POLITECNICO DI TORINO

Master's Degree in ICT for Smart Societies

Master's Degree Thesis

Joint Optimization of Relocation and Routing Strategies for E-Scooter Sharing System

Supervisors

Prof. Marco MELLIA

Prof. Luca VASSIO

Prof. Danilo GIORDANO

Candidate

Di WANG

April 2023

Summary

In recent years, shared micro-mobility services - including shared bicycles, e-scooters, etc. - have gained momentum, also as an extension of public transportation such as metro and buses. Thanks to their abilities to reduce air pollution and to improve traffic congestion, as well as the advantages of their low cost and flexibility, which could effectively alleviate "the last mile" problem, shared micro-mobility services hence have gradually become an indispensable part of the urban transportation system. However, the enhanced flexibility and the sharp increase in the number of vehicles have also brought new challenges. One of the most significant is the spontaneous imbalance problem especially in peak hours, which could lead to the accumulation of scooters in some areas while unsatisfying demand in others, resulting in a waste of resources and loss of profit.

In this context, this paper takes a dock-less e-scooters sharing system as the research object, and proposes joint optimization of relocation and routing strategies. Specifically, we solve the e-scooter imbalance problem in two phases. In the first phase, we assume multiple dispatch trucks perform dynamic repositioning of scooters while considering expected future demand, that is, trucks pick up a certain number of scooters from the zones with surplus scooters and drop them off at the zones with unsatisfied demand of scooters. At this phase, each truck is only taking scooters from a single pick-up zone and bringing them to a single drop-off zone, i.e., matching couples of zones. The relocation is dynamic: during a time slot, the relocation operations take place while at the same time the customers are also using scooters across zones. We formulate the dynamic repositioning problem as a Mixed Integer Linear Programming (MILP) model to optimize the dispatch trucks' relocation operation, that is, the number of scooters picked up/dropped off and the corresponding zones. The model aims at maximizing total profit (that is, the revenue of satisfied trips minus the relocation cost), taking into account parameters including the predicted demand, the number and capacity of dispatch trucks, the unit mileage cost per truck and the unit cost for relocating one scooter, the look-ahead horizon and the optimization horizon. In the second phase, given the output of the first model, a set of vertices representing city zones and the corresponding relocation (positive in case of drop off, negative in case of pick up) is given, the

dispatch trucks start from the depot with an empty load, passing all vertices and performing the designated relocation operations follow proper routes, and finally back to the depot. This problem is actually denoted as the Vehicle Routing Problem with Pickup and Delivery (VRPPD). Two MILP models are formulated to optimize the routes of multiple dispatch trucks. The first model $M2.1$ is constructed on the basis of the classical Pickup and Delivery Traveling Salesman Problem (PDTSP), in which each zone is visited exactly once. The objective function is to minimize total routing cost, with the additional constraint of maximum routing time. While the model $M2.2$ relax the traversing constraint, the dispatch trucks are allowed to ignore the unfavourable zones, and correspondingly we introduce the unit relocation revenue to incentivize dispatch trucks to fulfil as many designated relocation needs as possible in profitable situations.

Then, the models are applied to solve toy case problems. We have designed 3 toy cases: One for the first phase, including 4 zones and 100 shared scooters in the system, 2 scenarios are further distinguished based on different OD matrices. One for the second phase, two couples of zones having a total number of relocations of 11 scooters are given. And the third one is for joint optimization of relocation and routing strategies, which includes 8 zones and 200 scooters. Through these toy cases, we have verified the correctness and effectiveness of the proposed models, and further discussed the impact of the different parameters. Specifically, in toy case 1, larger truck capacities and numbers in most cases lead to higher profits until all trip demands are met. Relocation costs directly affect relocation decisions and profits, with higher costs resulting in fewer relocation operations and, in extreme cases, no relocation at all. The impact of the look-ahead horizon on profit and relocation strategy is not as straightforward as the other parameters, it is largely influenced by other factors such as OD matrix, truck capacity, etc. Therefore, the optimal look-ahead horizon should be customized to the specific scenario and parameter settings. In toy case 2, we explore the difference between the two models in terms of the feasible region. As well as in toy case 3, the results of 5 optimization round is presented, showing a raise of 28.64% on average profit.

Finally, the developed joint optimization is applied to a practical case, in which a trace of recorded e-scooter trips is collected from the city of Louisville. The result of the developed joint optimization shows an increase of 6.6% in the average profit, compared to the one without optimization.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Shared mobility and shared micro-mobility

The concepts of shared mobility have gained popularity in recent years as an alternative to traditional car ownership, providing affordable, convenient, and sustainable transportation options for urban residents. The earliest shared mobility service was car sharing, which was launched in Zurich, Switzerland in 1948[1]. Over the following decades, shared mobility services rapidly developed around the world and gave rise to various variants, Machado et al. [2] defined several modalities of shared mobility, including station-based and free-floating car-sharing system (CS), bike sharing system (BSS), personal vehicle sharing, carpooling, on-demand ride services, etc.

Nowadays, bike sharing are often classified along with e-scooters, skateboards and other small-scale vehicles as another innovative urban transportation solution, known as shared micro-mobility services, due to the rapid development of micro-transportation including shared bicycles in the past decade. Actually, the earliest bike sharing appeared as early as 1965 - Amsterdam's 'White Bicycle'[3], but it was not until after 2015 that bike sharing services really developed rapidly around the world and began to attract more and more attention[4]. E-scooters sharing service, on the other hand, have only emerged in the past few years, but have quickly become a significant part of micro-mobility services. Compared to other shared mobility, micro-mobility services have advantages in various aspects, including the following examples:

- Sustainability: Shared micro-mobility offers a sustainable mode of transportation that reduces the environmental impact of traditional transportation modes. Bicycles and e-scooters are (almost) emission-free and require minimal energy to operate. By encouraging the use of shared micro-mobility, cities can reduce their carbon footprint and improve air quality (see [5][6][7]).

- Accessibility: Shared micro-mobility offers increased accessibility, expands the reach of public transportation by providing users with a convenient and flexible last-mile transportation option. This makes it easier for people to access transportation in areas where public transit may not be available or convenient, such as suburban or rural areas, or in places where cars cannot easily reach, such as narrow streets in inner-city areas (see [8][9][10][11]).
- Flexibility: Shared micro-mobility devices can be faster and more maneuverable than cars, especially in congested urban areas and for short trips, such as commuting to school or running errands, as well as for longer journeys up to 20km (see [12][13]).
- Health benefits: Shared micro-mobility services, particularly bicycles, can provide health benefits to users by encouraging physical activity and reducing sedentary behavior. Regular physical activity has been linked to a range of health benefits, including reduced risk of chronic diseases and improved mental health (see [14][15]).

In addition to their advantages, researchers have increasingly focused on various aspects of shared mobility/micro-mobility, including the user behavior and preferences, as well as service design and optimization. One trend is that free-floating sharing services are becoming more popular and gradually replacing station-based sharing services. Taking bike sharing as an example, dock-less bike-sharing system (DBS) refers to a bike rental service provided by companies in public areas without fixed parking stations. It can be lock/unlock and pay by simply scanning the QR code on the bike through a smartphone application, which provides greater convenience and flexibility for users. In October 2017, a survey conducted in Tianjin, China revealed the convenience of renting and returning bicycles, as well as the ease of payment through a smartphone, were the two main reasons to choose DBS [9]. Similar study refers also to car sharing (see [2][16][17]).

1.2 E-Scooter sharing system

An e-scooter, also known as an electric scooter, is a stand-up scooter powered by an electric motor as shown in Figure 1.1. E-scooters typically have a maximum speed of 24 to 32 km/h. These vehicles are designed to be small, lightweight, and portable, making them an ideal option for short distance travel within urban areas.

The earliest e-scooter sharing system emerged in California in 2017, and rapidly expanded to various countries over the world within a short span of a few years [18]. According to statistics [19], by the end of 2019, electric scooter sharing had already garnered more than 200,000 users in Europe alone, a usage rate four times higher than that of bike-sharing.

Presently, the vast majority of e-scooter sharing systems operate with a free-floating model. A typical e-scooter sharing system operates through a smartphone app, allowing users to rent the e-scooters offered by carrier. Using the app, users can first locate nearby e-scooters available for rent, once the user finds a scooter, they can use the app to scan the QR code on the e-scooter as shown in Figure 1.2, which unlocks it and allows the user to start riding. When the user is finished with the ride, they can park the e-scooter in the destination and click on the app to end the rental. The app then charges the user's account according to the time they used the e-scooter.

Figure 1.1: electric-scooters

Figure 1.2: QR code on e-scooters

Similar to DBS, the ease of rental and return procedures is also one of the reasons why e-scooter sharing has gained popularity. In fact, beyond that, e-scooters share many similarities with bicycles, including the common benefits of shared micro-mobility mentioned in 1.1, or the problem of improper parking and safety concerns.

However, despite the similarities, as an emerging service, e-scooter sharing systems still possess some distinct features that set them apart. For instance, travel purposes of DBS and e-scooter sharing system are different [20], with DBS being primarily utilized for commuting between home and workplace, while e-scooter sharing system is used for leisurely purposes, owing to the greater fun factor and less technical skill required for riding an e-scooter compared to a bicycle; [21] found that e-scooter sharing systems had high repositioning ratios than DBS, as they require repositioning for battery charging purposes, which is not a concern for DBS. Conversely, weather conditions have a greater impact on DBS, with hot weather causing a decrease in DBS trips but having a smaller impact on e-scooter sharing systems; Due to the difference in propulsion, e-scooter sharing systems have longer ranges and a broader user base [22].

Due to the relatively short emergence of e-scooter sharing systems, there has not been sufficient research and reporting on their unique characteristics, and policy administrators and system carriers have not given them enough attention. A report released by the European Road Safety Research Institutes[23] indicates that in most European countries, e-scooter sharing systems are equated with DBS in the development of relevant regulations. However, ignoring the differences between e-scooter sharing systems and DBS could lead to improper management of the e-scooter sharing systems or inadequate regulations, thereby shortening the life cycle of shared scooters, increasing greenhouse gas emissions, and going against the expected sustainability of e-scooter sharing systems [24]. Therefore, it is important to treating e-scooter sharing systems differently from DBS in research and developing appropriate system management policies and regulations.

1.3 The re-balancing problem

As the free-floating shared e-scooter systems play an increasingly important role in cities due to their flexibility and convenience, the accompanying vehicle re-balancing issue has become a big challenge.

