining method. The concept of trips as an input of Vazifeh et al. [11] is used, but instead of transporting single passengers (in this case packages), the trips are pooled orders. This allows multiple packages to be transported simultaneously. The main difference between the two problems (the Dial-a-Ride problem and the last-mile on-demand delivery problem) is the pick-up location of an order. In the Dial-a-Ride problem, a single location for passenger pick-up is used. On the other hand, the depot to pick up a product is ambiguous in the last-mile on-demand delivery problem. Therefore, this work proposes an extension to the method of [15], to determine the fleet size, the vehicle routes and the order pick-up locations.

B. Contribution Statement

This work contributes by being the first to investigate methods for fleet design specifically for last-mile on-demand logistics considering multiple depots and variable pick-up locations. By allowing the method to assign another depot to a trip, the fleet size can be decreased significantly. The method allows for operator decisions on the trade-off between the use of more vehicles or more driven kilometres. In addition to that, the method is scalable for scenarios with thousands of orders. The method is evaluated on different data sets to analyse it under varied conditions.

II. Problem Formulation

Consider a weighted directed graph $G = (N, A)$, N being a set of nodes and A defining a set of weighted arcs. These weights represent the traveling times between two connected nodes. A depot $d \in D$ is a special node which serves as a pickup location for goods. Therefore, all depots conform to $D \subset N$. It is assumed that the goods inventory of the depots is always sufficient for pickup. The distance between two nodes $(n_1, n_2 \in N)$ on the graph is defined by $\tau(n_1, n_2)$. This distance is defined as the smallest sum of arc weights of all sets of arcs that span a path between the two nodes.

The demand is given by a set of orders R. A single order $r \in R$ represents the request of one customer to receive their order on location n at time t. Orders are incorporated into pooled trips P, which serve as the input to the trip chaining method. A trip always consists of one pick-up location, which is the start of the trip n^{start}, and one or multiple orders $O \subset R$. These orders O are pooled together into trips as decribed by Kronmueller et al. [10]. However, in this case, trips are not assigned to a limited number of vehicles, rather each trip represents a potential vehicle v. These vehicles are potential as, upon the connection of two trips, one potential vehicle is eliminated and the other is used to serve all orders of both trips.

A trip $p = (v, n^{start}, n^{end}, t^{start}, t^{end}, L) \in P$, represents a potential vehicle v which can be used to serve all orders of the trip in time. The set L holds the locations n of a set $O \subset R$. v represents the potential vehicle created to serve the orders O, with locations L, of the trip. The trip starts at location n^{start} at time t^{start} and ends at location n^{end} at time t^{end}. The starting location of a trip, n^{start}, is the depot which is closest to the first order of the trip. This depot is chosen initially as when the vehicle starts at the depot, this is the closest and therefore the best depot choice for a trip to start. However, the pick-up location could change without causing any inconvenience to the customer and we allow it to be changed if a better pick-up location exists. This could occur if an existing vehicle, which previously delivered other requests, has to deliver this order and therefore does not start at the given depot.

Given an operational environment G and a set of pooled trips P, the fleet design problem can be described as follows: "Find a set of operation plans Ω, defining the number of vehicles V needed to serve all trips P, the corresponding pick-up locations of all orders of each trip and the total distance T the vehicles travel, given that a vehicle $v \in V$ is available at each n^{start} on or before t^{start}." The goal of this study, therefore, is to determine the fleet design while considering the operational costs. This results in a multi-objective function to minimize the cost of operation, as both the cost of vehicles as well as the cost of distance travelled are considered. This multi-objective function is described as:

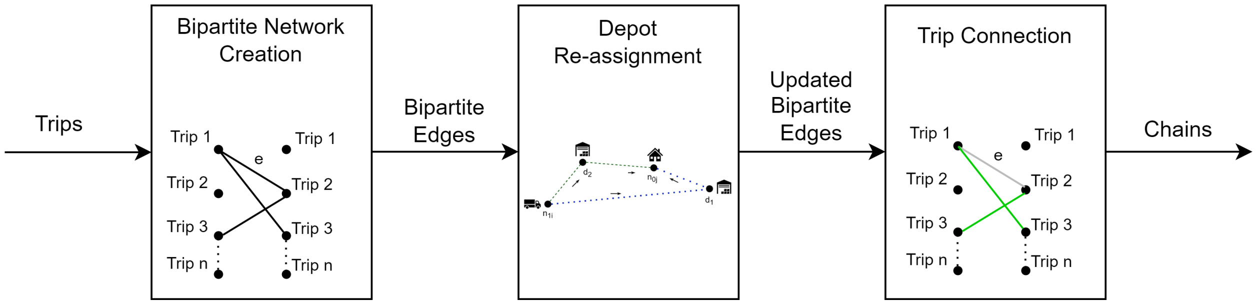


Fig. 2: Overview of the trip chaining method. During the bipartite network creation, all potential connections between trips are found. The depot re-assignment step assigns a new depot to a trip if this decreases the total distance travelled. The trip connection step solves the bipartite matching problem to find the best bipartite edge selection. The output of the system is a set of chained trips.

$$\min_{\Omega} \quad \beta \cdot c^{rej}_{total}(\Omega) + (1-\beta) \cdot \rho(\Omega). \tag{1}$$

The c^{rej} describes the rejection cost of the use of a vehicle, so the cost the operator needs to pay to start an additional vehicle for a trip. Its total sum $c^{rej}_{total}(\Omega)$ depends on the number of vehicles used. The ρ describes the total distance travelled, i.e. the sum of all travelled distances of all vehicles during the whole day following their corresponding plan in Ω. As both the fleet size and the distance travelled need to be taken into account, the objective function is defined as a combination of both objectives to determine the fleet size as the number of vehicles V used. The weight factor β controls the focus of the objective. For $\beta = 0$ the total distance travelled ρ is minimised and for $\beta = 1$ the focus is on minimising fleet size, while potentially increasing the distance travelled significantly.

III. Method

Every trip $p \in P$ resembles a potential vehicle v to deliver the respective orders O. However, a vehicle can serve another trip, after finishing the previous one, when it can reach the starting location n^{start} of the next trip before or at it's starting time t^{start}. This idea of connecting trips is often referred to as chaining. By chaining two trips, the number of potential vehicles to serve all trips is reduced by one. The connection of two trips is defined as a chain. Finding chains for all trips results in a set of chains. The number of vehicles used is represented by the number of chains, created from the trips. Therefore, the method to solve the fleet problem, as defined in Section II, is to determine in which way the individual trips can be chained together. This results in a set of operation plans Ω while considering the time constraints, $t^{end}_0 + \tau(n_0, n_1) \leq t^{start}_1$, and the distance travelled.
A rough outline of the proposed method, as illustrated in Figure 2, is given in this paragraph. A detailed explanation of the method can be found later on in this section. In the first step, all feasible potential connections for all pairs of trips are computed. The depot re-assignment step uses these connections to determine the best depot to use for these combinations of trips. This decision is dependent on both trips of a potential connection and is therefore conducted for every potential connection of trips. The edges and trips are then updated according to the outcome of the depot re-assignment step. The updated potential connections are finally used in the trip connection step to determine the best edge selection to connect the trips. The output of the trip connection step is a set of chained trips, where the number of chains specifies the number of vehicles needed. The trips now contain the right pick-up location, according to the chain they are included in.

A. Bipartite Network Creation

Each trip can be chained to all other trips that satisfy the time constraints of the trips as described above. To determine the feasible connections, we model a bipartite graph $B = (U, W, E)$. A bipartite graph consists of two rows of nodes, U and W, with edges E as connections between these nodes. Every trip is represented by a node $u \in U$ and $w \in W$ on either side of the graph. If a feasible chain between two trips i and j exists, an edge $e \in E$ exists between their corresponding nodes. The cost of such an edge is used to determine whether it is beneficial to make the connection between the two trips i and j, or deploy a new vehicle for trip j. This is dependent on the cost of deploying a new vehicle, as well as the cost to transfer a vehicle from trip i to trip j. Therefore, the edge cost between vertices i and j is described as:

$$c_{ij} = -c^{rej} * \beta + (1-\beta) * \Big[\tau(n^{end}_i, n^{start}_j) + \gamma * \tau(n^{start}_j, l_1)\Big]. \tag{2}$$

The cost of transferring a vehicle to the start of trip j is given in between the square brackets. The chain distance, $(\tau(n^{end}_i, n^{start}_j))$, describes the length of the edge connecting trips i and j. However, only including this length is not sufficient. With the starting location of a trip j being ambiguous, the distance from the depot to the first request depends on the depot that is used, as can be seen in Figure 3. It is, therefore, expedient to include this distance in the edge cost, to simultaneously determine the best depots and the best trip connections. As including this distance in the cost function does influence both the fleet size and the total distance, another weight factor, γ, is introduced. This weight factor controls how much of this distance (from the depot to the first request) is incorporated into the total edge cost. The extent of this inclusion, and therefore the knowledge of the cost-optimal solution, is with the system operator. Adding a weight factor facilitates the decision-making process of the operator and thus the finding of the optimal combination of total distance travelled and fleet size for that specific operation. The addition of the weight factor changes the costs of the edges. As these costs are the foundation of the decision later on in the method, this leads to different chaining options and, therefore, different solutions (different fleet sizes and total travelled distance). With $\gamma = 0$, the focus of depot re-assignment is purely on decreasing the fleet size. With $\gamma = 1$, the distance between the depot and the first request is included in the objective. As this distance is increased with depot re-assignment, the focus of the objective shifts to minimizing the total distance travelled.

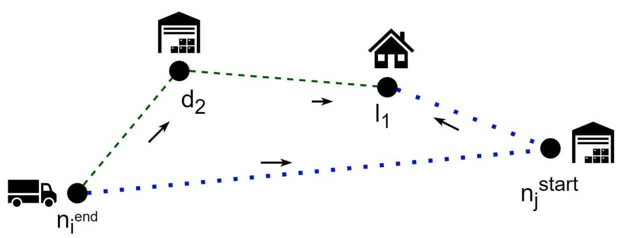


Fig. 3: Illustration of depot re-assignment. A chain between a trip i and a trip j, connects the last request of trip i, n_i^{end}, and the starting location (depot) of trip j, n_j^{start}. Another path using depot d_2, however, does exist, which results in an overall shorter distance to collect the item from this depot and deliver it to the order location l_1. It can be seen that the new route, the dashed green line, is overall shorter than the old connection, the dotted blue line.

B. Depot Re-assignment

Every trip consists of a depot and a minimum of one order. The depot initially selected to deliver the order(s) of a trip will always be the one closest to the first order. However, this does not mean this depot is the most efficient one after chaining trips together. In the example shown in Figure 3, depot n_j^{start} is the depot originally assigned as the pick-up location for the orders of trip j, with the first order location being l_1. However, as trip 1 is chained to the last order of trip 0, n_0^{end}, the total driven distance would be shorter when using depot d_2 instead of the initially assigned depot n_1^{start}. It is, therefore, expedient to check for every potential trip connection whether another depot can be assigned. This is checked for all depots, using the inequality:

$$\tau(n_i^{end}, d_2) + \tau(d_2, l_1) < \tau(n_i^{end}, n_j^{start}) + \tau(n_j^{start}, l_1). \tag{3}$$

That is to say, if there is a depot for which Equation (3) holds, the depot of the corresponding trip is re-assigned from n_j^{start} to d_2. If multiple depots satisfy the equation, the one with the lowest $\tau(n_i^{end}, d_2) + \tau(d_2, l_1)$ is used. The trip will be updated with the new depot and the bipartite edges $e \in E$ for which this re-assignment occurs, have their cost updated according to the new depot. The re-assignment of a depot changes both the chain distance $\tau(n^{end}, n^{start})$ as well as the distance from the depot to the first request $\tau(n^{start}, l_1)$. As the original depot is the closest to the order, the distance from the depot will increase. Hence, the chain distance has to decrease with a re-assignment (due to Equation (3)). With a shorter connection distance, it becomes more beneficial to connect trips instead of using separate vehicles. Depot re-assignment itself, therefore, affects the fleet size and the total distance travelled, and is controlled by the weight factor γ.

C. Trip Connection Selection

To find the chains to be used, the optimal edge selection of the bipartite network has to be determined. This problem can be modelled as an integer linear program and solved using available solvers. The objective function has to minimize the cost of the edges and therefore takes into account the cost of operating an extra vehicle or the distance to be travelled.