On one hand, the success of shared e-scooter services heavily relies on the efficient allocation and management of shared vehicles. Which allows users to easily find available vehicles nearby when they need the service. On the other hand, due to user behavior, shared e-scooter systems have a natural imbalance characteristic. At specific times, certain areas have a higher demand for scooter usage, but there may not be enough scooters available in those areas to satisfy users' needs. At the same time, other areas may have a lower demand for scooter usage, but still be popular destinations, leading to a surplus of scooters. The accumulation of scooters in some areas while unsatisfying demand in others, would result in a waste of resources and loss of profit and even negative effects on traffic in those areas. Hence, the relocation of shared vehicles has become a critical issue to ensure their availability and accessibility, as well as to increase profitability and reduce environmental impact.

Overall, the re-balancing issue can be divided into two categories: dynamic re-balancing and static re-balancing. If during a time slot, the relocation operations take place while at the same time there are also trips generated by users, it is considered dynamic re-balancing. Otherwise, it is considered static re-balancing. It is also possible to classify based on relocation strategies, most of the research is based on two primary relocation strategies, as follows:

- Carrier-based relocation: Suppliers balance vehicle supply and demand by utilizing carriers (i.e., employees of car-sharing providers) to drive, tow or carpool surplus vehicles from low-demand locations to high-demand locations

(see [25], [26], [27]).

- User-based relocation: Providers incentivize users by Dynamic pricing scheme, to charge fees and discounts for rentals and returns in certain areas, in an attempt to reduce rentals from high-demand starting locations to low-demand end locations or to change their destination and return vehicles to locations with insufficient supply rather than oversupplied locations (see [28], [29], [30]).

Generally speaking, the relocation costs of user-based relocation strategy are lower. However, developing and managing such a system is challenging, as it requires real-time collection and analysis of data on vehicle availability and demand, monitoring and managing relocation plans, devising rewards and incentives to encourage users' participation in re-balancing work, providing real-time information about relocation operations and communicating with users. It also entails tracking user engagement and ensuring that vehicles are moved to the correct areas. Therefore, there is more research on carrier-based relocation strategies. In this thesis, we focus on dynamic re-balancing issues and carrier-based relocation strategies.

One common way to solve the re-balancing problem is to abstract it as a vehicle routing problem (VRP). The VRP is a highly significant class of problems in the field of operations research. It involves determining the optimal routes for vehicles to travel from a depot to various customers, while satisfying certain constraints. The VRP has numerous practical applications, including logistics distribution, public transportation, waste collection, and mail delivery, etc.. By optimizing vehicle routes, it can conserve resources, enhance efficiency, and reduce pollution and congestion. The classical VRP can be represented as a directed or undirected network graph, where vertices represent each delivery point and depot, and edges connect vertices, representing the cost of travelling between vertices, typically is the distance or time between vertices.

1.4 Thesis overview

In this paper, we developed a joint optimization method for relocation and routing strategies to explore the following questions:

- What factors should be considered in the decision of the relocation strategy in e-scooter sharing system, in other words, how do different parameters affect the decision and result of the relocation strategy?

- On which route should the relocation strategy be implemented, and how will different vehicle routes affect the result of the relocation?

To this respect, we improved the mixed linear programming(MILP) model proposed in [31]. We introduced parameters such as distance matrix, revenue per

trip and relocation cost, to optimize the relocation strategy with maximizing the total profit as the objective function. And we demonstrate the relationship between profit, relocation strategy and different parameters in a case study.

In addition, we also proposed two new MILP models to solve the VRP considering the maximum routing time constraint. One model finds the optimal route while satisfying all designated relocation tasks, thus considering a sufficiently large number and capacity of relocation trucks, having an objective function to minimize the route cost. In the other model, the number and capacity of the dispatch trucks are limited, but they are allowed to selectively complete the relocation tasks. Accordingly, we introduce the revenue for completing relocation tasks, with the objective function of maximizing profit, to incentivize trucks to complete as many relocation tasks as profitable.

Eventually, our joint optimization algorithm is able to propose optimized relocation strategies and the corresponding vehicle routes for e-scooter sharing system.

The thesis is organized as follows:

In Chapter 2, we will review the current research related to relocation strategies and the VRP. In Chapter 3, we will introduce in detail the variables and parameters of the mathematical models, and their formulation and implementation. In Chapter 4, we will verify the model with three study cases, perform sensitivity and scalability analysis, and discuss the impact of each parameter. In Chapter 5, the model will be applied to a practical case and the results obtained will be discussed. Finally, in Chapter 6, we will summarize the major work and conclusions of this paper and indicate the shortcomings and further study directions.

Chapter 2

Literature review

Although the subject of this thesis is e-scooter sharing systems, there is a paucity of relevant literature, so interesting studies including those on car-sharing and bike-sharing services are also reviewed.

As mentioned in Section 1.1, the vehicle re-balancing issue has become more prominent. In recent years, researchers have proposed various optimization models, algorithms, and strategies to tackle this challenge. Studies related to VRP can provide a reference for the model building in this thesis, so a review has also been conducted.

2.1 Relocation strategy

2.1.1 Carrier-based relocation

Huang et al. [25] focus on bike-sharing systems. First, a density-based clustering method is used to partition the region into multiple clusters to reduce computational complexity. Then, using rental and return predictors, car rental and return events at each station are simulated through a Poisson process, and calculating the service level of each station and the number of bikes that need to be relocated at each station. Finally, a Monte Carlo tree search-based Dynamic Repositioning (MCDR) method has been developed to decide the optimal route for relocation trucks.

Zhang et al. [26] have developed a dynamic repositioning model that incorporates traffic equilibrium, inventory balancing, and temporal constraints, while leveraging data-driven neural networks (NN) to forecast the demand for bike sharing. To compute the solution, researchers employed a hybrid metaheuristic approach combining an adaptive genetic algorithm (AGA) and a granular tabu search (GTS) algorithm.

In [27] the concept of 'beautificator' was introduced with a focus on the repositioning of scooters within short distances (even just a few meters) to correct improper and disorderly parking by users. An integer linear programming model was proposed to mathematically represent the problem of joint scheduling and selection of beautification actions as well as traditional relocation operations. A new matheuristic combining a genetic algorithm with large variable neighborhood searches based on exact mathematical programming techniques was proposed to solve the model.

In [32] a novel modular and multi-stage decision-making tool is proposed, which divides the general relocation problem into three independent decision stages, where the output of each stage serves as the input of the next. The first stage is the prediction module, which predicts the maximum inventory imbalance for each vehicle-sharing area in the short future. The selection module of the second stage provides the total number of vehicles to be relocated between each feeder-receiver pair as output. Finally, the schedule module takes a set of relocation processes as input and schedules a sequence of relocation tasks to implement these processes. The method has been shown to have good scalability.

Jin et al. [33] proposed a simulation framework, focusing on the optimization of large-scale bike-sharing systems. The framework consists of a simulation module and an optimization module. The simulation module simulates the daily state changes and system operations at each station, where bike borrowing and returning are generated by using a non-homogeneous Poisson process (NHPP) to create bike pick-up and drop-off demands, and bike and dock failures are considered. Refers to the optimization module, using optimization Model P as an example, they employ a hybrid heuristic method that combines an enhanced k-means clustering method (EKM) and an Ant Colony Optimization (ACO) algorithm to generate routes and operational decisions for each bike.

2.1.2 User-based relocation

Losapio et al. [28] designed E-scooter Balancing DQN(ESB-DQN), a multi-agent system based on Deep Reinforcement Learning (Deep RL) capable of proposing convenient alternative locations for picking up or returning e-scooters. Monetary incentives are offered to encourage users to accept the proposing. Similarly, Yun et al. [34] proposes a deep reinforcement learning (RL) framework, in which it formulates an agent as the service operator to determine the incentives and the e-scooter sharing system as an environment.

Sasaki et al. [29] focused on one-way car-sharing system with reservation requirements and achieved real-time updates of trip demand through online calculation. They proposed an optimization model based on queuing theory, and it took into account three considerations: 1) the origin; 2) the destination; and 3) the departure

time. For users' desired departure and destination points during booking, the system administrator solved the optimization problem to determine the recommended combination of pick-up and drop-off locations and departure time. The discount amount based on the Vickrey-Clarke-Groves (VCG) mechanism designed for incentive measures. The robustness of the system was demonstrated in terms of efficiency, strategy-proofness, individual rationality, and coalition rationality, proving its ability to deal with false statements. Finally, a numerical simulation of a car-sharing station in Japan was conducted, and the results showed that the proposed method could effectively prevent deviations in the number of vehicles and had the lowest failure rate, meeting more demand. A similar strategy was also proposed by Sasaki et al. [29], and applied to the electric car sharing system.

Stokkink et al. [30] focused on car-sharing systems with reservation requirements and aimed to incentivize customers to relocate vehicles by offering discounts. However, the study also considered the current state of the system and a system state prediction model based on a Markov chain to forecast future demand. An adaptive learning algorithm was developed to estimate unknown user preferences and dynamically adjust the optimal values of incentives and the best pick-up and drop-off locations for the user.

Wang et al. [35] focuses on electric car-sharing and proposes an adaptive joint relocation model combining electric car-sharing with bike-sharing. Mixed integer linear programming (MILP) models were built respectively for user-based vehicle and bicycle relocation, taking into account factors such as electric vehicle charging status, cycling distance, station status, and user order history trends. Subsidies were introduced to encourage users to use bicycles for vehicle relocation, thereby optimizing the distribution of shared electric vehicles. The genetic algorithm (GA) was utilized to minimize total relocation costs in solving the model.

Yang et al. [36] focusing on electric bike-sharing, they also takes into account the battery status. To achieve the aggregation of low-power electric bicycles at certain stations and reduce the route cost of battery-swapping trucks, this paper proposes to incentivize selected users according to their destination to choose low-power electric bicycles. The paper describes the problem as three modules based on integer programming: the swapping station selection module, user relocation module, and truck routing module. Additionally, a heuristic algorithm based on variable neighborhood search (VNS) is proposed to solve this problem.

Zhang et al. [37] proposes a bi-level programming model for the repositioning of docked bike-sharing systems. Based on the frequency of a station as an origin or destination, the upper-level model outputs the proportion of trips that need to be reallocated while enforcing station balance as a constraint. The lower-level model takes the proportion as a constraint and seeks to maximize user benefits by producing a reallocation matrix, which indicates the proportion of bikes that should be redistributed between each unbalanced station. Furthermore, the lower-level

model's output can serve as feedback to the upper-level model when the system's status changes, allowing for real-time optimization of bike repositioning.

2.1.3 Carrier-based and user-based hybrid relocation

Many researchers have combined the advantages of the two type of strategies and developed a hybrid relocation strategy.

A two-stage integer stochastic programming model is proposed by Pantuso et al. [38] for an on-demand rental car-sharing system, i.e allowing users to rent cars without prior reservation. The model considers the uncertainty of customer preferences, described by probability distributions. A common pricing scheme is used, in which the final price is determined by the per-minute fee and the return fee. The per-minute fee is fixed, while the return fee may vary for each target period and each origin-destination pair, and may be negative to encourage desired car movements and increase demand. The model determines the return fee that will be applied in the target period, takes actions on prices to influence customer demand, and conducts proactive relocation accordingly to maximize expected profits. The proposed exact solution algorithm is based on the integer L-shaped method. Finally, the method is extensively tested on a case study based on the city of Milan. The results show that the proposed method increases demand by about double and yields higher expected profits.

Similarly, Liu et al. [39] addressed the system with dynamic demand and travel time, as well as uncertain user preferences. They formulated the problem as a Markov decision process. A data-driven approach was proposed, which used online collected user preference data to instantaneously calibrate their relocation willingness, and an online module developed through Bayesian learning to learn the preference model for further enhancing the expected decision. The data results showed that the proposed method could meet the rapid response needs in real situations and improve the quality of the solution.

Xu et al. [40] focus on bike-sharing systems. A hybrid rebalancing strategy is proposed and described as a mixed integer linear programming model. Some surplus bicycles are designated as 'lucky bikes', and users who relocate these lucky bikes to designated locations are rewarded with incentive rewards. Meanwhile, ordinary bicycles do not provide incentive rewards and need to be redistributed by the re-balancing truck fleet. Based on an improved adaptive diversity control hybrid genetic search algorithm (HGSADC), a hybrid genetic algorithm (HGA) is proposed to solve the problem.

G.Guo et al. [41] considers non-linear elastic demand and congestion effects, proposing a mixed Basket-Chandy-Muntz-Palacios (BCMP) queuing network that takes into account charging delay, resource availability, and user wait time. A joint optimization framework is presented, where nested queuing networks are used to

describe system behavior. A joint relocation strategy is determined with queue stability control and wait time constraints, maximizing system commercial value as the objective.

2.2 Vehicle Routing Problem(VRP)

Considering different additional constraints, VRP has given rise to many variants. The different VRP variants have been extensively discussed and classified into sixteen (16) categories in [42]. For instance, based on the varying capacity constraints of the vehicles, there are the Capacitated Vehicle Routing Problem (CVRP), where all vehicles possess identical capacity, and the Heterogeneous Fleet Vehicle Routing Problem (HFVRP), where there exist multiple vehicle types, each with distinct fixed and variable costs and capacities. Based on customers' time window requirements, there are the Vehicle Routing Problem with Time Windows (VRPTW), where access to each customer is required within a predetermined time window. Furthermore, considering the delivery and pick-up operations during the route execution, there are the Vehicle Routing Problem with Backhauls (VRPB) and the Vehicle Routing Problem with Pick-Up and Delivery (VRPPD). In this thesis, the problem is modeled as a VRPPD with capacitated vehicles and maximum travel time constraint.

2.2.1 Capacitated VRP (CVRP)

The research on CVRP primarily focuses on optimizing and improving its solving algorithms. Altabeeb et al. [43] proposes an improved Firefly Algorithm, CVRP-FA. By embedding two types of local search algorithms, improved 2-opt and 2-h-opt, the proposed algorithm is able to expedite convergence towards the optimal solution. Moreover, they introduced PMX crossover and two mutations operators to prevent the fireflies from converging to a local optimum. Additionally, the Taguchi method is used to determine the optimal values of the parameters of the proposed algorithm. Subsequently, the researchers further improved the Firefly Algorithm in [44] and proposed a cooperative hybrid firefly algorithm,CVRP-CHFA, which incorporates multiple FA populations to enhance the quality of solutions and computational efficiency.

İlhan et al. [45] proposes an improved simulated annealing algorithm with crossover operator, named ISA-CO. The algorithm employs a population-based simulated annealing approach and applies Partially Mapped Crossover (PMX) and Order Crossover (OX) operators to solutions in the population. A hybrid selection method is adopted to ensure a balance between exploitation and exploration. ISA-CO is tested on 91 instances with varying numbers of demand points and vehicles.

Sajid et al. [46] addresses the bi-objective CVRP problem by minimizing both the total distance traveled by the fleet and the longest route distance. To tackle this problem, the paper proposes a novel giant tour best cost crossover (GTBCX) operator and a NSGA-II-based routing algorithm employing GTBCX. The algorithm is tested on 88 CVRP benchmark instances.

Máximo et al. [47] introduces a hybridization of a novel adaptive version of Iterated Local Search with Path-Relinking(AILS-PR) to the CVRP, which utilizes adaptive strategies to guide the perturbation intensity and acceptance criteria, thus ensuring the diversity during the search process. Furthermore, by combining with Path-Relinking (PR), the algorithm achieves further enhancement in solution quality and computation time, which is approved on 100 benchmark CVPR instances.

2.2.2 VRP with Time Windows (VRPTW)

D.Wu et al. [48] focuses on the delivery of fresh agricultural products and investigates the time-varying split delivery green vehicle routing problem with multiple time windows. This is accomplished by utilizing a satisfaction metric based on time windows and an economic cost metric, with the objective of minimizing the total economic cost while maximizing the average customer satisfaction level. To address this problem, a variable neighborhood search combined with a non-dominated sorting genetic algorithm II (VNS-NSGA-II) is designed.

B.Pan et al. [49] studied the route problem for goods delivery in urban transportation, while taking into account time-varying travel time, multiple trips, time windows, and maximum travel duration constraints. The researchers modeled the problem as a multi-trip time-dependent vehicle routing problem with time windows (MT-TDVRPTW). To solve this model, they designed a hybrid metaheuristic algorithm that utilizes adaptive large neighborhood search (ALNS) for guided exploration and variable neighborhood descent (VND) for intensive exploitation.

R.Kuo et al. [50] concerned the collaborative form between trucks and unmanned aerial vehicles (UAVs), where trucks can not only provide services to customers but also act as "launch pads" for UAVs. The UAVs can be launched to serve customers and then recovered at rendezvous points. Simultaneously, the hard time window constraints were taken into account, as well as some realistic features of UAV routes, such as limitations on vehicle capacity, UAV battery life, and the setting time for launching and recovering UAVs at rendezvous points. A mixed-integer programming (MIP) model was developed, and a VNS-based heuristic with novel chromosome representation was proposed to solve the problem.

The constraint of time windows can be soft or flexible. In [51] a multi-objective vehicle routing problem with flexible time windows (MOVRPFlexTW) was proposed. In this problem, logistics companies aim to save distribution costs by sacrificing customer satisfaction. A novel Pareto-based multi-objective approach was also

introduced, which combines the ant colony optimization (ACO) algorithm with a mutation operator to solve the problem.

Sarbijan et al. [52] presents the real-time collaborative feeder vehicle routing problem (RTCFVRP) with flexible time windows. The feeder vehicle routing problem (FVRP) consists of a heterogeneous fleet of vehicles, including trucks and motorcycles. A Pareto-based bi-objective method was developed, which combines the multi-objective particle swarm optimization (MOPSO) and the variable neighborhood search (VNS) algorithms.

X.Sun et al. [53] considers flexible time windows and order release dates simultaneously. And proposes a mixed-integer linear programming model and six types of valid inequalities to enhance the model. Additionally, according to the model and the inequalities, a branch-and-cut algorithm (B&C) is proposed to compute the solution.

2.2.3 VRP with Pick-Up and Delivery (VRPPD)

Berbeglia et al.[54] presents a summary and classification of studies on the Pickup and Delivery Problem (PDP). The author categorizes PDP into three main types based on the route structures. These include the Many-to-Many (M-M) problems, which involves multiple origins and destinations where any location may be a source or destination for multiple goods; the One-to-Many-to-One (1-M-1) problems, where some goods need to be delivered from one warehouse to multiple customers while others need to be collected from multiple customers and transported back to the warehouse; and the One-to-One (1-1) problems, where each good has a single origin and destination and must be transported between these locations. The VRPPD is classified under the One-to-One problem, where each request specifies the size of goods to be transported as well as the pickup and delivery locations, and vehicles must visit pickup locations before delivery locations, and the depot cannot request or supply any goods. In this thesis the problem is modelled as VRPPD, which focuses on the collection of e-scooters from certain surplus zones and delivery to zones with e-scooter shortages, which closely mirrors real-world e-scooter relocation scenarios.

Nagy et al. [55] considers the VRPPD for multiple depots and proposes an integrated heuristic approach. The approach first uses standard VRP techniques to generate a weakly feasible solution. The solution is then turned into a strongly feasible one using route modification routines. Finally, maintaining the strong feasibility, the routes can be further improved. The proposed heuristic approach is approved of providing high-quality solutions for VRPPD problems with up to 5 depots and 249 customers within a matter of seconds.

In [56], the VRPPD problem is further classified into three types: Delivery-First Pickup-Second VRPPD, where delivery requests are processed first followed by

pickup requests; Mixed Pickup and Delivery VRPPD, where there is no priority between delivery and pickup, and both types of requests are served in a mixed sequence; and Simultaneous Pickup and Delivery VRPPD, where both pickup and delivery requests are served simultaneously, this type of problem is specifically referred to as VRPSPD, and customers are only visited once. The authors discuss the VRPSPD with Hard Time Windows (VRPSPDHTW) and develop a heuristic algorithm to minimize the waiting time between nodes as the model objective.

H.Zhang et al. [57] considers VRPPD with multiple arrival time and traffic congestion constraints, where different batches of goods arrive at the depots in different time windows, and the truck can only pick up the goods after they have arrived. Moreover, varying traffic congestion results in significant differences in vehicle speeds during different time windows, leading to different delivery times. To address this problem, the authors propose a Dynamic Memory Memetic Algorithm(DMMA).

Q.Chen et al. [58] aims to address the issue of unpaired VRPPD with split deliveries involving multiple commodities. In this problem, pickup and delivery requests are not pre-matched, multiple vehicles can serve a customer, and supply and demand points are not separated, as a demand point for one product may be a supply point for another. The authors propose a variable neighborhood search heuristic for solving this problem.

Gutiérrez-Jarpa et al. [59] considers the VRP with Delivery, Selective Pickup, and Time Windows (VRPDSPTW), in which a fleet of vehicles must fulfill all delivery requests while adhering to time windows, but may choose not to satisfy all pickup requests. Customers may have delivery requests, pickup requests, or both; each customer may be visited once or multiple times; and visits to pickup and delivery vertices may be sequential or unordered. The authors propose an exact Branch-and-Price algorithm that can handle these multiple variants, and yield optimal solutions for the instances with up to 50 customers.

2.3 Research group works

The study in this thesis has been carried out on the basis of the work done by the group SmartData@Polito.