As the time constraints of all trip connections are already accounted for in the edge creation of the bipartite network in III-A, the optimisation problem, modelled as an integer linear program, for trip connection becomes as follows:

Trip Connection Problem Given the edges E of the bipartite graph, solve:

$$\min_{i,j \in E} \sum c_{ij} * x_{ij} \tag{4a}$$

subject to

$$\sum_{j \in W} x_{ij} \leq 1 \qquad \forall i \in U \tag{4b}$$

$$\sum_{j \in U} x_{ji} \leq 1 \qquad \forall i \in W. \tag{4c}$$

The integer optimisation variable $x_{ij} \in \{0, 1\}$ takes the value of 1 if the edge e_{ij} between bipartite nodes i and j is selected and 0 otherwise. Practically a value of $x_{ij} = 1$ means the chaining of trips i and j, which will be served by the same vehicle. The objective is constrained by the fact that every node can only have one incoming (Equation (4b)) and one outgoing (Equation (4c)) connection as each vehicle can only serve one trip at a time.

To solve this problem in polynomial time, a Hungarian Algorithm [16] can be used. The problem can also be solved with available solvers such as Gurobi [17] or Mosek [18].

Theoretically, this method can be used to solve for all trips and thus consider all potential connections. However, if the problem is too large, considering all potential connections results in too many optimization variables to be solved efficiently. Therefore, to solve a large problem, a heuristic can be applied. Instead of considering all trips and potential connections, the trips are divided into batches for which the chains are calculated separately. On top of that, an extra iteration is done to chain in between different batches. Trips which are the first or last of a chain and thus have no trip connected in front or after them are candidates for chaining to trips with the same conditions as other batches. Similarly, trips which are not part of any formed chain and therefore have no trips connected in front or after them are also candidates for inter-batch chaining. This allows for long chains to be made and, thus, vehicles to be used over the course of the entire day. The output of the system is a set of chained trips. Concatenation of the connected trips provides full chains which correspond to the route one vehicle has to drive. These chains are the set of operation plans Ω, whereas the number of these chains represents the number of vehicles used for the whole operation.

IV. Experiments and Results

The methodology is evaluated on three generated datasets, simulating a potential day of order requests in the city centre of Amsterdam. A directed graph of 2717 nodes and 5632 edges represents the city. The depots are distributed onto this graph following a k-center algorithm. The travel distance between two nodes is calculated as the distance in meters divided by a constant speed of 36 km/h. A set of orders is created with a uniform spacial distribution and a temporal distribution, including a noon and evening peak. These request orders are pooled into trips , which are then used by the method as described. Three different datasets, with ten thousand orders each, are used to produce the average results as shown in this section.

To evaluate the proposed depot re-assignment method, a comparison to the same method without re-assignment is made. From here on the method without re-assignment is called the base method. The average results over all datasets for both methods over a variety of β-values are shown in Figure 4, displaying the effect on the fleet size and the total distance travelled. The logarithmic value of the fleet size is plotted to show the trend in fleet size and simultaneously visualising the difference between the two methods. For $\beta = 0$, the distance travelled is minimised, and for $\beta = 1$ the focus is on minimising the fleet size. Both effects clearly stand out in Figure 4. For all values of β, the fleet size is either smaller or equal for the method with re-assignment, compared to the base method. A saturation effect can be noticed for high β-values. This occurs because the chosen value for c^{rej} multiplied by β is higher than the value of the distances (multiplied by β) of the right side of the edge cost, Equation (2), for high values of β. As the value for c_{rej} is significantly higher than the potential chaining distance, in (almost) all cases, connecting two trips is considered a better option than deploying a new car. This effect occurs for β-values close to and 1.0.

However, in combination with the chosen c_{rej}, this effect already emerges for $\beta = 0.5$, seen in the fact that this fleet size is almost the same value as for $\beta = 1.0$. It was determined that for β values of 0.5 and higher, at peak times 108 out of 112 vehicles were used, achieving a fleet utilization of 98%. The total distance travelled can either increase or decrease compared to the base method due to depot re-assignment, as can be seen for β-values 0.25 and 1.0, respectively. This effect occurs as re-assignment

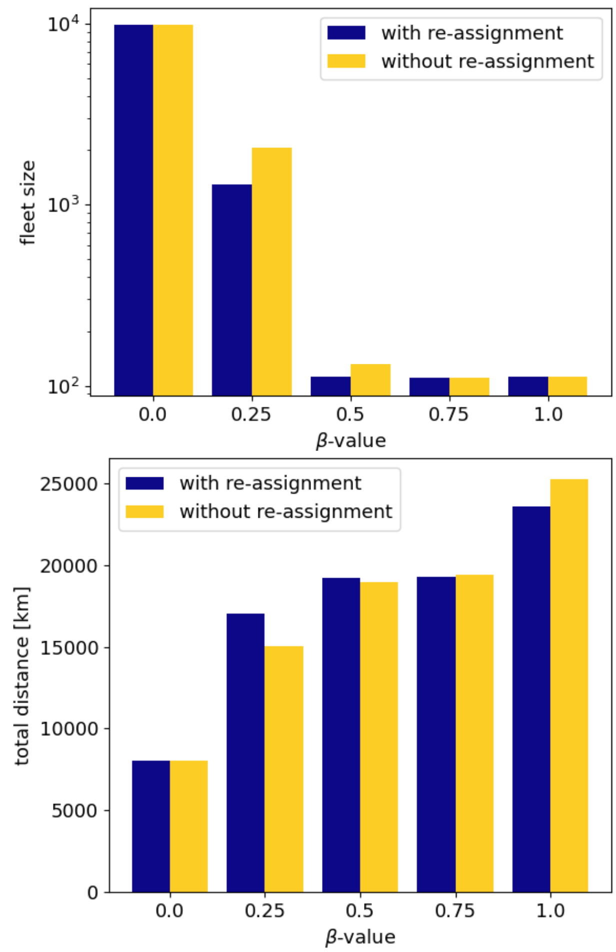


Fig. 4: Average results for different values of β. **Top:** The logarithmic value of the fleet size for the method with re-assignment and the base method. **Bottom:** The total distance travelled in kilometres for both methods.

decreases the length of the connection between the two trips, and increases the path from the depot to the first request. For a value of $\gamma = 0.5$, the increase from the depot to the first request is not fully incorporated into the objective function, thus making it possible to surpass the decrease in chain length. This leads to an overall increase in the total distance travelled. Interestingly this is dependent on the value of β. For $\beta = 0.25$, the base method outperforms the method with re-assignment with regard to the total travelled distance. The difference between the methods becomes minor for $\beta = 0.5$, whereas for $\beta = 0.75$, the method with re-assignment outperforms the base method slightly. For $\beta = 1.0$, the method with re-assignment clearly outperforms the base method with regards to the total distance travelled. Thus, it can be concluded that re-assignment lowers the fleet size for the cost of increasing the total distance for low β-values. For high β-values, the method does not affect the fleet size, but does decrease the total distance.

As depot re-assignment influences both the fleet size and the total distance travelled, the decision variable γ is introduced.

To study the effect of the value γ, a sensitivity analysis for this variable has been conducted. Results for various γ values (Equation (4)) in the range of 0 to 1 in terms of fleet size and travelled distance are visualized in Figure 5. The figure shows that a significant trade-off between the total distance travelled and fleet size exists.

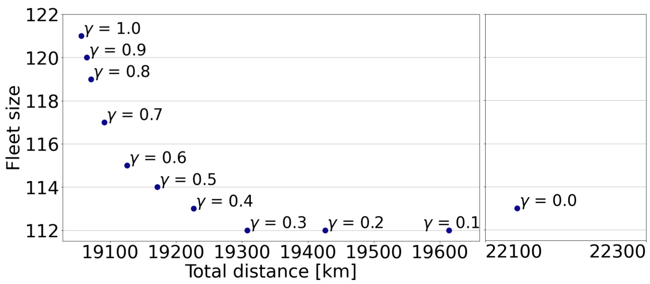


Fig. 5: Average fleet size compared to the average total distance travelled for different values of γ.

Fully including the distance from the depot results in the highest fleet size, but on the other hand, the lowest total distance travelled. For a value of $\gamma = 0$, the opposite is true: it results in one of the lowest fleet sizes, and the highest total distance travelled. The other values can be found in between these. The fact that the fleet size for $\gamma = 0$ is 1 higher than for $\gamma = 0.1$ is most likely due to an edge with a cost value of 0 among the connected trips. The increase of γ from 0 to 0.1 has a significant impact on the distance travelled with no increase in fleet size. However, for higher values of γ, ranging from 0.7 - 1.0, this impact on the total distance is almost negligible concerning the increase in fleet size it causes. This graph can be used for operator decision-making, to which the value of travelled kilometres and the use of extra cars are known. It can be used to find for which value of γ the operation runs the most cost-efficient.

V. Conclusion

This work presented an optimization method for fleet design for last-mile on-demand logistics, yielding the required number of vehicles. Given a graph and a set of trips, the method determined the fleet size, the order's pick-up locations and the total distance travelled for each order concerning a last-mile on-demand logistics problem. This is done by determining the chaining of trips, by solving a multi-objective problem with an ILP. Chaining two trips has the benefit of reducing the number of vehicles used, as chained trips are served by a single vehicle. The output of the method is the number of vehicles needed to serve all trips, the individual operational plans of these vehicles, the pick-up locations of all requests and the total distance all vehicles travel. Three datasets, each containing 10,000 order requests in the city centre of Amsterdam, were used to analyse the results of the given method. It is shown that the method is proven to be effective in determining the fleet size for a last-mile on-demand logistics problem, with a fleet utilization of 98%. On top of that, a significant decrease of the required fleet size was established by a minor increase in total travelled distance. The optimal trade-off between the fleet size and the total distance travelled can be determined for a specific operation with the knowledge of operational costs for that operation. Future research could include the addition of depot re-assignment for infeasible edges of the bipartite network. As depot re-assignment shortens the distance to the first order of a trip, connections could become feasible because of that. In addition, with the shorter distance to the first order of a trip, the total delivery time of all orders of a trip decreases. As such, future research could determine the effect of allowing a variable pick-up time according to possible depots. Moreover, this study only includes experiments with generated datasets. Additionally, future research case studies on real-life data and/or different cities can be done.

References

[1] fsgjournal, "A bottle of wine in 10 minutes? - the growth of on demand delivery." https://fsgjournal.nl/article/2022-01-11-a-bottle-of-wine-in-10-minutes-the-growth-of-on-demand-delivery, 2022. Accessed: 2022-09-09.

[2] de econometrist, "The rise of the on-demand grocery delivery industry." https://www.deeconometrist.nl/economy/the-rise-of-the-on-demand-grocery-delivery-industry/, 2022. Accessed: 2022-09-09.

[3] P. M. Bösch, F. Becker, H. Becker, and K. W. Axhausen, "Cost-based analysis of autonomous mobility services," *Transport Policy*, vol. 64, pp. 76–91, 2018.

[4] S. Hörl, C. Ruch, F. Becker, E. Frazzoli, and K. W. Axhausen, "Fleet control algorithms for automated mobility: A simulation assessment for zurich," in *2018 TRB Annual Meeting Online*, pp. 18–02171, Transportation Research Board, 2018.

[5] V. Pillac, M. Gendreau, C. Guéret, and A. L. Medaglia, "A review of dynamic vehicle routing problems," *European Journal of Operational Research*, vol. 225, no. 1, pp. 1–11, 2013.

[6] H. N. Psaraftis, M. Wen, and C. A. Kontovas, "Dynamic vehicle routing problems: Three decades and counting," *Networks*, vol. 67, no. 1, pp. 3–31, 2016.

[7] B. Golden, A. Assad, L. Levy, and F. Gheysens, "The fleet size and mix vehicle routing problem," *Computers & Operations Research*, vol. 11, no. 1, pp. 49–66, 1984.

[8] J.-F. Cordeau and G. Laporte, "The dial-a-ride problem: models and algorithms," *Annals of operations research*, vol. 153, pp. 29–46, 2007.

[9] J. Alonso-Mora, S. Samaranayake, A. Wallar, E. Frazzoli, and D. Rus, "On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment," *Proceedings of the National Academy of Sciences*, vol. 114, no. 3, pp. 462–467, 2017.

[10] M. Kronmueller, A. Fielbaum, and J. Alonso-Mora, "Automated last-mile on-demand logistics from multiple depots." 2022.

[11] M. M. Vazifeh, P. Santi, G. Resta, S. H. Strogatz, and C. Ratti, "Addressing the minimum fleet problem in on-demand urban mobility," *Nature*, vol. 557, no. 7706, pp. 534–538, 2018.

[12] M. Cáp and J. A. Mora, "Multi-objective analysis of ridesharing in automated mobility-on-demand," in *RSS 2018: Robotics-Science and Systems XIV*, 2018.

[13] A. Wallar, J. Alonso-Mora, and D. Rus, "Optimizing vehicle distributions and fleet sizes for shared mobility-on-demand," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 3853–3859, IEEE, 2019.

[14] A. Wallar, W. Schwarting, J. Alonso-Mora, and D. Rus, "Optimizing multi-class fleet compositions for shared mobility-as-a-service," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 2998–3005, IEEE, 2019.