In [60], the authors focus on car-sharing systems, using a variety of machine learning methods to make long-term and short-term predictions about the spatio-temporal patterns of user demand for cars.

In [61, 62] and [63], the authors aim to optimize the distribution of charging stations in the context of the Free Floating Electric Car Sharing system. The former proposes a data-driven approach to optimize the placement of charging stations while taking into account complex customer habits. The latter compares

two different vehicle charging infrastructures: centralised charging centres located in highly dynamic areas of the city and distributed charging posts around the most used areas, using a data-driven mobility demand model developed to investigate the performance and cost of fleet charging management. Using Turin, Milan, New York City and Vancouver as case studies, it is shown that the distributed infrastructure performs better in terms of both trip fulfilment and charging relocation costs.

[64] shows the impact of system design options, such as the number of charging posts, their allocation and the number of cars. The scalability of the overall system is also investigated. The results show that concentrating charging stations in key locations can help optimize car distribution in cities to better intercept demand. The charging infrastructure must intuitively grow in proportion to the demand for mobility. And fleet size grows much more slowly, showing some good economies of scale gains.

[65] compared the performance of possible Free Floating Car Sharing (FFCS) systems using internal combustion engine vehicles (ICEV) or electric vehicles (EV) in three main areas: meeting demand, emissions and system profitability. The results show that an EV FFCS system can meet the same demand as an ICEV-based solution. The EV fleet has fewer emissions, reducing them by more than 50%. However, there are also higher costs. The use of inexpensive, low-power chargers can reduce maintenance costs to some extent.

[66] implements a data-driven mobile demand model based on a Poisson process in the time domain and kernel density estimation in space to spatially characterise users' habits towards different shared services. The authors show the importance of correctly optimizing fixed bandwidth KDEs, in addition to presenting variable bandwidth KDE methods. The advantages of variable KDEs are also illustrated.

[67] takes a shared electric scooter system as objective and analyses reactive and active relocation algorithms to understand how they affect the system, both in terms of performance and cost. The results show that proactive operations are generally superior to reactive operations and, in particular, can keep the number of deployed electric scooters low or even reduce them, making the overall system more profitable and sustainable.

[31] focuses on FFCS systems. Different optimization methods for improved repositioning strategies considering demand forecasts for multiple relocation steps are analysed. The methods proposed are based on dynamic programming, linear programs, mixed integer linear programs and combinations of both, respectively. Dynamic programming method, which is theoretically able to find the best solution, has very little scalability. MILP-based method proves to be the best performing method, while LP-based method always provides good performance in a relatively short time. The author also discuss the impact of look-ahead horizon on optimization performance, showing that a foresighted strategy outperforms a greedy one, with a two-hour look-ahead horizon improving performance by about 7%.

Chapter 3

Methodology

In this chapter, we presents in details the two phases of the joint optimization for the relocation and routing strategies in the development of the e-scooter sharing system.

3.1 Problem description

Due to user behavior, the distribution of scooters tends to be unbalanced. Therefore, operators need dispatch trucks to pick up a certain number of scooters from zones with a surplus of scooters and deliver them to zones where the demand is not met. As the demand for scooters changes frequently over time, an area with high demand in one time period may have low demand in the next period. Therefore, in this thesis, we consider multiple dispatch trucks and take into account expected future demand to avoid shortsightedness, as well as the impact of customers' usage on the system. For simplicity, we assume that time is discrete and divided into time frames.

The joint optimization for the relocation and routing strategies consists of two consecutive phases, the schematic diagram of the optimization process is shown in Figure 3.1. We focus on the dynamic repositioning problem in the first phase, the relocation strategy is to be optimized, each truck only takes scooters from one pick-up area and brings them to one drop-off area, that is, matching couples of zones, without considering the specific routes of the dispatched trucks. In the second phase, the output from the first phase, including a set of zones and the corresponding relocation operations (positive in case of drop off, negative in case of pick up), is used as input of the VRP, to obtain the optimal truck routes. The red vertex represents the depot, the yellow vertexes are the source zones(where the scooters are picked up), and the blue vertexes represent the destination zones(where the scooters are dropped off). The arrows indicate the dispatch truck routes, where

the dispatch trucks depart from the depot with an empty load, visit the relocation zones and perform the corresponding relocation operations. Once the dispatch trucks have completed all the relocation tasks or reached the designated maximum routing time, they return to the depot. The number of trucks is also optimized in this second phase.

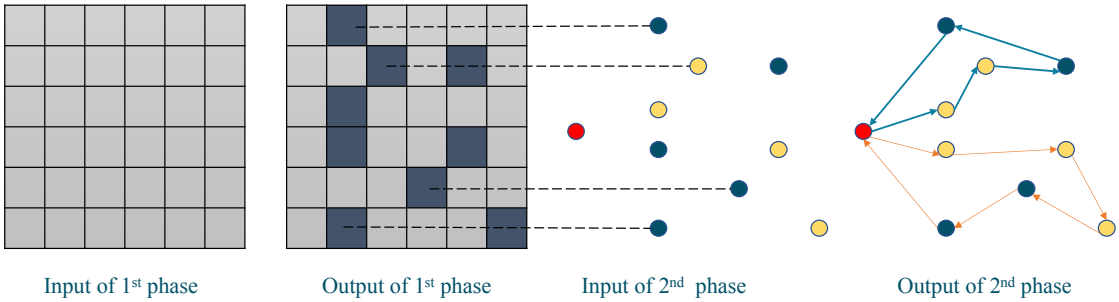

Figure 3.1: Optimization process

3.2 Mathematical model

3.2.1 The First Phase - Repositioning problem

In this phase the problem is formulated as a Mixed Integer Linear Programming (MILP) model. The notation, the mathematical formulation and the implementation is presented.

Let $Z = \{0, ..., z\}$ be the set of city zones, 0 be depot. Let $A = \{(i, j) : i, j \in Z, i \neq 0, j \neq 0, i \neq j\}$ be the set of arcs between two zones. Let P be the number of optimization horizon, i.e. the number of time-frames the optimal relocation operation is to be found, and T be the number of look-ahead horizon, i.e. the number of time-frames over which the decisions are made. As shown in Figure 3.2, over the set of time-frames $\mathcal{P} = \{0,1, ..., P-1\}$, the optimization procedure is performed once for each $p \in \mathcal{P}$, while in each optimization procedure, the relocation decisions is made on considering the demand of future time-frames up to T. As such, the set of time-frames involved in the p-th optimization procedure is $\mathcal{T}_p = \{p, p+1, ..., p+T-1\}$.

Variable definition Other variables and notation are introduced as follows:

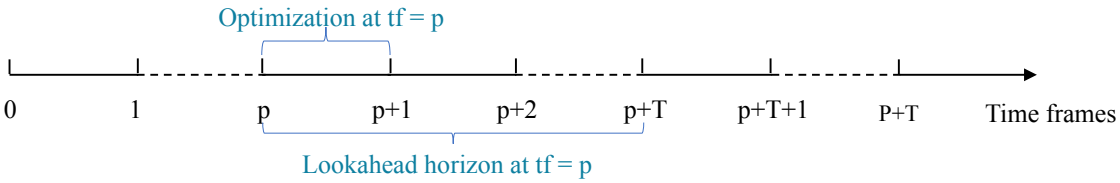

Figure 3.2: Optimization horizon and look-ahead horizon

- $r_{ti} \in \mathbb{Z}$: relocation operation in zone i during time-frame t,

$$\begin{cases} r_{ti} > 0 & \text{if e-scooters are dropped off to zone i} \\ r_{ti} < 0 & \text{if e-scooters are pick up from zone i} \\ r_{ti} = 0 & \text{if no relocation performed in zone i} \end{cases}$$

  $\forall i \in Z \setminus \{0\}, \forall t \in \mathcal{T}_p$

- $w_{tij} \in \mathbb{N}$: number of scooters be relocated from zone i to zone j during time-frame t, $\forall (i,j) \in A, \forall t \in \mathcal{T}_p$
- $s_{ti} \in \mathbb{R}^+$: the number of scooters in zone i at the beginning of time-frame t, $\forall i \in Z \setminus \{0\}, \forall t \in \mathcal{T}_p$
- $EO_{ti} \in \mathbb{R}^+$: the number of scooters leaving zone i during time-frame t, $\forall i \in Z \setminus \{0\}, \forall t \in \mathcal{T}_p$
- $EI_{ti} \in \mathbb{R}^+$: the number of scooters arriving in zone i during time-frame t, $\forall i \in Z \setminus \{0\}, \forall t \in \mathcal{T}_p$
- $O_{ti} \in \mathbb{N}$: the number of outgoing trip requests from zone i during time-frame t, $\forall i \in Z \setminus \{0\}, \forall t \in \mathcal{T}_p$
- $I_{ti} \in \mathbb{N}$: the number of incoming trip requests to zone i during time-frame t, $\forall i \in Z \setminus \{0\}, \forall t \in \mathcal{T}_p$
- $p_{ij}^t \in \mathbb{R}^+$: the probability of a scooter to be moving from zone i to zone j during time-frame t, $\forall (i,j) \in A, \forall t \in \mathcal{T}_p$
- $N \in \mathbb{N}$: the total number of scooters in the system
- $Nt \in \mathbb{N}$: the total number of trucks in the system
- $Ct \in \mathbb{N}$: the capacity of each truck
- $d_{ij} \in \mathbb{R}^+$: the distance between zone i and zone j
- $I \in \mathbb{R}^+$: average revenue per trip

- $C_m \in \mathbb{R}^+$: unit mileage cost
- $C_n \in \mathbb{R}^+$: unit cost per relocated e-scooter
- Decision variable:

$$x_{tij} = \begin{cases} 1 & \text{if arc (i, j) is traversed by a truck during time-frame t} \\ 0 & \text{else} \end{cases}$$

, $\forall (i,j) \in A, \forall t \in \mathcal{T}_p$

Notice that in reality, all variables related to the number of scooters involved should ideally take integer values. However, solving integer linear programming models that involve hundreds of integer variables can be prohibitively expensive, especially when the number of vehicles, zones, and look-ahead horizon increases, leading to an exponential increase in the number of variables. Computationally expensive models are neither economically feasible nor practical. Therefore, the integer constraints on s_{ti}, EI_{ti} and EO_{ti} have been relaxed to expedite the model's solution time.

Assumption Additionally, in this phase, we assume that:

- Dispatched trucks are all homogeneous, they have the same capacity and speed, etc.
- Each dispatched truck travels from at most one pick-up zone to at most one drop-off zone, i.e. each truck is matching one couple of zones.
- The cost of arriving at the pick-up zones and of leaving the drop-off zones are not considered.
- The possibility for multiple or no visits to each zone.
- For each effective trip, a constant average profit is gained.
- E-scooters are uniformly distributed in city zones in the initial system state.