[15] M. Cap, M. Kronmueller, I. Medrano, R. Ruigrok, A. Fielbaum, and J. Alonso-Mora, "Quantifying the design trade-offs of shared mobility-on-demand systems." 2022.

[16] J. Munkres, "Algorithms for the assignment and transportation problems," *Journal of the society for industrial and applied mathematics*, vol. 5, no. 1, pp. 32–38, 1957.

[17] G. Optimization, "Gurobi for c++."

[18] M. ApS, "Mosek for c++."

II

Extended Report

2

Additional Information on Method

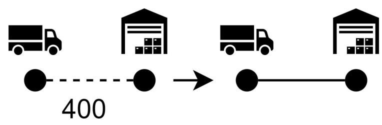
The basics of the method to solve the fleet design problem are described in the method section of Part I of this report. In this extended method section, several concepts of the description in Part I, which are in need of a more thorough elaboration, are revisited with an in-depth explanation and illustration. Section 2.1 describes the effect of chaining and the influence of the value of c^{rej} on this. Section 2.3 illustrates the need for the γ value, and it's effect on the bipartite network and corresponding trip connection. The next section, Section 2.2, outlines the choice of method to find alternative depots within the depot re-assignment step. Section 2.4 illustrates the method-wise difference between solving the trip connection step of the method as an Integer Linear Program or with a Hungarian Algorithm. The last section of this chapter, Section 2.5, contains the pseudo-code of the method, describing the algorithms used to get from the input of trips to the individual vehicle plans.

2.1. The Purpose of Connecting Trips

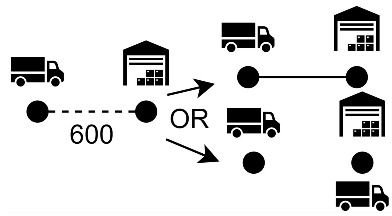
The trip connection step uses all potential chaining connections to select the optimal connection between the trips. Recall, these potential connections are the connections as created with the bipartite network. The edge cost as described in Equation 2 in Part I of this report is used to compare the different potential chaining connections. The most simplified version of a set of bipartite edges is a set consisting of only one potential edge. Such a set is used in this section to demonstrate the effect of chaining with respect to the effect of no chaining. Considering a set with one edge and decision variable values $c^{rej} = 600$, $\beta = 0.5$, $\gamma = 0$, three different scenarios can occur as displayed in Figure 2.1. Every scenario consists of a potential trip connection with a certain (edge) cost. Dependent on this cost, the situation following the arrow happens. This is either a chaining or the deployment of a new vehicle. With an edge cost lower than the $c^{rej} = 600$, the first scenario occurs as depicted in Figure 2.1a. For these values, making the chaining connection is beneficial according to the objective function. As the value of the connection is 400 and the decision variables are as described, the edge cost has a value of -100 ($-600 \cdot 0.5 + 400 \cdot 0.5 = -100$). Comparing this to the option of no chaining, which does not add value to the objective, it is beneficial for the chaining to take place. Because of this, only one vehicle is needed to serve both trips. An edge cost value larger than the $c^{rej} = 600$ results in the third scenario, visualised in Figure 2.1c. As the cost of connecting the two trips is larger than the cost of deploying a new vehicle, chaining is not beneficial in this case. As the total edge cost is now positive ($-600 \cdot 0.5 + 800 \cdot 0.5 = 100$), the option of not chaining will result in a lower objective. Therefore, the chaining does not occur and a new vehicle is deployed to serve the second trip. For an edge cost value equal to the $c^{rej} = 600$, the outcome is indifferent, as shown in Figure 2.1b. The cost to deploy a new vehicle to serve the second trip is exactly the same as the cost of transferring the existing vehicle to the second trip's starting location, resulting in an objective value of 0 ($-600 \cdot 0.5 + 600 \cdot 0.5 = 0$). In this work, as the cost is equal, either of the outcomes is selected by the solver and are considered equivalent.

2.2. Depot Selection for Depot Re-Assignment

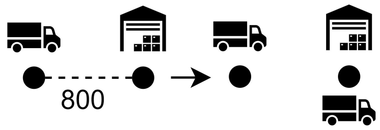
The depot re-assignment step determines which depot is the best to pick up the orders of a trip after delivering the orders of another trip. Equation (3) of Part I is used to determine whether a depot is



(a) Chaining occurs



(b) Indifferent



(c) Chaining does not occur

Figure 2.1: Three different examples of whether chaining occurs with certain distance values. Every scenario consists of a potential trip connection with a certain (edge) cost. Dependent on this cost, the situation following the arrow happens, which is either a chaining or the deployment of a new vehicle. Decision variable values are $c^{rej} = 600$, $\beta = 0.5$, $\gamma = 0$

strictly better than the depot initially assigned to the trip. The equation is used for all depots and the depot with the lowest sum of distances is used as the new depot. This results in the pseudo-code given in Algorithm 1.

Algorithm 1: Depot re-assignment method

$fastest \leftarrow \tau(n_i^{end}, n_j^{start}) + \tau(n_j^{start}, l_1)$;
`// Find re-assignment depot in set of all depots`
for $d \in D$ **do**
  if $\tau(n_i^{end}, d) + \tau(d, l_1) < fastest$ **then**
    $fastest \leftarrow \tau(n_i^{end}, d) + \tau(d, l_1)$;
    $new\_depot \leftarrow d$;
  end
end

For the number of depots used in this work, the method as described in Algorithm 1 is sufficient with respect to computational time. As the computational time linearly scales with the number of depots, an alteration of the method could be necessary for a high number of depots. A first suggestion on improving this is illustrated in Figure 2.2. To determine which depot is the best, first an estimate of applicable depots can be made. In Figure 2.2, the distance from the first trip to the depot of the second trip (connection distance) is visualised in orange. The distance from the depot to the first request of the second trip is visualised in pink. A depot is strictly better than the current depot when it shortens the total distance from the last request of the first trip via the depot to the first request of the second trip. This means that all depots lying within this total distance from both the last request of the first trip and the first request of the second trip, can be qualified as potentially applicable for re-assignment. The set of potentially applicable depots can be found in the overlapping region of the two circles in Figure 2.2. From the set of applicable depots, the strictly better depots can be determined, by checking if depots in this subset decrease the distance to and from the depot as described in Equation (3) of Part I. Recall, a depot is strictly better than another depot if the total distance from the last request of the first trip via the depot to the first request of the second trip is lower for that depot. Using this method of doing depot re-assignment, thus, first requires the determination of the set of potentially applicable depots after which this set is used to determine whether another depot is strictly better than the other depots of the set and the originally assigned depot. This method is described in Algorithm 2.

However, as the first step of the algorithm requires looping over all depots, Algorithm 2 is not better regarding computational time. In fact as this algorithm requires a second step to loop over the subset of depots, the computational time has increased. Therefore, the method as proposed in Algorithm 2 is more complex and computationally worse than Algorithm 1, so not beneficial to use.

Algorithm 2: Find depot re-assignment in applicable depots zone

$radius \leftarrow \tau(n_i^{end}, n_j^{start}) + \tau(n_j^{start}, l_1)$;
$D^* \leftarrow []$;
`// find potentially applicable depots`
for $d \in D$ **do**
    if $\tau(n_i^{end}, d)) < radius$ & $\tau(d, l_1) < radius$ **then**
        append d to D^*;
    end
end
$fastest \leftarrow radius$;
$new\_depot \leftarrow 0$;
`// Find re-assignment depot in potentially applicable set of depots`
for $d \in D^*$ **do**
    if $\tau(n_i^{end}, d) + \tau(d, l_1) < fastest$ **then**
        $fastest \leftarrow \tau(n_i^{end}, d) + \tau(d, l_1)$;
        $new\_depot \leftarrow d$;
    end
end

However, the method in Algorithm 2 could be slightly altered to fit the purpose. This proposal is described in Algorithm 3. Instead of looping over every node on the graph, the distance to all depots can be predetermined and saved. These distances can be sorted from shortest to longest, for each node. Another look-up table contains the distances from the depots to every node on the graph. In the depot re-assignment step, only the depots up to the depot with a larger distance than the distances of the current depot will be considered. This differs from the method described above as the depots that are too far from the requests are now discarded before doing any calculations instead of afterwards. The second part of the algorithm is identical to the second part as described in Algorithm 2.

Algorithm 3: Find subset of Depots

import look_up_dis_to_depot;
import look_up_dis_from_depot;
$D^* \leftarrow []$;
`// find potentially applicable depots`
for $d \in D$ **do**
    if $look\_up\_dis\_to\_depot < \tau(n_i^{end}, n_j^{start})$ **then**
        append d to D^*;
    end
    else if $look\_up\_dis\_to\_depot < \tau(n_j^{start}, l_1)$ **then**
        append d to D^*;
    end
    else
        break
    end
end

2.3. Impact of Depot Re-Assignment Variable γ on Trip Connection

To more thoroughly demonstrate the impact of γ and thus including the distance from the depot to the first request in the edge cost function, the example in Figure 2.3 is used. This example does not represent a real scenario and is solely used for explanatory reasons. Similarly, the depot re-assignment used is an arbitrary example to showcase the two different scenarios. The values of decision variables used are as follows: $\beta = 0.5$ and $c^{rej} = 10$.

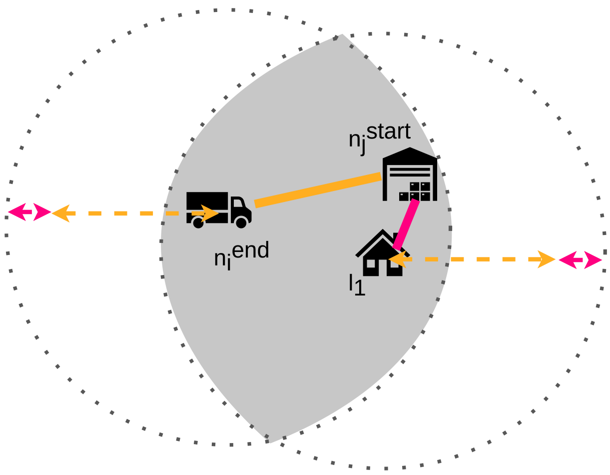


Figure 2.2: Illustration of the area to search for the subset depots to be used in the depot re-assignment step. The circles drawn have a radius of the distance travelled to connect the last order of the first trip and the first order of the second trip. Depots within the joined (grey) area of the two drawn circles could be applicable for depot re-assignment. Depots within this subset of depots could reduce the distance travelled to connect the two trips, making them options for re-assignment.

In the left top of Figure 2.3, a small bipartite graph is displayed. The graph consists of three nodes on either side of the graph marked by the vehicle icon on the left and the depot icon in the middle, connected with edges and their corresponding edge costs. Next to the bipartite graph another row of nodes is displayed, marked by the house icon. These nodes represent the first request of the second trip and the edges represent the connection distance from the depot to the first request of that particular trip on the right side of the bipartite graph. Solving this connection problem with the trip connection step without depot re-assignment results in the graph on the top right, with the selected edges displayed in green. The combination of the green edges results in the lowest possible objective of -3.5 ($-10 \cdot 0.5 + 9 \cdot 0.5 - 10 \cdot 0.5 + 4 \cdot 0.5 = -3.5$), resulting in this selection of connections. Using the same bipartite network, but applying an arbitrary depot re-assignment for the connection between the first trip on the left and the second trip on the right, results in the network in the bottom centre of the figure. The arbitrary depot re-assignment decreases the connection distance from 8 to 3 and increases the distance from the depot to the first request from 2 to 6 (as $3+6 < 2+8$, depot re-assignment takes place). In this scenario, the connection between the first trip on the left and the second trip on the right has another depot assigned. The connection distance between the trips is changed from 8 to 3 and the distance from the depot is changed from 2 to 6. Since the distance from the depot is not changed for the connection between the third trip on the left and the second trip on the right, the distance two is still present in the graph. As the depot re-assignment improves the connection distance and for $\gamma = 0$ only the connection distance is included in the objective function, the edge with the re-assigned depot is now selected as in the middle right graph. However, next to the decrease in trip connection distance, the distance from the depot to the first request increases. As this change is not included in the objective function for $\gamma = 0$, this increase is unbounded and leads to a higher total distance travelled. To prevent this from happening, the distance from the depot to the first request is included in the objective function for every $\gamma \neq 0$. In the given example, this results in not selecting the edge with the re-assigned depot, as the increase of the distance from the depot outperforms the decrease in trip connection distance. Although the scenario with $\gamma = 0$ (middle right) results in more connections and therefore a smaller fleet size, an increase in the total distance travelled happens. This increase is due to the higher number of connections as well as the change due to depot re-assignment. This trade-off between fleet size and total distance travelled shows the importance of including the distance from the depot in the objective function, controlled by the decision variable γ. For $\gamma = 0$ re-assignment is solely used for decreasing the fleet size independent of the increase of the distance from the depot to the first request. $\gamma = 1$ completely incorporates the distance from the depot to the first request in the objective function and, therefore, the focus lies on decreasing the total distance travelled.

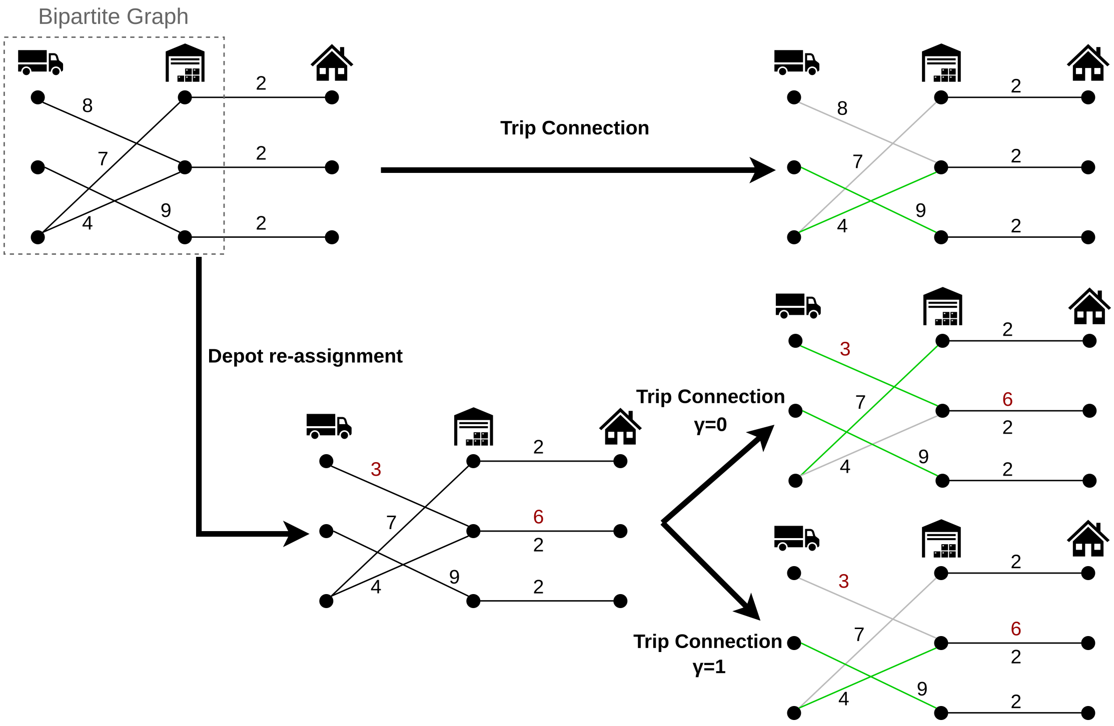


Figure 2.3: The impact of γ on the bipartite network and the trip connection. An example network is given in the top left and is exposed to the method without re-assignment (top) and the method with re-assignment (bottom). The method with re-assignment is separated in two trip connection problems with values $\gamma = 0$ and $\gamma = 1$. The method with re-assignment and a value of $\gamma = 0$ has more selected edges and therefore a smaller fleet size, but the distance travelled is increased. With a value of $\gamma = 1$, the original edge selection is made to prevent this increase in distance travelled. Decision variables: $\beta = 0.5$ and $c^{rej} = 10$

2.4. Solving the ILP

The trip connection problem, composed as an Integer Linear Program in Equation 4 of Part I of this report can be solved with a Hungarian Algorithm. However, to be provided with a solution, the program has to be slightly changed. For a Hungarian Algorithm to be used, the connecting constraints need to be altered. To be applicable for the use of a Hungarian Algorithm, every trip has to be connected to exactly one other trip. As this would eliminate the possibility of unconnected trips and therefore finishing or newly starting vehicles, another addition has to be made. This results in the altered Trip Connection Problem:

$$\begin{aligned} &\min_{i,j \in E} \sum c_{ij} * x_{ij} \\ &\qquad \text{subject to} \\ &\sum_{j \in W} x_{ij} = 1 \qquad \forall i \in U \\ &\sum_{j \in U} x_{ji} = 1 \qquad \forall i \in W. \end{aligned} \tag{2.1}$$

The bipartite network is extended with a set of dummy nodes and dummy edges. These dummy edges make sure the constraints can be met while allowing finishing and newly starting vehicles. Due to the addition of these dummy nodes and edges, the summations do now consider E^*, U^* and W^*, which resemble the original E, U and W respectively but include the dummy nodes and edges. The number of added dummy nodes is identical to the number of nodes already existing (equal to the number of trips) per side of the bipartite network. The newly added edges are constructed in three different steps:

1. Every node is connected to their respective dummy node in the other column (node 1 on the left is connected to dummy node 1 on the right).

2. Every edge existing in the original bipartite network is inversely added to the dummy part of the network (an edge between node 2 on the right and node 3 on the left is added to the dummy nodes as edge between node 3 on the left and node 2 on the right). As all dummy nodes also require exactly one connecting edge, this option can be used when the edge in the 'real' network is selected.
3. All edge cost for the dummy edges is set to 0, to prevent direct influence on the objective function.

The new bipartite graph edges E^*, including the original ones and the dummy ones, can now be used to solve the altered trip connection problem. The output of the Hungarian algorithm contains the selected edges, and with the elimination of the dummy edges, the chained trips can be found.

For an ILP solver such as Mosek or Gurobi, such change is not needed as the original constraints can be used. However, to present the edges and their corresponding cost to the algorithm, they are fitted into a two dimensional matrix. As the rows and columns correspond to the bipartite nodes of the left and right side respectively, also non-existent edges are now involved. To make sure these non-edges are not selected accidentally, their edge cost in the matrix is set to the maximum integer value. The results of using the Hungarian algorithm or the ILP are optimally equivalent. This means the objective value obtained by either method were equal. However, as the computation of the dummy nodes and edges caused longer runtimes upon execution, the ILP implementation was chosen to conduct the experiments. Due to memory problems occurring with the implementation of the ILP solver Mosek during runtime, another implementation was made in Gurobi.

2.5. Pseudo-Code of the Method

This section will contain the pseudo-code of the method as described in the method section of Part I of this report. Algorithm 4 describes the creation of the bipartite network, creating the edges for potential trip connections according to the time constraints. The algorithm used for the depot re-assignment step is given in Algorithm 1 in Section 2.2. For all depots, it is checked whether Equation (3) of Part I of this report holds with respect to the depot which has the shortest distance at that point. At first, this is the originally assigned depot, but this changes when a certain depot shortens the distance (as described by Equation (3)), causing this depot to become the fastest. Algorithm 5 describes the trip connection step to determine the ultimate connection of trips. This is done by solving the ILP of Trip Connection Problem (4). At last, Algorithm 6 describes the reconstruction of the vehicle plans. From every start of a vehicle plan, the whole plan is created by following the connections between the different trips. This results in the set of operational plans Ω and the fleet size |Ω|. The fleet mileage is determined by tracking the routes of the vehicles through all orders and trip connections.

Algorithm 4: Bipartite Network Creation

Function CreateBipartiteNetwork($\tau, P, \beta, \gamma, c^{rej}$)**:**
  $U \leftarrow [1, 2, ..., P]$;
  $W \leftarrow [1, 2, ..., P]$;
  $E \leftarrow []$;
  `// Create edges for potential trip connections within time constraints`
  for $u \in U$ **do**
    for $w \in W$ **do**
      if $t_u^{end} + \tau(u, w) \leq t_w^{start}$ **then**
        create e_{uw};
        $e_{uw}.cost \leftarrow -c^{rej} * \beta + (1-\beta) * \left[\tau(n_u^{end}, n_w^{start}) + \gamma * \tau(n_w^{start}, l_1)\right]$;
        append e_{uw} to E;
      end
    end
  end
  return E, U, W

Algorithm 5: Trip Connection

```
Function TripConnection(E, U, W):
    for i ∈ U do
        for j ∈ W do
            c_{i,j} ← ∞;
        end
    end
    for i, j ∈ E do
        x_{i,j} ← 0;
        c_{i,j} ← e_{i,j}.cost;
    end
    x_{i,j} ← solve Trip Connection Problem (4) using the sets of nodes U and W and the set of
      potential connections E;
    return x_{i,j}
```

Algorithm 6: Individual vehicle plans reconstruction

```
Function ReconstructVehiclePlans(x_{i,j}):
    Y ← [];
    Ω ← [];
    // add the starting trip of every chain to the set Y
    for i ∈ U do
        for j ∈ W do
            if x_{i,j} == 1 & x_{j,i} == 0 then
                append i to Y;
            end
        end
    end
    ω ← [];
    // create vehicle plans by tracking the trip connections from start to
       end
    for y ∈ Y do
        i ← y;
        append y to ω;
        // track the trip connections till the end
        while ∑_j x_{i,j} ≠ 0 do
            for j ∈ W do
                if x_{i,j} == 1 then
                    append j to ω;
                    i ← j;
                end
            end
        end
        append ω to Ω;
    end
    return Ω
```

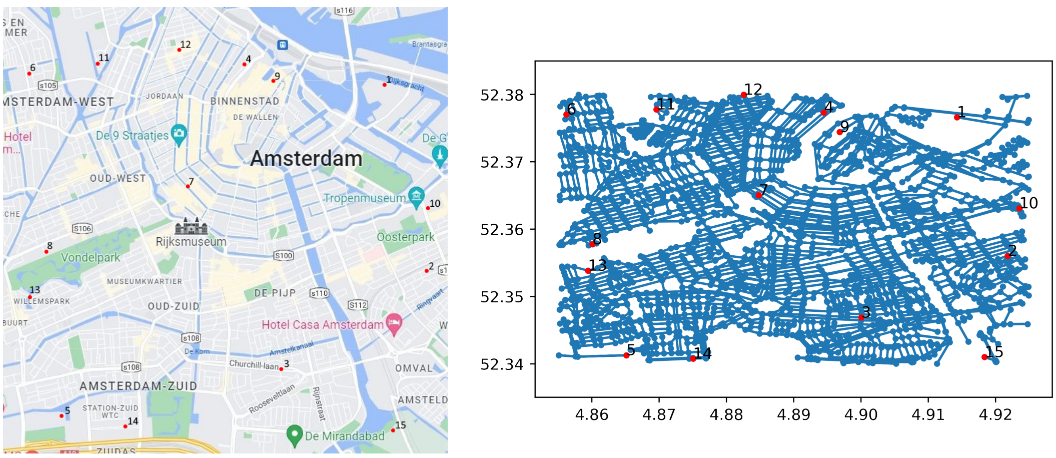
3

Generation of Input Data

To evaluate the proposed method, experiments have to be conducted and analysed. At the time of conducting these experiments, to the best of the knowledge of the authors, no open-source (benchmark) datasets applicable to evaluate the capabilities of the proposed methodology exist. Therefore, an operational environment of the city of Amsterdam was created, as described in Section 3.1. Additionally, three different trip datasets were generated as described in Section 3.2

3.1. Operational Environment

The operational environment is represented as a directed and weighted graph. The graph used in this work is a representation of the city centre of Amsterdam. A directed graph of 2717 nodes and 5632 edges is constructed representing the city as displayed in Figure 3.1. The arc weights resemble the travel times between two, by an arc connected, nodes calculated as the distance in metres divided by a constant speed of 36 km/h. The distance between two nodes on the graph is defined as the smallest sum of arc weights of all sets of arcs that span a path between the two nodes. The distances between every node on the graph are assembled into a lookup table to be used during runtime. The depot locations on the graph are determined using a k-center algorithm. With a greedy implementation, the first depot is placed on a random node and the other depots are, iteratively, located furthest from the existing depots. This procedure is done twenty times accounting for the random starting node. With the algorithm minimizing the maximum distance of all nodes to their closest depot, the iteration with the lowest of this maximum distance is the selected set of depots. As the distance used in this work is represented as traveltime, the distances in meters are divided by a constant speed of 36 km/h.



(a) Map of Amsterdam

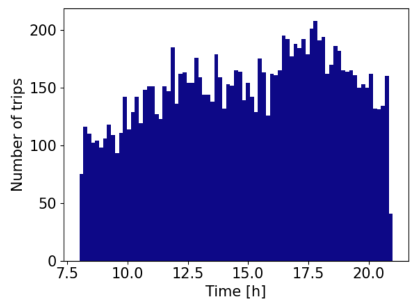
(b) Graph with 15 depots

Figure 3.1: Map of area of Amsterdam and the corresponding graph $G = (N, A)$ in blue with 15 numbered depots in red. The 15 depots D are displayed on the map for correspondence between the two.