Model $M1$

$$\max_{r_{ti}} \quad \sum_{t\in\mathcal{T}_p}\sum_{i\in Z} EO_{ti} \times I - C_m \times \sum_{t\in\mathcal{T}_p}\sum_{i\in Z}\sum_{j\in Z} x_{tij} \times dij - C_n \times \sum_{t\in\mathcal{T}_p}\sum_{i\in Z}\sum_{j\in Z} w_{tij} \quad (3.1)$$

subject to

$$s_{ti} = s_{t-1,i} + EI_{t-1,i} - EO_{t-1,i} + r_{t-1,i}, \forall i \in Z \setminus \{0\}, \forall t \in \mathcal{T}_p \tag{3.2}$$
$$0 \leq EI_{ti} \leq Iti, \forall i \in Z \setminus \{0\}, \forall t \in \mathcal{T}_p \tag{3.3}$$
$$0 \leq EO_{ti} \leq Oti, \forall i \in Z \setminus \{0\}, \forall t \in \mathcal{T}_p \tag{3.4}$$
$$\sum_{i \in Z} r_{ti} = 0, \forall t \in \mathcal{T}_p \tag{3.5}$$
$$\sum_{i \in Z} s_{ti} = N, \forall t \in \mathcal{T}_p \tag{3.6}$$
$$s_{ti} \geq 0, \forall i \in Z \setminus \{0\}, \forall t \in \mathcal{T}_p \tag{3.7}$$
$$-r_{ti} \leq s_{ti}, \forall i \in Z \setminus \{0\}, \forall t \in \mathcal{T}_p \tag{3.8}$$
$$s_{ti} + \frac{EI_{ti} + r_{ti}}{2} \geq EO_{ti}, \forall i \in Z \setminus \{0\}, \forall t \in \mathcal{T}_p \tag{3.9}$$
$$EI_{ti} = \sum_{j \in Z} p^t_{ji} * EO_{tj}, \forall i \in Z \setminus \{0\}, \forall t \in \mathcal{T}_p \tag{3.10}$$
$$\sum_{t \in T_p} \sum_{i \in Z} x_{tii} = 0 \tag{3.11}$$
$$w_{tij} \leq Ct \times x_{tij}, \forall (i,j) \in A, \forall t \in \mathcal{T}_p \tag{3.12}$$
$$w_{tij} \geq 0, \forall (i,j) \in A, \forall t \in \mathcal{T}_p \tag{3.13}$$
$$\sum_{j \in Z} w_{tji} - \sum_{j \in Z} w_{tij} = r_{ti}, \forall (i,j) \in A, \forall t \in \mathcal{T}_p \tag{3.14}$$
$$\sum_{i \in Z} \sum_{j \in Z} x_{tij} \leq Nt \tag{3.15}$$

The objective function (3.1) aims to maximize the total profit within the look-ahead horizon $\mathcal{T}_p$, which is determined by three components: revenue from effective trips, costs of traveling, and costs of relocation operations. The revenue from effective trips is paid by users who actually use the scooters, which is proportional to the number of effective trips. The costs of traveling are determined by the cost of traveling between relocation zones, which is proportional to the distance between zones. The costs of relocation operations refer to the cost of pick-up and drop-off operations of the scooters, which is proportional to the number of scooters that are relocated.

Constraint in Equation (3.2) represents the state transition for each zone, where the current state at the beginning of time-frame t is determined by the starting state in the previous time-frame, incoming and outgoing trips, and the relocation of vehicles by the dispatch trucks.

Constraints (3.3) and (3.4) require that the number of vehicles arriving and leaving each zone during each time-frame must be positive and cannot exceed the demand for incoming and outgoing trips.

Equation (3.5) ensures that the relocation operations are complete within each time-frame. A negative value of $r_t i$ means that vehicles are picked up and a positive value of $r_t i$ means that vehicles are dropped off. In other words, all vehicles been picked up are dropped off then to other zones within each time-frame.

Constraints (3.6) and (3.7) define the total number of vehicles in the system and ensure the number of vehicles in each zone is always non-negative.

Constraint (3.8) ensures that the number of vehicles picked up from a zone does not exceed the initial number of vehicles in that zone. This constraint only applies when $r_{ti} \leq 0$, i.e., when vehicles are picked up. When $r_{ti} \geq 0$, i.e., when vehicles are dropped off, no constraints are necessary regardless of the initial availability of the zone.

Constraint (3.9) restricts the upper bound of the number of vehicles leaving each zone to be no more than the sum of the corresponding initial availability and the average number of vehicles arriving at and relocated to the zone.

Equation (3.10) introduces a probabilistic constraint to balance the incoming and outgoing vehicles in each zone. The number of vehicles moving from zone i to zone j must reflect the estimated probability from the flow data, in order to avoid the model maximizing the number of satisfied trips unrealistically by greatly enhancing convenience flows.

Equation (3.11) requires that there should be no arcs with the same start and end zones, meaning that trucks cannot depart and arrive at the same zone.

Constraint (3.12) ensures that the relocation operation between zones can only be achieved when the dispatching trucks are in transit between the two zones ($x_{tij} = 1$), and subject to the capacity constraint of the trucks.

Constraints (3.13) and (3.14) indicate that (w_{tij}) has the same absolute value

with r_{ti}, and it is always positive.

Constraint (3.15) states that the total number of arcs should not exceed the number of dispatching trucks, with the assumption that each dispatch truck travels from at most one pick-up zone to at most one drop-off zone.

3.2.2 The Second Phase - VRP

In this second phase, we focus on the VRP, taking the output of the first phase as input and giving the optimal vehicle route that satisfies the relocation operation. We present two MILP models of the VRP. The notation and the implementations are presented.

Let $r_i \neq 0 \in \mathbb{Z}$ be the optimized relocation operation in zone i. Let $Z' = \{1, ..., z'\}$ be the set of zones where relocation operations are performed in phase 1. Let $Z' = S \cup P$ where $S = \{1, ..., n\}$ be the set of source zones and $D = \{n+1, ..., z'\}$ be the set of destination zones, i.e. $r_i > 0$ for $\forall i \in D$, and $r_i < 0$ for $\forall i \in S$.

Variable definition Other variables and notation in this phase are introduced:

- $r_i^s = -r_i$: number of e-scooters picked up from zone i, $\forall i \in S$
- $r_i^d = r_i$: number of e-scooters dropped off to zone i, $\forall i \in D$
- K: the set of trucks
- $C \in \mathbb{N}$: the capacity of each truck
- $L_i^k \in \mathbb{N}$: the load of truck k when leaving zone i, $\forall i \in Z', \forall k \in K$
- $B_i^k \in \mathbb{R}^+$: the time of truck k arriving in zone i, $\forall i \in Z', \forall k \in K$
- $T \in \mathbb{R}^+$: maximum routing time
- $Sr \in \mathbb{R}^+$: service time per scooter
- $Tr \in \mathbb{R}^+$: travel time parameter (inverse of truck speed)
- Decision variable:

$$x_{ij}^k = \begin{cases} 1 & \text{if arc (i,j) is traversed by truck k} \\ 0 & \text{else} \end{cases}$$

, $\forall (i, j) \in A, \forall k \in K$

Assumption Additionally, in this phase, we assume that:

- Dispatched trucks are all homogeneous, they have the same capacity and speed, etc.
- Each dispatched truck departs from the same depot empty and finally returns to the depot empty.
- The task of relocation is indivisible, meaning that all the e-scooters within a zone must be picked up or dropped off in their entirety during the relocation process.

Model M2.1

The Model $M2.1$ is based on the classical Pickup and Delivery Traveling Salesman Problem (PDTSP), see, e.g., Parragh [68]. In which one or more dispatch trucks depart from a depot, with a free initial load. Each zone is visited exactly once, and every unit picked up is used to satisfy every delivery customer's requirement.

The model is given as follows:

$$\min \quad \sum_{k\in K}\sum_{i\in Z'}\sum_{j\in Z'} x_{ij}^k \times d_{ij} \tag{3.16}$$

subject to

$$x_{ii}^k = 0 \quad \forall i \in Z', \forall k \in K \tag{3.17}$$

$$\sum_{k\in K}\sum_{j\in Z'} x_{ij}^k = 1 \quad \forall (i,j) \in A \tag{3.18}$$

$$\sum_{j\in Z'} x_{0j}^k \leq 1 \quad \forall k \in K \tag{3.19}$$

$$\sum_{i\in Z'} x_{i0}^k \leq 1 \quad \forall k \in K \tag{3.20}$$

$$\sum_{i\in Z'} x_{ij}^k = \sum_{i\in Z'} x_{ji}^k \quad \forall (i,j) \in A, \forall k \in K \tag{3.21}$$

$$0 \leq L_i^k \leq C^k \quad \forall i \in Z', \forall k \in K \tag{3.22}$$

$$L_i^k \geq r_i^s \times x_{ij}^k \quad \forall i \in S, \forall j \in Z', \forall k \in K \tag{3.23}$$

$$L_i^k \leq C^k - r_i^d \times x_{ij}^k \quad \forall i \in D, \forall j \in Z', \forall k \in K \tag{3.24}$$

$$x_{ij}^k = 1 \Rightarrow L_j^k = L_i^k + r_j^s \quad \forall i \in Z', \forall j \in S, \forall k \in K \tag{3.25}$$

$$x_{ij}^k = 1 \Rightarrow L_j^k = L_i^k - r_j^d \quad \forall i \in Z', \forall j \in D, \forall k \in K \tag{3.26}$$

$$0 \leq B_i^k \leq T \quad \forall i \in Z', \forall k \in K \tag{3.27}$$

$$B_j^k \geq B_i^k + Sr \times r_i + Tr \times d_{ij} \quad \forall (i,j) \in A, \forall k \tag{3.28}$$

The objective function (3.16) minimizes the routing cost, which is proportional to the distance of the path traversed by dispatch trucks

Equation (3.17) requires that there should be no arcs with the same start and end zones, meaning that trucks cannot depart and arrive at the same zone.

Equation (3.18) ensures that each zone is visited exactly once.

Constraints (3.19) and (3.20) present that one dispatch truck departures from the depot and returns to the depot at the end of its route. Not every truck has to be used.

(3.21) is the flow conservation equation. Every truck arriving in a zone leaves.

Constraint (3.22) states the lower and upper bound of the total load of the truck, which has to be, for sure, positive and smaller than the maximum capacity of the truck.