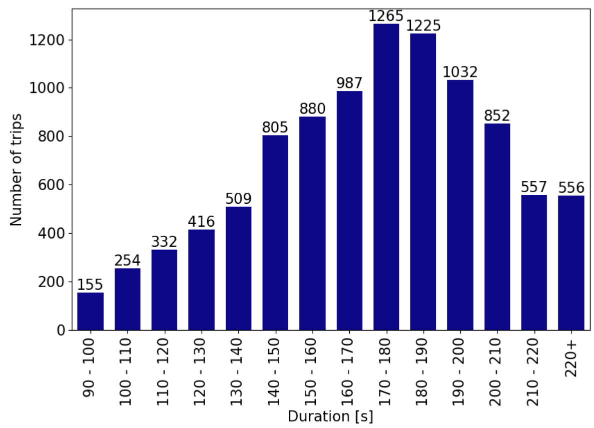
3.2. Trip Data

The datasets containing the trip data are generated, using the graph as described above. Three different datasets, each consisting of trips containing a total of 10.000 orders, with similar characteristics, are used. This section describes the temporal and spacial characteristics of these datasets. Although the examples given comply with one of the datasets, similar results could be expected from the other datasets.

The temporal characteristics of the trip data are shown in Figure 3.2. The start time of the trips over the course of the day are visualised in the left part of the figure, Figure 3.2a. Every bar represents a time segment of ten minutes and their height corresponds to the trips starting in that particular segment. At the start of the day the number of orders rises to a peak around noon. A second peak occurs at the end of the afternoon, followed by a decrease in orders during the evening. The duration of the trips is visualised in Figure 3.2b. The bars contain the trips with the duration slots as noted on the horizontal axis, with their height representing the number of trips concentrated in the particular slot. It can be observed that more than 50% of the trips have a duration between 140 and 190 seconds. The trips higher than 210 seconds are grouped together for visualisation reasons, but range up to 695 seconds. As the quantity of these high value durations is low, the mean value of the total set is approximately 174 seconds.



(a) Start time of trips over the course of a day



(b) Duration of trips

Figure 3.2: Temporal characteristics of one trip dataset. On the left the distribution of the starting time t^{start} for all trips $p \in P$ is visualised. Each bar resembles the number of trips starting within that ten minutes. On the right, the duration of the trips is visualised. Every bar resembles the number of trips with the duration of the range given for that specific bar.

The spatial characteristics of the datasets used comprise the information on the spatial distribution of the orders and depots upon the graph. All sets used consist of a uniform distribution of orders over the graph, as in Figure 3.3. The spatial distribution of the depots is accomplished with the use of a k-center algorithm. Due to the nature of this algorithm, depots tend to be placed along the graph boundaries, as shown in the graph images in Figure 3.4. In combination with the uniform distribution of the orders upon the graph, the distribution of orders among the depots is not uniform. To be clear, as this comprises the input data of the method, the characteristics shown were made without the use of depot re-assignment. The effect of depot re-assignment on the spatial characteristics will be shown in Chapter 4. Over the whole range of numbers of depots in Figure 3.4, the depots located more towards the centre of the graph, for example depot 3 and depot 7, serve significantly more orders than the other depots. Similarly, depots on outskirts of the graph, such as depot 1, 9 and 15, serve significantly less orders than other depots. These characteristics remain visible with the increase of the number of depots, however in strict numbers the depots have to serve less orders.

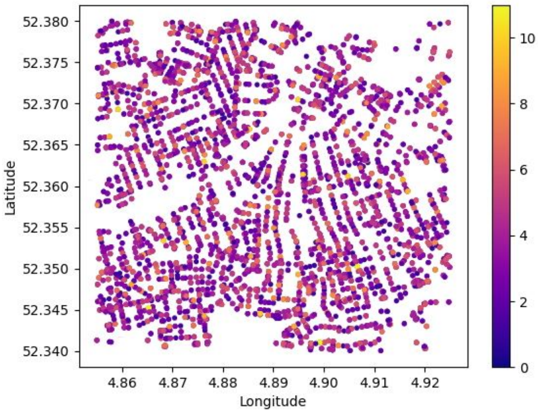
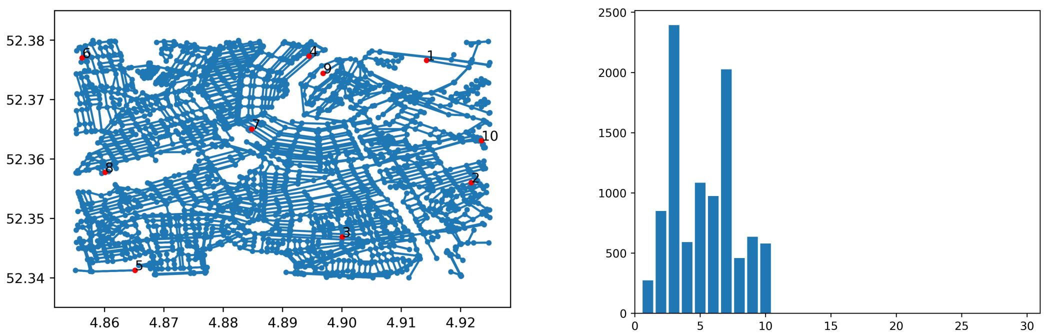
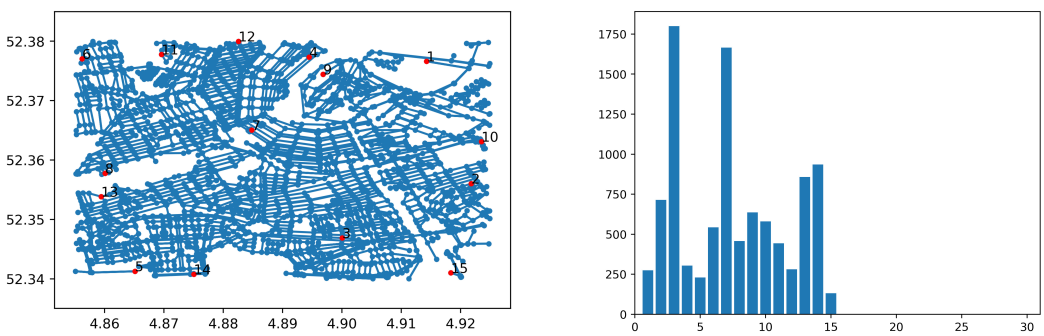


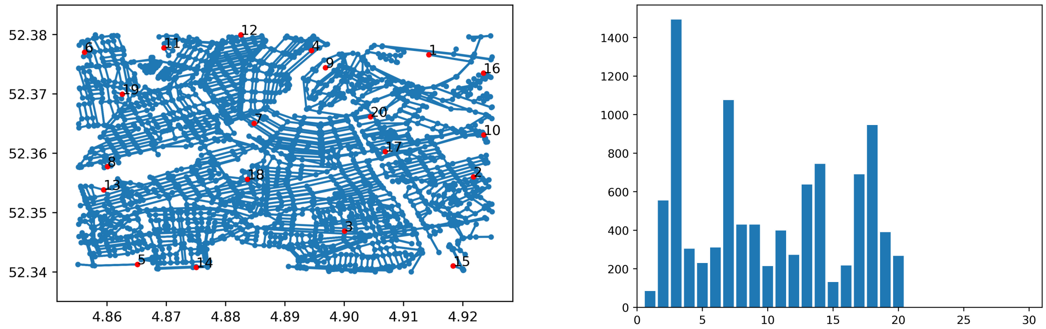
Figure 3.3: Spacial distribution of order locations n for all order requests R on the graph. The color of the points on the graph represents the occurrence of the location in the set of order locations as given by the bar on the right.



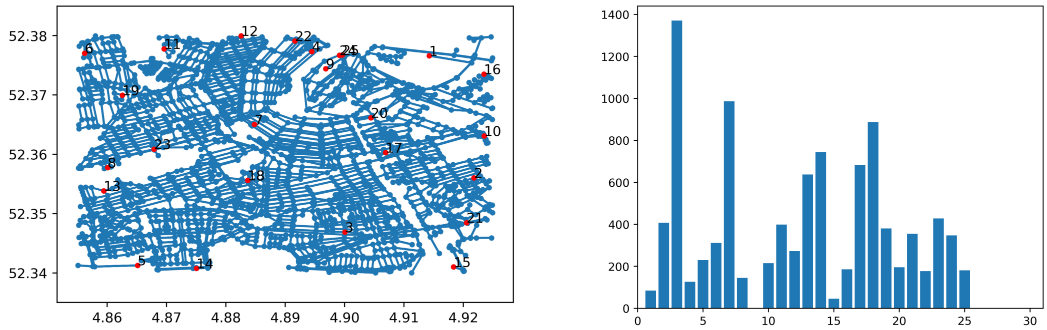
(a) 10 Depots



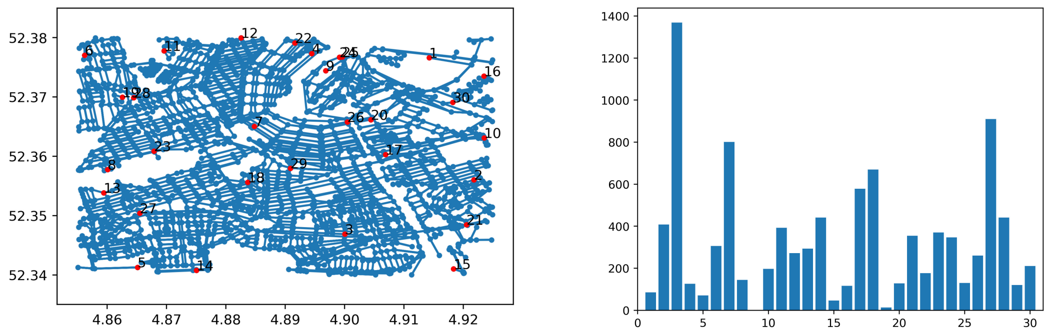
(b) 15 Depots



(c) 20 Depots



(d) 25 Depots



(e) 30 Depots

Figure 3.4: Spacial distribution of depots with corresponding order distribution per depot for different total numbers of depots. The graphs on the left show the distribution of the depots on the graph for different numbers of depots. The figures on the right represent the distribution of trips starting at the depot as numbered on the horizontal axis (n^{start} for all $p \in P$). Depots located in the centre area of the graph can be seen to have more trips started than the other depots.

4
Extended Experiments, Results and Discussion

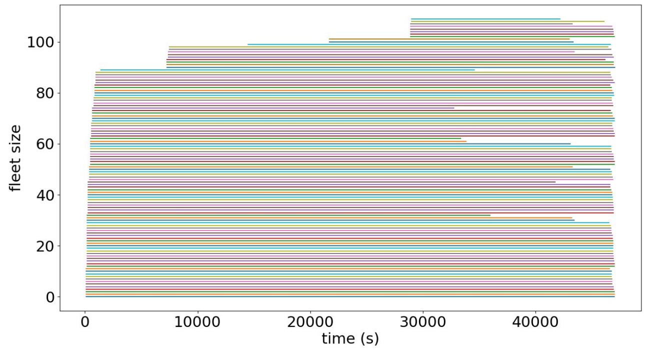
This chapter serves as an extension of the Experiments and Results section of Part I. Due to the page restrictions in the paper, the results of the experiments are only briefly illustrated and discussed. This extended experiments, results and discussions chapter builds on these results and provides an extended view on them. Next to that, the results are discussed throughout the chapter to provide more insight and a critical view on them. Section 4.1 directly extends the results of Part I by providing a more thorough analysis on the figures in this part. In Section 4.2 a sensitivity analysis for the different values of the added vehicle cost c^{rej} is conducted. The final section of this chapter, Section 4.3 analyses the effect of different numbers of depots.

4.1. Extended Analysis of Main Results

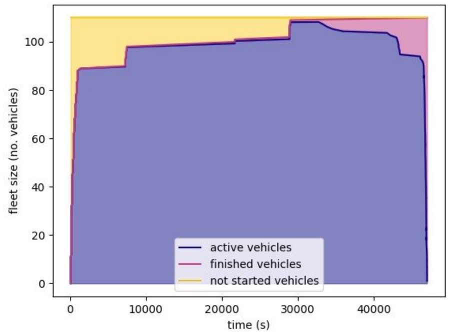
This section analyses the results as shown in Figure 4 of Part I more thoroughly. All illustrations in this section are visualisations of only one of the three datasets, as the results shown are very similar over all three datasets.

4.1.1. Individual Vehicle plans and Their Activity During the Day

The goal of the trip connection method is to connect different trips, and therefore eliminate potential vehicles. The chaining of two trips eliminates the vehicle potentially used by the second trip, as the vehicle serving the first trip now also serves the second trip. Chaining over the course of the whole day leads to long chains of trips, each individually served by a single vehicle. These individual vehicle plans are visualised from their start to end time in Figure 4.1a. The individual chains served by a single vehicle are plotted as horizontal lines, one for each vehicle. The lines start at the starting time of the first trip of the chain and end at the end time of the last request of the last trip of the chain. It can be seen that chaining indeed creates single-vehicle journeys over the course of the whole day. Several bumps can be identified in the plot, which are caused by the way the problem is solved. As stated in the method section of Part I, the trips are divided into batches for which the chains are calculated separately. On top of that, an extra iteration is done to chain in between different batches. The jumps in the plot correspond with the boundaries of these different batches. The chains created within one batch mostly span the time of the full batch. Additionally, the number of potential vehicles needed to serve the trips of one batch are not by default identical over the different batches. This results in a situation where, upon connecting between different batches, some chains cannot be inter-batch connected. As these chains tend to start at the batch boundary, time-wise, bumps tend to appear on batch boundaries.
Using the timeline information of the chains, the status of the vehicles over the course of the day can be visualized as done in Figure 4.1b. The dark blue area corresponds to the number of vehicles actively participating in the system. These vehicles are either delivering orders, driving to the start of another trip or waiting at a location to resume their route at a later moment in time. The yellow and pink areas, respectively, correspond to vehicles that have not yet entered and vehicles that are finished and have



(a) Timeline of individual full chains over the course of the day



(b) The size of the fleet and the status of each vehicle over time. The dark blue area corresponds to the active vehicles and the yellow and pink areas respectively correspond to vehicles that have not yet entered and vehicles that have left the system.

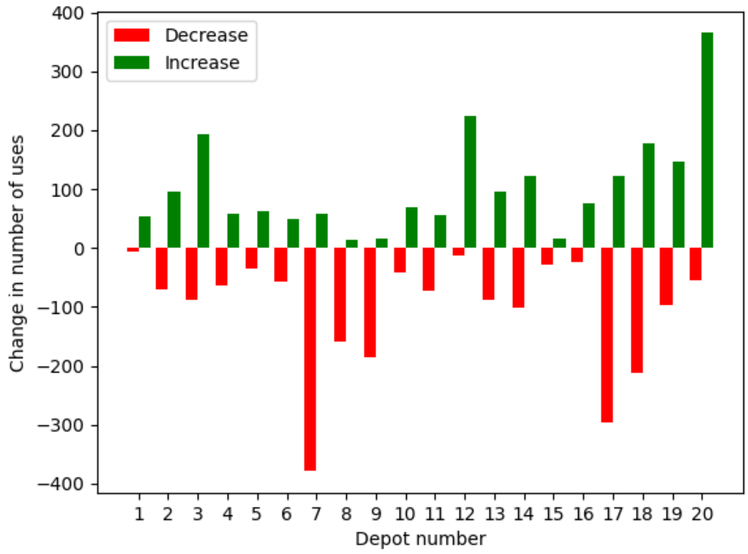
Figure 4.1: Timeline of the individual chains and the status of the vehicles over the course of a day. Decision variables: $\beta = 0.5$, $c^{rej} = 600, \gamma = 0.5$, re-assignment is used

left the system. It can be seen that of the total fleet size, most of the vehicles are actively used during the day. Similarly, the peak in the vehicles active in the system is close to the fleet size, addressing the need to have a fleet of this size available. In other words, vehicles are utilized efficiently and remain active for long periods of time. The majority of vehicles finish after most of the vehicles have started, confirming this efficient management.

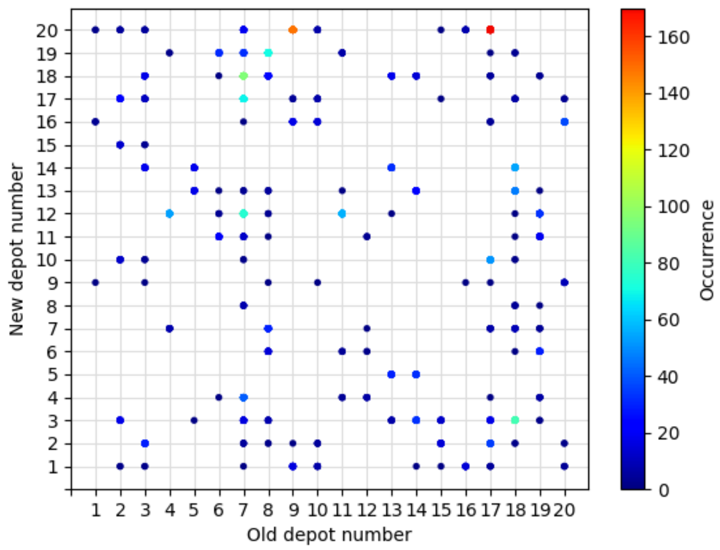
4.1.2. Change of Depot Use due to Depot Re-Assignment

Depot re-assignment changes the distribution of trips starting at a certain depot as originally depicted in Figure 3.4. The effect of depot re-assignment on this distribution for 20 depots is visualized in Figure 4.2a. The other values for the decision variables used are as follows: $\beta = 0.5, c^{rej} = 600, \gamma = 0.5$. The change in the number of uses of a depot is visualised per depot, split into the increase (green) and decrease (red) of the number of uses for that specific depot. Figure 4.2b visualizes the change in depot per depot. The x-axis holds the depot numbers of the originally assigned depots, whereas the y-axis holds the depot numbers of the new depots. A dot on the intersection of an old and new depot exists when a trip changes from the depot number on the x-axis to the depot on the y-axis. The color of the dot depends on the occurrence of the particular changes. Comparing these two figures to the locations of the depots and the initial distribution among these depots in Figure 3.4c, some interesting conclusions can be drawn. It can be seen that depots 3, 12 and 20 receive the largest increase in uses because of depot re-assignment. For these depots, it can be concluded from Figure 4.2b that this increase mostly originates from the same one or two depots as originally assigned locations. For depot 3, the most re-assignments come from depot 18, for depot 12 they come from depot 7 and for depot 20 they come from depot 9 and 17. This is logically explainable, as for all three depots, as the originally assigned depots are neighbouring depots. Neighbouring depots have a higher chance of being candidates for re-assignment, as they are more likely to be on the route to the depot. When a depot is on route to another depot, it is a candidate for re-assignment. The reason these depots have the highest increase in depot use is due to their location. For example, looking at depot 20, it can be seen that to go to depot 9 to pick up an order from any node of the map, in most cases a vehicle must travel through or close to depot 20. Therefore depot 20 is the better one to use. For the other depots, similar situations occur: from 18 to 3 and from 11 and 7 to 12.

Depots 7, 17 and 18 receive the largest decrease in use due to depot re-assignment. Due to their central location on the graph, the original number of uses is relatively high with respect to most other depots. As the areas of orders initially assigned to these depots are large, the radii of these areas are also large. This means orders are prone to be located further from a depot, which increases the chances of another depot being more suitable for pick-up upon trip connections. Depots 8, 9 and 15 are located on the outskirts of the graph and therefore have the lowest increase in uses due to depot re-assignment. Additionally, depots 8 and 9 have another depot in their close proximity, making them prone to a significant decrease in depot uses. In the end, almost half of their uses end up being done by another



(a) Increase and decrease of depot use due to depot re-assignment per individual depot.



(b) Shift of depot use from the initial situation to after depot re-assignment. A dot appears when at least one request was first served by the depot number on the x-axis and is now served by the depot on the y-axis. The color of the dot represents the occurrence of a particular shift.

Figure 4.2: The changes in depot uses due to depot re-assignment. Depot re-assignment causes an increase and decrease in the number of uses per depot as shown in the top graph. The specifications of these changes are visualised in the bottom graph, illustrating the shifts from individual depot to individual depot. Decision variables: $\beta = 0.5, c^{rej} = 600, \gamma = 0.5$.

depot. To conclude, it is shown how the individual depot uses shift due to depot re-assignment. The impact of the shift on an individual depot depends on its location on the graph, the close proximity of other depots, the density of orders in their initial use area and the initial number of uses. It is important to take this information into account for initial depot placement, suggesting to evenly spread the uses over all depots.

4.1.3. Pattern Change in Distance due to Depot Re-Assignment

The depot re-assignment method influences both the distance from the last request of the one trip to the depot of the other trip (connection distance) as the distance from the depot to the first request. For all selected trip connections, their connection distance is plotted against the distance from the depot to request 1, to analyse how depot re-assignment specifically affects these distances. This is visualised in Figure 4.3. The values for the other decision variables used are as follows:$c^{rej} = 600, \gamma = 0.5$. The connections which are unaffected by depot re-assignment are visualised in pink, whereas the affected connections are visualised in purple for the situation before depot re-assignment and in yellow after re-assignment. The situation before depot re-assignment assumes the trip connections which have another depot assigned by the method, but with their initial depot. A clear shift from the purple region to the yellow region can be observed in the figure due to the depot re-assignment. The reason for this shift is a decrease in the connection distance and an increase in the distance from the depot to request 1. This is exactly what can be expected from depot re-assignment. As explained in the method section of the paper, the initial choice of depot for a trip is the closest to the first request. As a result of this, with depot re-assignment the distance from the depot to the first request will increase. Due to Equation (3) of the paper, with an increase in distance from the depot to the first request, the connection distance has to decrease. Another interesting fact in the plot is the fact that all unaffected connections are located in the left bottom of the figure. This implies that the larger the connection distance, the more likely a trip connection is to have another depot assigned. Similarly, a small connection distance is less likely to be influenced by the depot re-assignment. This is logical, as for smaller connection distances, it is less likely that another depot will make this connection distance even smaller. With a larger connection distance this is more likely and thus more depot re-assignments occur.

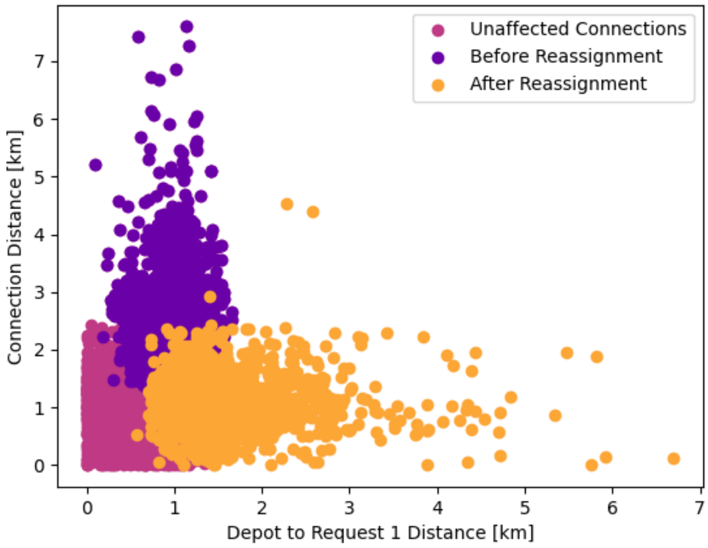


Figure 4.3: Effect of depot re-assignment on connection distance and distance from depot to the first request. The connections unaffected by depot re-assignments are visualised in pink. The affected connections are visualised in purple using the initially assigned depot (before depot re-assignment) and in yellow using the new depot. The trend shift from purple to yellow is inherent to the depot re-assignment method. Decision variables: $\beta = 0.5, c^{rej} = 600, \gamma = 0.5$.

4.1.4. Absolute Change in Distances due to Depot Re-Assignment

As shown in the previous section, depot re-assignment induces a trend shift in the connection distance to the depot and the distance from the depot to the first request. To analyse the effect of these changes on the total distances as shown in Figure 4 of Part I of this report, the changes in total, trip and connection distance are visualised for the different β-values in Figure 4.4. The values for the other decision variables used are as follows:$c^{rej} = 600, \gamma = 0.5$. These distances are the summation of all individual vehicle plans. The trip distance includes all distances travelled within the trips, and the connection distances includes all distances travelled in between trips. The absolute values plotted in the figure are the difference with respect to the base method. Therefore, a negative value represents a decrease with respect to the base method and a positive value an increase with respect to the base method. In Figure 4 of Part I of this report, the trend of the total distance difference due to depot re-assignment is visible.