Constraints (3.23) and (3.24) restrict further the lower and upper bound of the truck load considering the intermediate process of relocation. In one case, the truck is leaving the source zone i after the pick-up operation is completed, therefore the total load of the truck must be no less than the relocated number of scooters in zone i. In the other case, the truck is leaving the destination zone i after the drop-off operation is completed, thus the total load of the truck must be no greater than the maximum capacity of the truck subtracts the relocated number of scooters in zone i. These constrains are non-linear, they can be linearized with the "big M" formulation, obtaining:

$$L_i^k \geq r_i^s - (1 - x_{ij}^k) \times M \quad \forall i \in S, \forall j \in Z', \forall k \in K \tag{3.29}$$

$$L_i^k \leq C^k - r_i^d + (1 - x_{ij}^k) \times M \quad \forall i \in D, \forall j \in Z', \forall k \in K \tag{3.30}$$

Constraints (3.25) and (3.26) are the load transition equations. The indicator functions are used to ensure that the relocation operation between zones can only be achieved when the trucks are in transit between the two zones ($x_{ij}^k = 1$). They can also be linearized with the "big M" formulation, obtaining:

$$L_j^k - (L_i^k + r_j^s) \leq (1 - x_{ij}^k) \times M \quad \forall i \in Z', \forall j \in S, \forall k \in K \tag{3.31}$$

$$(L_i^k + r_j^s) - L_j^k \leq (1 - x_{ij}^k) \times M \quad \forall i \in Z', \forall j \in S, \forall k \in K \tag{3.32}$$

$$L_j^k - (L_i^k - r_j^d) \leq (1 - x_{ij}^k) \times M \quad \forall i \in Z', \forall j \in D, \forall k \in K \tag{3.33}$$

$$(L_i^k - r_j^d) - L_j^k \leq (1 - x_{ij}^k) \times M \quad \forall i \in Z', \forall j \in D, \forall k \in K \tag{3.34}$$

Constraint (3.27) presents the lower and upper bound of the travel time of the truck, which is positive and limited by the maximum routing time T.

Constraint (3.28) states that the truck arrives to the origin before the destination, and time variables must be consistent with the service time in the zone and the travel time between zones. This constraint also eliminates subtours.

Therefore, the model $M2.1$ is given by (3.16)-(3.22), (3.27)-(3.34). In addition, since constraint (3.18) requires each zone to be visited once, and the relocation task within a zone is indivisible, i.e., all relocation needs must be fulfilled. We make the following additional assumptions for the model $M2.1$ to ensure the feasibility of a solution:

- The available number of dispatched truck is infinite.
- The maximum capacity of dispatched truck is greater than the relocation task in any zone, i.e. $C^k > |r_i|, \forall i \in Z', \forall k \in K$.

Model M2.2

For the Model $M2.2$, we relax the constraint (3.18) as follows:

$$\sum_{k \in K} \sum_{j \in Z'} x_{ij}^k \leq 1 \quad \forall (i,j) \in A \tag{3.35}$$

The dispatch trucks can selectively visit zones, so although the relocation task within one zone cannot be split, the total relocation needs can be satisfied partially.

Thanks to the constraint (3.35), the Model $M2.2$ has a larger feasible region and does not require the additional assumptions made for the model $M2.1$. However, the objective function must be adjusted, otherwise, the optimal solution of the original objective function (3.16) will simply be 0, which is not consistent with our purpose. Therefore, we propose an objective function that maximizes the total profit:

$$\max \quad \sum_{k \in K} \sum_{i \in Z} \sum_{j \in Z} x_{ij}^k \times r_j^d \times I - x_{ij}^k \times d_{ij} \tag{3.36}$$

The total profit is obtained by subtracting the route cost from the relocation revenue. We introduce the unit relocation revenue I to incentivize dispatch trucks to fulfill as many relocation needs as possible in profitable situations, while the route cost is still proportional to the distance of the path traversed by dispatch trucks.

Other constraints remain the same with the model $M2.1$, hence, the model $M2.2$ is given by the objective function (3.36), the constraints (3.17), (3.35), (3.19), (3.20), (3.21), (3.22) and (3.27)-(3.34).

3.3 Implementation

3.3.1 Branch and cut Algorithm

The models are solved using the CBC solver through the optimization software OR-Tools [69] implemented in Python. The CBC (Coin-or branch and cut) is an

open-source solver, which employs a combination of branch and bound and cutting plane methods to solve integer linear programming problems.

Branch and Bound (B&B) is a search algorithm commonly used for solving optimization problems. It is primarily utilized for solving integer linear programming problems. In the B&B approach, the optimization problem is decomposed into multiple subproblems, each of which is a local subset of the original problem. The algorithm begins the search from one of these subproblems and uses branching operations to break down the problem into smaller subproblems. Each subproblem is solved separately until the optimal solution is found or it is determined that the problem is infeasible. If the optimal solution is not found, the results of the solved subproblems can be used to prune the search tree, narrowing the feasible solution space and speeding up the search process. The primary idea behind B&B is to select a variable for each subproblem and branch on it, dividing the problem into two subproblems. This process is recursively applied to each subproblem until it is impossible to continue branching or until the optimal solution is found. It is often combined with other optimization algorithms to improve the efficiency and accuracy of the optimization process.

Branch-and-Cut (B&C) algorithm combines B&B with cutting planes to reduce the feasible solution space and improve the efficiency of the optimization process. The main steps of the Branch-and-Cut algorithm are as follows:

1. At the beginning, the integer programming problem is transformed into a linear programming problem, and a feasible solution that satisfies all the constraints is obtained using a linear programming solver.

2. The current solution is checked to see if it is an integer solution. If it is, the algorithm terminates. Otherwise, proceed to step 3.

3. A variable x_i in the integer programming problem is chosen, and it is split into two constraints, $x_i \leq b$ and $x_i \geq b + 1$, which are added to the original problem to form two subproblems. A linear programming solver is used to find feasible solutions for the two subproblems, which become the current solutions for each subproblem.

4. If both subproblems have integer solutions, the subproblem with the smaller objective function value is chosen as the current solution, and the algorithm returns to step 2. Otherwise, proceed to step 5.

5. For the subproblems that do not have integer solutions, a cutting plane is used to find a new constraint, which is added to the subproblem to form a new subproblem. A linear programming solver is used to find a new feasible solution for the new subproblem, which becomes the current solution. If

the new solution satisfies all the constraints, the algorithm returns to step 2. Otherwise, proceed to step 5.

6. Steps 5 are repeated until an integer solution is found.

Overall, the Branch-and-Cut algorithm uses the Branch and Bound algorithm to search for feasible solutions while using cutting planes to strengthen the linear programming relaxation of the integer programming problem. This helps to efficiently reduce the feasible solution space and find optimal solutions to integer programming problems.

3.3.2 Implementation - The First Phase

Given the initial system state and the predicted demand, the set of optimal solutions, including the maximum profit, the optimal relocation strategy, and the corresponding system state s_{ti}, incoming trips EI_{ti} and outgoing trips EO_{ti}, is obtained by the solver for each round of optimisation starting with time-frame $p \in P$. Where s_{ti}, EI_{ti} and EO_{ti} are used to update the system state for the next time-frame. However, as mentioned in Section 3.2.1, they are defined as real numbers to reduce the computational complexity. Hence, the solver may provide non-integer solutions for these variables, which is impractical. Therefore, it is not feasible to use these variables directly for the optimisation of the next time-frame, and further processing is necessary. Two different procedures are employed, the *rounding_algorithm* and the *simulate_system*, which will be described subsequently, and the *optimization_solution* and the *simulation_solution* are obtained respectively, a comparison of the two solution is presented in Chapter **??**. The processed variables are integer values that satisfy the model constraints, and the resulting s_{ti} is the initial system state of the following time-frame. The main procedure of the algorithms are shown in Algorithm 1 and in Algorithm 2.

When employing the *rounding_algorithm*, it is imperative to take into account the constraints specified in Section 3.2.1, as simply rounding each non-integer variable will not suffice. In order to ensure the feasibility of the results, it is also necessary to consider the potential errors that may arise from rounding. Algorithm 3 shows the Pseudo-code of the *rounding_algorithm*.

The *simulate_system* takes the optimized relocation strategy and the current time-frame p as input, and is capable of simulating the effect of the relocation strategy on the system. The simulation system emulates the vehicle movement through a sequence of trips generated based on the origin-destination (OD) matrix of the predicted demand. At the start of the simulation, the number of picked up scooters according to the given relocation strategy is subtracted from the initial system state. Then the first half of the trips are simulated and if a sufficient number of scooters are available at the corresponding origin zones, the demand can be

Algorithm 1 Optimization of relocation algorithm with rounding algorithm.

procedure OPTIMIZER($s_0, \mathcal{T}_p, P$)
    ▷ s_0 is the initial system state
    ▷ $\mathcal{T}_p$ is the involved time-frames
    ▷ P is the optimization horizon
    ▷ $f(s, \mathcal{T}_p)$ is the objective function (3.1)
    ▷ Initialization
    $s \leftarrow s_0$
    ▷ Execution
    for $p = 0, ..., P$ **do**
        ▷ Solve the the objective function $f(s, \mathcal{T}_p)$
        $r, EO, EI \leftarrow f(s, \mathcal{T}_p)$
        ▷ Round EO, EI and update system state
        ▷ Calculate the maximum profit of the time-frame
        $s_{new},\ profit \leftarrow rounding\_algorithm(s, r, EO, EI)$
        $s \leftarrow s_{new}$
    end for
    return $r, profit$
end procedure

Algorithm 2 Optimization of relocation algorithm with simulate system.

procedure OPTIMIZER($s_0, \mathcal{T}_p, P$)
    ▷ Initialization
    $s \leftarrow s_0$
    ▷ Execution
    for $p = 0, ..., P$ **do**
        ▷ Solve the the objective function $f(s, \mathcal{T}_p)$
        $r \leftarrow f(s, \mathcal{T}_p)$
        ▷ Simulate the system and update system state
        ▷ Calculate the maximum profit of the time-frame
        $s_{new},\ profit \leftarrow simulate\_system(r, s, p)$
        $s \leftarrow s_{new}$
    end for
    return $r, profit$
end procedure

satisfied, the trips are effective. Then, the system state is updated by adding the number of dropped off scooters to the corresponding zones. And the second half of trips are simulated following the similar procedure. The profit can be calculated by counting the number of effective trips. Figure 3.3 illustrates the workflow of the *simulate_system*.