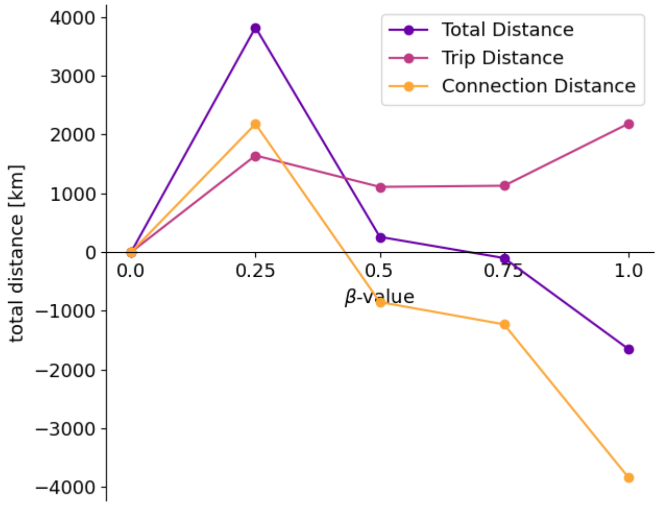
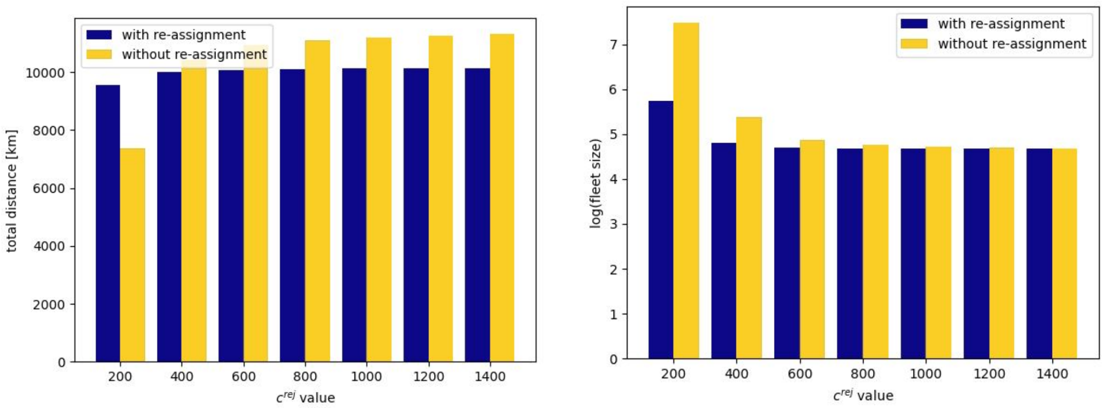


Figure 4.4: Change in total, trip and connection distances for different values of β due to depot re-assignment. The trip distances always increase due to depot re-assignment, whereas the connection distances can both increase or decrease. The total distance changes according to the magnitude of the changes of the other two distances. Decision variables: $c^{rej} = 600, \gamma = 0.5$.

Figure 4.4 gives more context on where this distance originates from. It can be seen that for $\beta = 0.25$, both the trip distance and the connection distance increase due to depot re-assignment, resulting in an increase in the total distance. The trip distance always increases due to depot re-assignment. As the initial depot is the closest to the order, changing this depot will always result in a larger trip distance. Surprisingly, the connection distance increases as well. At first glance, this is counter-intuitive as depot re-assignment should only happen when the connection distance is decreased. However, it can be explained due to the decrease in fleet size caused by depot re-assignment. A decrease in fleet size is due to an increase in the number of trip connections. With a higher number of trip connections, it can be logically explained that the total trip connection distance is higher as well. This effect is less prominent for the other β-values, as the difference in fleet size is less prominent as well. For $\beta = 0.5$, the effect is less present but still causes the total distance to increase due to depot re-assignment. For $\beta = 0.75$ and $\beta = 1.0$, there is no difference in fleet size for the two methods, and as such, the total distance decreases due to depot re-assignment. The trip distance increases from $\beta = 0.75$ to $\beta = 1.0$, which can be explained by the presence of more re-assignments in the selected trip connections. To conclude, the difference in total distance between the base method and the method with depot re-assignment is mostly related to the increase or decrease of the connection distance.

4.2. Sensitivity Analysis for Added Vehicle Cost c^{rej}

In the Problem Formulation section of Part I of this work, the value of c^{rej} is described as the rejection cost of the use of a vehicle. This influences the chaining as described in Section 2.1. According to the c^{rej} value, a new vehicle is deployed, or an existing vehicle is used to serve another trip. Low values of c^{rej} will result in less chaining and therefore a lower total distance travelled and a higher fleet size. High values of c^{rej} will result in the opposite: high total distance travelled and low fleet size. To analyse this effect, a sensitivity analysis for different values of c^{rej} is conducted. Figure 4.5 displays the effect of different values of c^{rej} on both the method with and without depot re-assignment. The values for the other decision variables used are as follows: $\beta = 0.5, \gamma = 0.5$. The above-described trends are visible in the figure, namely the low total distance travelled and high fleet size for low values of c^{rej} and vice-versa for high values of c^{rej}.
For low values of c^{rej}, the total distance is low as described above. For a value of $c^{rej} = 200$ the re-



(a) Total distance for different values of c^{rej} **edit for legend**

(b) The logarithm of the fleet size for different values of c^{rej}

Figure 4.5: Total distance and fleet size for different values of c^{rej}. Both the effect on the method with and without depot re-assignment are displayed. Decision variables: $\beta = 0.5, \gamma = 0.5$.

assignment has the largest impact on the total distance travelled with respect to the base method. This is interesting, as for this value the number of potential connections is the lowest. Therefore, the percentage of chains with re-assignment must be higher with respect to the other c^{rej} values. This can be explained by the fact that for lower values of c^{rej}, re-assignment makes a more significant impact on the edge cost. Relatively more edges can have their edge cost changed from a positive to a negative value, making them available for edge selection. For higher values of c^{rej} this effect does not appear as the c^{rej} value is significantly larger than the distances included in the edge cost $(\tau(n_i^{end}, n_j^{start}), \tau(n_j^{start}, l_1))$. A similar effect as described in the previous section appears between the method with re-assignment and the base method. The total distance is increased for low values of c^{rej} and for higher values this increase shifts to a decrease, which becomes saturated with higher values of c^{rej}. This saturation effect appears as the largest part of the objective value consists of the left part of the edge cost in Equation 2 of Part I of this report. This effect starts taking place at values of $c^{rej} = 800$ and higher. This occurs because the chosen value for c^{rej} in combination with the β-value is that high, it outperforms the value of the distances of the right side of the edge cost already for values of $c^{rej} > 800$. As the value for $c_{rej} \cdot \beta$ is significantly higher than the potential chaining distance, in all cases a chain is considered a better option than deploying a new vehicle. As the left hand side of Equation 2 of Part I solely determines the outcome of the trip connection method, re-assignments do not significantly alter the trip connection (as the changes in edge cost are mostly overshadowed by the left hand side of the equation). Therefore the effect of re-assignment is almost identical for values $c^{rej} > 800$.

The fleet size of the method with re-assignment is always better or equal with respect to the base method independent of c^{rej}. The largest difference occurs for the lowest values of c^{rej}. This is due to the higher percentage of re-assignments as explained for the total distance travelled plot in the previous paragraph. The effect of the re-assignment decreases with the increase of c^{rej}, until around $c^{rej} = 1200$ no difference between the methods can be observed. This is due to the fact that for high c^{rej} values, all bipartite edges (all possible chaining options timewise) will have a negative edge cost before the depot re-assignment step. The re-assignment will therefore not create chaining options which were not available in the base method (because their positive edge cost changed to negative), and the effect of the re-assignment is not significant enough to change the edge selection. Due to the high value of c^{rej}, the relative influence of the re-assignment on the edge cost is low. Therefore, the re-assignment does not change the edge selection and the same fleet size as with the base method is established. As the edges themselves are influenced by the re-assignment, the total distance travelled is affected and thus lower than the base method.

4.3. Analysis of Different Numbers of Depots

The experiments, as described before, are all conducted using a fixed number of depots, namely 20. To analyse the effect of the number of depots used, a comparison for different numbers of depots

is constructed. Using one dataset, the results of the method with and without depot re-assignment using different numbers of depots are visualised in Figure 4.6. The depot locations used can be found in Figure 3.4. The values for the decision variables used are as follows: $\beta = 0.5, c^{rej} = 600, \gamma = 0.5$. For both methods, the increase of the number of depots leads to a decrease in both the fleet size and the total distance travelled. This is due to the fact that with more depots available, the distance to deliver an order is shorter on average and deliveries become more efficient. As vehicles spend less time delivering orders, more chaining options become available and less vehicles are needed to deliver all orders. This effect is more significant among the low depot numbers, as adding a depot on a graph with a small number of depots decreases the distance of orders to a depot more significantly than adding a depot to a graph with a higher number of depots. This can be seen in Figure 3.4, where the addition of, for example, depots 11 and 12 are more significant in decreasing the distance of orders to a depot than the addition of depots 26 and 27. For all depot numbers tested, the method with re-assignment performs better than the method without re-assignment.
In all cases, a significant decrease in fleet size is visible, with a small increase in total distance travelled. What is interesting to see is that this increase in total distance travelled decreases with the number of depots. The effect of the re-assignment, measured in the difference between the method with and the method without re-assignment, increases with the number of depots used. This is logically explainable as with more depots present, the chances for re-assignment are higher. Therefore, more re-assignments take place and a more significant effect is visible.

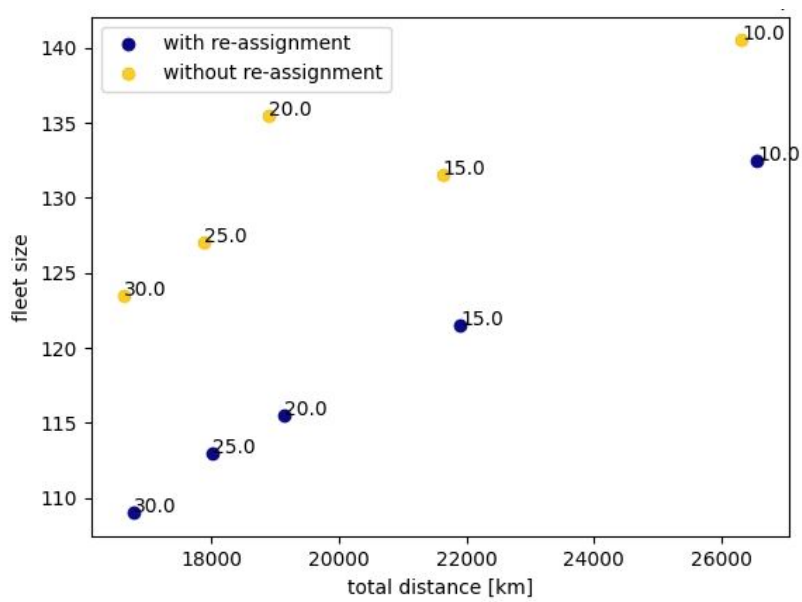


Figure 4.6: Fleet size against total distance for different numbers of depots for the base method and the method with re-assignment. Decision variables: $\beta = 0.5, c^{rej} = 600, \gamma = 0.5$.

5 Conclusion

This chapter contains the conclusion of this thesis report and suggestions for future research.

5.1. Conclusion

This work presented an optimization method for fleet design for last-mile on-demand logistics, yielding the required number of vehicles. The method determined the fleet size, the total distance travelled for each order concerning a last-mile on-demand logistics problem. This included the determination of the optimal pick-up location for the order requests, as it is ambiguous which depot should supply the requested goods. This work contributed by being the first to investigate methods for doing fleet design specifically for last-mile on-demand logistics considering multiple depots and variable pick-up locations.

Given a graph representing the operation environment and a set of trips required to be delivered, the method determined the optimal chaining of the trips. Chaining two trips has the benefit of reducing the number of vehicles used, as chained trips are served by a single vehicle. First, the time-wise feasible connections between pairs of trips were calculated. Per these connections, the method determined the optimal pick-up location. The last step of the method was to find the ultimate trip connection, which is done by solving an ILP. The output of the method was the number of vehicles needed to serve all trips, the individual operational plans of these vehicles, the pick-up locations of all requests and the total distance all vehicles travel.

Different experiments were conducted to prove the value of the depot re-assignment method compared to the method without depot re-assignment. The re-assignment method decreased or retained the fleet size while decreasing or increasing the total distance travelled. The increase in total distance travelled proved the existence of a trade-off between the fleet size and the total distance travelled. The way the method was designed allowed for operator choice on the trade-off between more kilometres travelled for a smaller fleet size and vice versa. It was shown that a significant decrease of the required fleet size could be established by a minor increase in total travelled distance. The optimal trade-off between the fleet size and the total distance travelled could be determined for a specific operation with the knowledge of operational costs for that operation. A sensitivity analysis showed the effect of the method's decision variables on the fleet size and the total distance travelled. The described trade-off between fleet size and the total distance travelled was evident over the different values of the decision variables. Therefore, with an appropriate selection of values of the decision variables, based on the operational scenario at hand, the best trade-off for that scenario can be found and an efficient utilization of the fleet can be established.

In general, the results were promising, allowing the operator to make such operational decisions with the results of the proposed method. Most importantly, as this was the main goal of this research, the method was proven to be effective in determining the fleet size for a last-mile on-demand logistics problem, with a fleet utilization of 98%.

5.2. Future Work

Future research could include the addition of depot re-assignment for infeasible edges of the bipartite network. The depot re-assignment, as it is implemented now, is part of the trip connection method. Because of this, only the already existing potential connections between trips are applicable for depot re-assignment. Doing so ignores the connections which at first are not feasible, as the time constraints do not allow these connections. However, with depot re-assignment unfeasible connections between trips can become feasible, as depot re-assignment decreases the connection time between the trips. Therefore, by allowing depot re-assignment upon creating the set of potential connections, more potential connections can be found. This has the potential of creating better operational plans and further decreasing the fleet size.