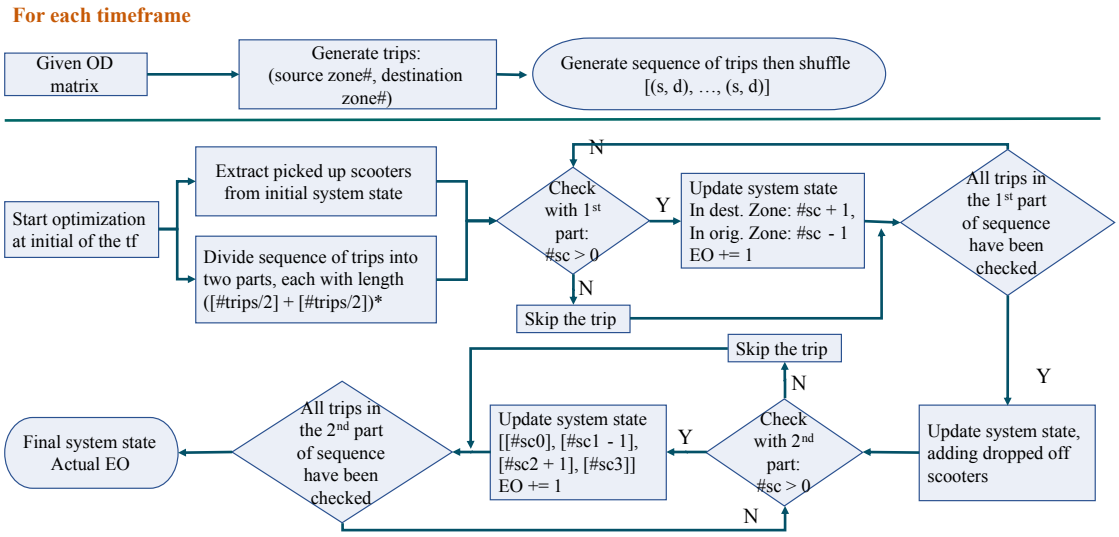

Figure 3.3: The workflow of the *simulate_system*

3.3.3 Implementation - The Second Phase

Before solving the model, it is necessary to process the input data, which is the output from the first phase. As the output of the first phase consists of a set of zones where relocation operations are performed, denoting as $Z' = \{1, ..., z'\}$, along with the corresponding number of e-scooters to be relocated. Notice that $Z' \neq Z$, the former does not include 0, as relocation operations do not involve the depot, and $z' \leq z$, because not all zones require relocation operation. It is essential to match the zone indexes (#), coordinates (x,y), relocation tasks (r), and types (depot/source/destination) one by one to avoid confusion and accurately identify the optimal route in the subsequent process. At the same time, the coordinate of the depot (zone 0) is also defined in this step. The models are also solved with CBC solver, using branch-and-cut algorithm to find the optimal solution.

Algorithm 3 *Rounding_algorithm.*

procedure ROUNDING(s, r, EO, EI)
    ▷ s is the current system state
    ▷ r is the set of optimized relocation strategy
    ▷ EO, EI are the sets of incoming and outgoing trips
    ▷ Round EO, EI to the nearest integer
    $\hat{EO} \leftarrow round\,(EO)$
    $\hat{EI} \leftarrow round\,(EI)$
    ▷ Compute the potential unbalance between incoming and outgoing trips
    $unbalance \leftarrow \sum_i \hat{EO} - \sum_i \hat{EI}$
    ▷ Get the list of indexes of entries which have been rounded
    $indexes\_out \leftarrow get\_rounding\_indexes(EO, \hat{EO}, unbalance)$
    $indexes\_in \leftarrow get\_rounding\_indexes(EI, \hat{EI}, unbalance)$
    ▷ In case of $unbalance > 0$, the incoming trips are more than outgoing trips, therefore, the re-balancing is needed
    while $unbalance > 0$ **do**
        $indexes \leftarrow indexes\_out$
        if $len(indexes) > 0$ **then**
            $index \leftarrow extract\_the\_first\_index(indexes)$
        else
            $index \leftarrow sample(0, ..., Z-1)$
        end if
        $\hat{EO}[index] \leftarrow \hat{EO}[index] - 1$
        $unbalance \leftarrow unbalance - 1$
    end while
    ▷ In case of $unbalance < 0$, there are more incoming trips than outgoing ones, and a similar re-balancing step is required
    while $unbalance < 0$ **do**
        $indexes \leftarrow indexes\_in$
        if $len(indexes) > 0$ **then**
            $index \leftarrow extract\_the\_first\_index(indexes)$
        else
            $index \leftarrow sample(0, ..., Z-1)$
        end if
        $\hat{EI}[index] \leftarrow \hat{EI}[index] + 1$
        $unbalance \leftarrow unbalance + 1$
    end while
    $s \leftarrow s - \hat{EO} + \hat{EI} + r$
    $profit \leftarrow compute\_profit(s, r)$
    return $s, profit$
end procedure

Chapter 4

Model verification and sensitivity analysis

To verify the correctness of the established model and explore the influence of various parameters on the optimization results, we have designed three small-scale toy cases in total: One for the first phase, including 4 zones and 100 shared scooters in the system, 2 scenarios are further considered based on different OD matrices. One for the second phase, two couples of zones having a total number of relocations of 11 scooters are given. And the third one for joint optimization of relocation and routing strategies, which includes 8 zones and 200 scooters, the results of 5 optimization round is presented.

All experiments were conducted on a virtual machine which belongs to the BigData@Polito Cluster[1], with 35 CPU threads and 120 GB of memory reserved.

The time limit for the solver to solve the model was set to 3600 seconds, namely 1 hour, to avoid too long execution time.

4.1 Model verification and sensitivity analysis in toy case 1

In the first phase, the model is supposed to find the optimal relocation strategy which maximizes the total profit, given a distance matrix between zones, the trips demand represented by the origin-destination matrix (OD matrix), the initial system state and other parameters. The designed toy case includes 4 zones with a total of 100 shared e-scooters. The initial state of the system is set as uniformly distributed, which means that the initial number of e-scooters in each zone is set

[1]https://smartdata.polito.it/computing-facilities/

to 25. Different distance matrices between zones are considered, which will be introduced in the following sections.

Two scenarios distinguished by different trips demand, i.e., OD matrices, are considered. In the first scenario, there is the same 50 demand from zones 1, 2, and 4 to zone 3, while no outgoing demand from zone 3. In the second scenario, the outgoing demands between zones were randomly assigned values ranging from 0 to 10. In this toy case, the OD matrix was identical for each time-frame $p \in P$, and Figure 4.1, 4.2 illustrates the OD matrices for one time-frame.

The correctness of the model is verified in Section 4.1.1 under scenario 1. The results included optimization and simulation results obtained through the *rounding_algorithm* and *simulate_system* described in Section 3.3.2.

Furthermore, under each scenario we have carried out the experiments by changing different parameters. Section 4.1.2 and 4.1.3 presented the impact of various parameter changes on the average profit of each time-frame. In most cases, although the optimization and simulation results are different, they have the similar trend in essence. Unless specifically stated, we will use the optimized result as an example for the algorithm sensitivity analysis. Section 4.1.4 showed the effect of various parameter changes on the running time of the algorithm.

O/D zones	1	2	3	4
1	0	0	50	0
2	0	0	50	0
3	0	0	0	0
4	0	0	50	0

Figure 4.1: OD matrix for scenario 1

O/D zones	1	2	3	4
1	0	3	4	0
2	9	0	9	1
3	4	1	0	2
4	4	2	5	0

Figure 4.2: OD matrix for scenario 2

4.1.1 Model verification

In the first scenario, the outgoing demands are the same for zone 1, 2 and 4, all trips were going to zone 3, namely, a deterministic OD matrix is considered, as shown in Figure 4.1. The setup of Scenario 1 is very particular and simple, allowing us to verify quantitatively the correctness of the model.

The optimization and simulation result of an example experiment are presented in Figure 4.4 and Figure 4.5, with the specific parameter setting shown in Table 4.1.

Notice that the parameters here were consistent with the ones defined in Section 3.3.2: I is the average revenue per trip, C_m and C_n are the unit mileage cost and

unit cost per relocated e-scooter respectively, Ct and Nt refers to the capacity of each truck and the total number of dispatch trucks in the system, T and P are the look-ahead horizon and optimization horizon. And finally the *distance*#0 refers to the distance matrix #0 shown in Figure 4.3.

N_zones	$N\_scooter$	I	C_m	Ct	Nt	C_n	T	P	$Distance$
4	100	10	1	20	3	0.1	2	5	#0

Table 4.1: Parameters setting

Dist. #0	1	2	3	4
1	0	1	√2	1
2	1	0	1	√2
3	√2	1	0	1
4	1	√2	1	0

Dist. #1	1	2	3	4
1	0	1	1	1
2	1	0	1	1
3	1	1	0	1
4	1	1	1	0

Dist. #2	1	2	3	4
1	0	50	50	50
2	50	0	50	50
3	50	50	0	50
4	50	50	50	0

Dist. #3	1	2	3	4
1	0	50	50√2	50
2	50	0	50	50√2
3	50√2	50	0	50
4	50	50√2	50	0

Figure 4.3: Distance matrices in the toy case 1

It can be seen that for each time-frame, a set of optimized relocation operation and optimized profit are presented, as well as the initial system state and the current system state. The initial system state and the current system state are the number of shared scooters in each zone at the beginning and at the end of each time-frame, respectively. The relocation operation consists of the zone index and the specific operation, taking time-frame 0 as an example, the set of optimized relocation operation is [('3', -5.0), ('2', 5.0), ('3', -20.0), ('4', 20.0)], which means that during the time-frame, 5 scooters are picked up from zone 3, 5 scooters are dropped off in zone 2, 20 scooters are picked up from zone 3 and 20 scooters are dropped off in zone 4. Thus a total of 25 scooters are picked up from zone 3 and dropped off in zones 2 and 4 respectively. Such a denotation is able to distinguish the behavior of each truck in the existence of multiple dispatch trucks. In time-frame 0, only 25 scooters are available for relocation due to the limitation of the initial number of scooters in zone 3. In the following time-frames, all three available trucks are used and each relocation operation reaches the maximum capacity of each truck in order to obtain the maximum profit. The relocation path consists of a source zone index and a destination zone index, allowing a clear representation of the zones where the relocation operation are performed. In time-frame 0, the relocation path is [('3','2'), ('3','4')], which are the shorter ones, as expected.