In addition, with the shorter distance to the first order of a trip, the total delivery time of all orders of a trip decreases. This decrease in time is not used to its full benefit. The orders of a trip are not shifted in time, which results in the time benefit being used as waiting time. By allowing the order delivery times to be changed, earlier completion times of the trips can be established. As well as the previous suggestion, this has the potential to lead to more potential connections and, thus, better operational plans. As such, future research could determine the effect of allowing a variable pick-up time according to possible depots.

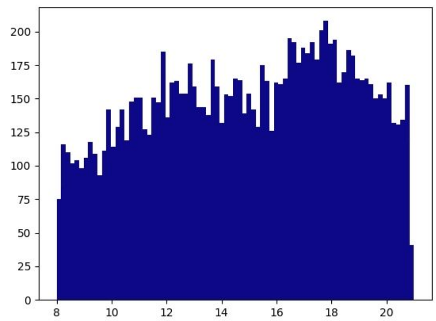
Moreover, the execution of the method could be altered. As was stated in the method section, the trips are handled in batches for computational reasons. These different batches are then connected with a second iteration. This is sufficient, however, it prevents certain trip connections from being made and therefore influences the final trip connection and the fleet size. Despite the second iteration, and, therefore, the ability to make inter-batch connections, the batch boundaries cause bumps in the number of active vehicles. To prevent this, future research should include investigating this issue and finding a way to facilitate a more smooth transition over the batch boundaries.

On top of that, the execution with different batches and the second iteration has one limitation. A situation in which a small number of connections is made during the batch iterations, for example for a low β-value, can prove to be computationally too expensive for the second iteration. A low number of connections in the batches results in a high number of connections in the second iteration, to the extent in which it cannot be solved efficiently. Future research could include finding a way to prevent this and make even these problems efficiently solvable. A first thought on this would be to divide the second iteration into sub-iterations, whereas every sub-iteration makes the connection between two neighbouring batches of the first iteration. This will cover most inter-batch connections and further investigation has to determine whether a third iteration has to be done to connect the starts and ends of each chain created per sub-iterations (similar to how the second iteration currently connects the different batches).

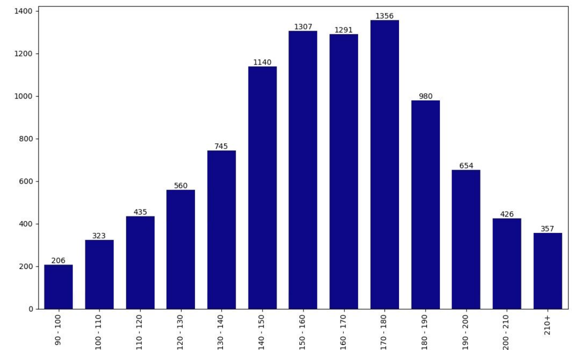
Furthermore, this study only includes experiments with five different generated trip datasets. While this is sufficient to prove the capabilities and value of the method at hand, this will not suffice for practical usage and accurate operational decisions. To decide on the values of the decision variables for a particular use case and implementation in practice, more and more varied datasets need to be used. Ideally, this data is close to the real-life application.

A
Spacial and Temporal Information Different Datasets

This appendix contains the spacial and temporal information figures similar to Figures 3.2 and 3.3 for the other two datasets used.



(a) Start time of trips over the course of a day



(b) Duration of trips

Figure A.1: Temporal characteristics of the second trip dataset. On the left the distribution of the starting time t^{start} for all trips $p \in P$ is visualised. Each bar resembles the number of trips starting within that ten minutes. On the right the duration of the trips is visualised. Every bar resembles the amount of trips with the duration of the range given for that specific bar.

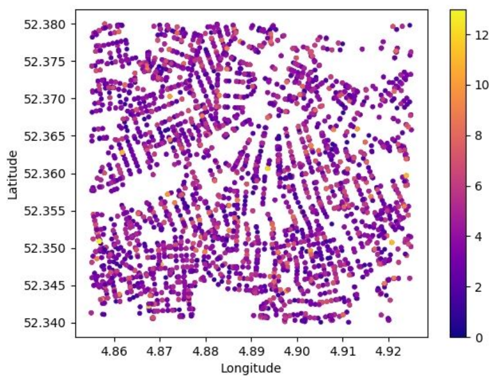
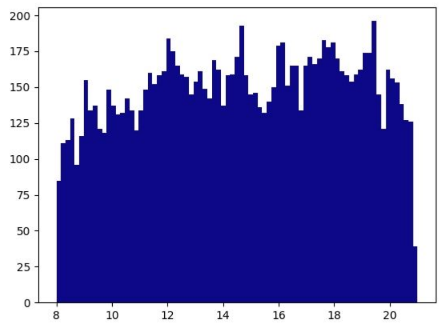
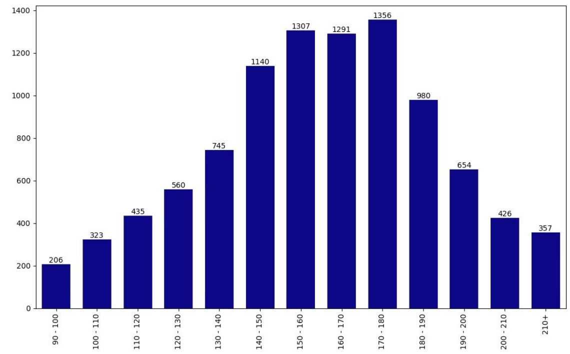


Figure A.2: Spacial distribution of order locations n for all order requests R on the graph of the second trip dataset. The color of the points on the graph represent the occurrence of the location in the set of order locations as given by the bar on the right.



(a) Start time of trips over the course of a day



(b) Duration of trips

Figure A.3: Temporal characteristics of the third trip dataset. On the left the distribution of the starting time t^{start} for all trips $p \in P$ is visualised. Each bar resembles the number of trips starting within that ten minutes. On the right the duration of the trips is visualised. Every bar resembles the amount of trips with the duration of the range given for that specific bar.

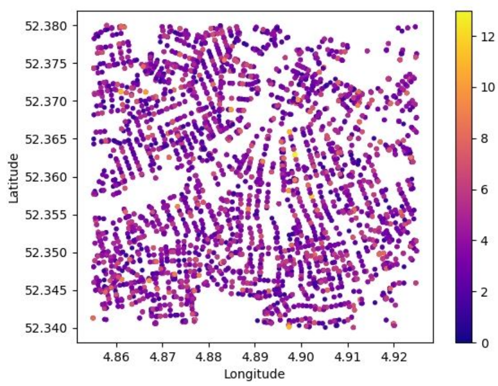


Figure A.4: Spacial distribution of order locations n for all order requests R on the graph of the third trip dataset. The color of the points on the graph represent the occurrence of the location in the set of order locations as given by the bar on the right.

Bibliography

[1] V. Pillac, M. Gendreau, C. Guéret, and A. L. Medaglia, "A review of dynamic vehicle routing problems," European Journal of Operational Research, vol. 225, no. 1, pp. 1–11, 2013.

[2] kpmg, "The truth about online consumers." https://assets.kpmg/content/dam/kpmg/xx/pdf/2017/01/the-truth-about-online-consumers.pdf, 2017. Accessed: 2022-11-11.

[3] Invespcro, "Same day delivery." https://www.invespcro.com/blog/same-day-delivery/, 2022. Accessed: 2022-11-11.

[4] Yahoo Finance, "Insights on the same day delivery global market to 2027." https://finance.yahoo.com/news/insights-same-day-delivery-global-105800728.html, 2022. Accessed: 2022-12-05.

[5] fsgjournal, "A bottle of wine in 10 minutes? - the growth of on demand delivery." https://fsgjournal.nl/article/2022-01-11-a-bottle-of-wine-in-10-minutes-the-growth-of-on-demand-deliver 2022. Accessed: 2022-09-09.

[6] de econometrist, "The rise of the on-demand grocery delivery industry." https://www.deeconometrist.nl/economy/the-rise-of-the-on-demand-grocery-delivery-industry/, 2022. Accessed: 2022-09-09.

[7] P. M. Bösch, F. Becker, H. Becker, and K. W. Axhausen, "Cost-based analysis of autonomous mobility services," Transport Policy, vol. 64, pp. 76–91, 2018.

[8] S. Hörl, C. Ruch, F. Becker, E. Frazzoli, and K. W. Axhausen, "Fleet control algorithms for automated mobility: A simulation assessment for zurich," in 2018 TRB Annual Meeting Online, pp. 18–02171, Transportation Research Board, 2018.

[9] S. N. Kumar and R. Panneerselvam, "A survey on the vehicle routing problem and its variants," 2012.

[10] K. Braekers, K. Ramaekers, and I. Van Nieuwenhuyse, "The vehicle routing problem: State of the art classification and review," Computers & Industrial Engineering, vol. 99, pp. 300–313, 2016.

[11] B. Yu, N. Ma, W. Cai, T. Li, X. Yuan, and B. Yao, "Improved ant colony optimisation for the dynamic multi-depot vehicle routing problem," International Journal of Logistics Research and Applications, vol. 16, no. 2, pp. 144–157, 2013.

[12] H. Xu, P. Pu, and F. Duan, "A hybrid ant colony optimization for dynamic multidepot vehicle routing problem," Discrete Dynamics in Nature and Society, vol. 2018, 2018.

[13] Y. Wang, J. Zhe, X. Wang, Y. Sun, and H. Wang, "Collaborative multidepot vehicle routing problem with dynamic customer demands and time windows," Sustainability, vol. 14, no. 11, p. 6709, 2022.

[14] M. Kronmueller, A. Fielbaum, and J. Alonso-Mora, "Automated last-mile on-demand logistics from multiple depots." 2022.

[15] J. Alonso-Mora, S. Samaranayake, A. Wallar, E. Frazzoli, and D. Rus, "On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment," Proceedings of the National Academy of Sciences, vol. 114, no. 3, pp. 462–467, 2017.

[16] P. C. Jones and J. L. Zydiak, "The fleet design problem," The Engineering Economist, vol. 38, no. 2, pp. 83–98, 1993.

[17] S. Klosterhalfen, J. Kallrath, and G. Fischer, "Rail car fleet design: Optimization of structure and size," International Journal of Production Economics, vol. 157, pp. 112–119, 2014.

[18] A. Hoff, H. Andersson, M. Christiansen, G. Hasle, and A. Løkketangen, "Industrial aspects and literature survey: Fleet composition and routing," Computers & Operations Research, vol. 37, no. 12, pp. 2041–2061, 2010.

[19] V. M. Dalfard, M. Kaveh, and N. E. Nosratian, "Two meta-heuristic algorithms for two-echelon location-routing problem with vehicle fleet capacity and maximum route length constraints," Neural Computing and Applications, vol. 23, no. 7, pp. 2341–2349, 2013.

[20] B. Golden, A. Assad, L. Levy, and F. Gheysens, "The fleet size and mix vehicle routing problem," Computers & Operations Research, vol. 11, no. 1, pp. 49–66, 1984.

[21] F. Gheysens, B. Golden, and A. Assad, "A comparison of techniques for solving the fleet size and mix vehicle routing problem," Operations-Research-Spektrum, vol. 6, no. 4, pp. 207–216, 1984.

[22] M. W. Levin, K. M. Kockelman, S. D. Boyles, and T. Li, "A general framework for modeling shared autonomous vehicles with dynamic network-loading and dynamic ride-sharing application," Computers, Environment and Urban Systems, vol. 64, pp. 373–383, 2017.

[23] D. J. Fagnant and K. M. Kockelman, "Dynamic ride-sharing and fleet sizing for a system of shared autonomous vehicles in austin, texas," Transportation, vol. 45, no. 1, pp. 143–158, 2018.

[24] M. Cáp and J. A. Mora, "Multi-objective analysis of ridesharing in automated mobility-on-demand," in RSS 2018: Robotics-Science and Systems XIV, 2018.

[25] A. Wallar, W. Schwarting, J. Alonso-Mora, and D. Rus, "Optimizing multi-class fleet compositions for shared mobility-as-a-service," in 2019 IEEE Intelligent Transportation Systems Conference (ITSC), pp. 2998–3005, IEEE, 2019.

[26] M. M. Vazifeh, P. Santi, G. Resta, S. H. Strogatz, and C. Ratti, "Addressing the minimum fleet problem in on-demand urban mobility," Nature, vol. 557, no. 7706, pp. 534–538, 2018.

[27] B. Qu, L. Mao, Z. Xu, J. Feng, and X. Wang, "How many vehicles do we need? fleet sizing for shared autonomous vehicles with ridesharing," IEEE Transactions on Intelligent Transportation Systems, 2021.