More results were also checked to prove that the model always gives the optimal relocation strategy. Additionally, it is observed that for the optimization and simulation results, the optimized relocation operations are identical, since the

tf#	Initial system state	Current system state	Relocation operation	Relocation Path	Opt_Profit
0	[25, 25, 25, 25]	[25.0, 3.0, 62.0, 10.0]	[('3', -5.0), ('2', 5.0), ('3', -20.0), ('4', 20.0)]	[('3', '2'), ('3', '4')]	615.5
1	[25.0, 3.0, 62.0, 10.0]	[10.0, 20.0, 60.0, 10.0]	[('3', -20.0), ('1', 20.0), ('3', -20.0), ('2', 20.0), ('3', -20.0), ('4', 20.0)]	[('3', '1'), ('3', '2'), ('3', '4')]	570.5857864
2	[10.0, 20.0, 60.0, 10.0]	[10.0, 20.0, 60.0, 10.0]	[('3', -20.0), ('1', 20.0), ('3', -20.0), ('2', 20.0), ('3', -20.0), ('4', 20.0)]	[('3', '1'), ('3', '2'), ('3', '4')]	590.5857864
3	[10.0, 20.0, 60.0, 10.0]	[10.0, 20.0, 60.0, 10.0]	[('3', -20.0), ('1', 20.0), ('3', -20.0), ('2', 20.0), ('3', -20.0), ('4', 20.0)]	[('3', '1'), ('3', '2'), ('3', '4')]	590.5857864
4	[10.0, 20.0, 60.0, 10.0]	[10.0, 20.0, 60.0, 10.0]	[('3', -20.0), ('1', 20.0), ('3', -20.0), ('2', 20.0), ('3', -20.0), ('4', 20.0)]	[('3', '1'), ('3', '2'), ('3', '4')]	590.5857864

Figure 4.4: Relocation strategy and Optimized Profit - optimization result under scenario 1

tf#	Initial system state	Current system state	Relocation operation	Relocation Path	Simu_Profit
0	[25, 25, 25, 25]	[0, 0.0, 100.0, 0.0]	[('3', -5.0), ('2', 5.0), ('3', -20.0), ('4', 20.0)]	[('3', '2'), ('3', '4')]	995.5
1	[0, 0.0, 100.0, 0.0]	[0.0, 0.0, 100.0, 0.0]	[('3', -20.0), ('1', 20.0), ('3', -20.0), ('2', 20.0), ('3', -20.0), ('4', 20.0)]	[('3', '1'), ('3', '2'), ('3', '4')]	590.5857864
2	[0.0, 0.0, 100.0, 0.0]	[0.0, 0.0, 100.0, 0.0]	[('3', -20.0), ('1', 20.0), ('3', -20.0), ('2', 20.0), ('3', -20.0), ('4', 20.0)]	[('3', '1'), ('3', '2'), ('3', '4')]	590.5857864
3	[0.0, 0.0, 100.0, 0.0]	[0.0, 3.0, 97.0, 0.0]	[('3', -20.0), ('1', 20.0), ('3', -20.0), ('2', 20.0), ('3', -20.0), ('4', 20.0)]	[('3', '1'), ('3', '2'), ('3', '4')]	560.5857864
4	[0.0, 3.0, 97.0, 0.0]	[0.0, 0.0, 100.0, 0.0]	[('3', -20.0), ('1', 20.0), ('3', -20.0), ('2', 20.0), ('3', -20.0), ('4', 20.0)]	[('3', '1'), ('3', '2'), ('3', '4')]	620.5857864

Figure 4.5: Relocation strategy and Optimized Profit - simulation result under scenario 1

optimization process of the relocation strategy is not affected by the subsequent algorithms, as shown in Algorithm 1 and 2.

Regarding the profit, the total profit obtained during each time-frame is calculated as follows:

$$profit_t = \sum_{i \in Z} EO_{ti} \times I - C_m \times \sum_{i \in Z} \sum_{j \in Z} x_{tij} \times dij - C_n \times \sum_{i \in Z} \sum_{j \in Z} w_{tij} \tag{4.1}$$

It can be seen that there is a large discrepancy between the optimization results and the simulation results, which is caused by the different calculation methods for the outgoing trips EO_{ti} in *rounding_algorithm* and *simulate_system*. In the former EO_{ti} is subject to the constraint 3.9, which is no more than the sum of the number of scooters in the zone at the beginning of each time-frame and the average

number of vehicles arriving and relocated to the zone. While in *simulate_system* (see Figure 3.3), the demand is represented by two parts of the sequence of trips, and the relocated scooters can only be used to satisfy the second part of the trip sequence. EO_{ti} here is the count of effective trips, and since the order of the trip sequence is random, the number of effective trips is indeterminate. Such as in time-frame 3 in Figure 4.5, there are 33 trip demands from zone 2 to zone 3 within the first part of the trip sequence and 17 in the second part of the trip sequence. Since the initial number of scooters in zone 2 is 0, non of the demand in first part of the trip sequence is satisfied, but only the 17 trip demands in the second part of the trip sequence are satisfied thanks to the 20 relocated scooters, and 3 are stay in zone 2. Therefore in this time-frame the effective trips in zone 2 are 17. Knowing that the effective trips in zone 1, 3, and 4 are 20, 0, 20, respectively, the total profit in this time-frame can be found through equation 4.1:

$$\begin{aligned} profit_{t=3} = (20 + 17 + 0 + 20) \times 10 - 1 \times (1 + 1 + \sqrt{2}) \\ - 0.1 \times (20 + 20 + 20) \approx 560.5858 \end{aligned}$$

The result is consistent with the one proposed by the model.

4.1.2 Sensitivity analysis in scenario 1

By analyzing the results of an example solution, we verified the correctness and effectiveness of the algorithm. More experiments were conducted to explore the effect of different parameters of the model on the optimization results. A summary of the specific settings of the parameters for the experiments under this scenario is shown in Figure 4.6.

Paras	Plotting					
	1	2	3	4	5	6
N_zones	4					
N_scooter	100					
I	10					
Cm	1					
Ct	[0-95]					
Nt	1	[0-4]	1	1	1	1
Cn	0.1	0.1	[0.1-10]	0.1	0.1	0.1
T	2	2	2	[1-4]	2	2
P	20	20	20	20	[5,10,15,20]	20
Distance	#0					#[0-3]

Figure 4.6: Parameters setting under scenario 1

Average profit with different truck capacities The parameter settings for this experiment are shown in Table 4.2:

N_zones	$N\_scooter$	I	C_m	Ct	Nt	C_n	T	P	$Distance$
4	100	10	1	[0-95]	1	0.1	2	20	#0

Table 4.2: Parameters setting - with different truck capacities

Figure 4.7 shows the changing trend of average profit with respect to the maximum truck capacity. Zero maximum truck capacity means, of course, no relocation is able to be performed. At this point, the average profit is 37.5 with an average number of effective trips of 3.75, which is generated by the demand fulfilled by the initial parked e-scooters in the zones. Since it is not possible to perform relocation operations, all scooters end up accumulating in zone 3, while the number of scooters in zones 1, 2, and 4 is 0. At this stage, the system state does not change any more after the first time-frame, and no further travel demand can be fulfilled, leading to no further increase in profit. As the maximum truck capacity increases, the generated average profit increases proportionally. This is due to an increasing number of scooters being allowed to relocate to the required zones, as shown in Figure 4.8, leading to more travel demand being fulfilled and generating more revenue. Although Figure 4.9 indicates that relocation costs also increase proportionally, they are negligible compared to the revenue gained. Until the maximum truck capacity reaches 70, the average number of relocated scooters is 64.55 and the average profit reaches 658.55. Although the truck capacity continues to increase, the number of scooters being relocated no longer significantly increases, as well as the profit. This is because the number of relocated scooters is already sufficient to meet the demand in each zone, and any additional relocated scooters would simply accumulate within the zone without generating new revenue. Therefore, the algorithm no longer schedules more relocation tasks.

Changing number of trucks In this experiment we analyzed the trend of average profit with respect to the maximum truck capacity with changing different number of dispatch trucks. The parameter settings for the experiment are shown in Table 4.3:

N_zones	$N\_scooter$	I	C_m	Ct	Nt	C_n	T	P	$Distance$
4	100	10	1	[0-95]	[0-4]	0.1	2	20	#0

Table 4.3: Parameters setting - with changing number of trucks

From Figure 4.10, it can be observed that the relocation is not possible to

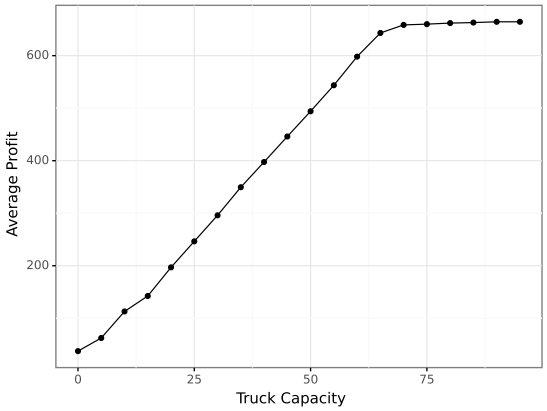

Figure 4.7: Average profit v.s. Ct

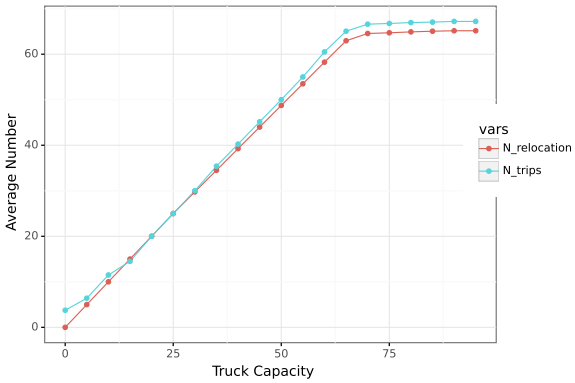

Figure 4.8: Average relocation number v.s. Ct

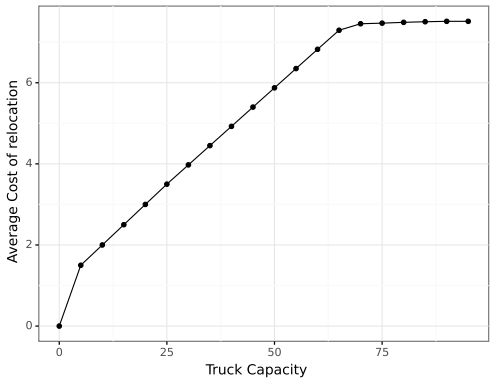

Figure 4.9: Average relocation cost v.s. Ct

perform when the number of trucks is zero, which is similar to the case when the truck capacity is zero in the previous experiment. The curves representing a larger number of trucks reach the saturation faster, indicating that when there are more trucks available, even with small capacity, it can fulfill all demands. The curves for $Nt = 3$ and $Nt = 4$ overlap completely, as only three trucks are employed to carry out relocation tasks despite the availability of four trucks in the system. This is due to the optimal relocation paths are, easy to identify, $zone3 \rightarrow zone1$, $zone3 \rightarrow zone2$, and $zone3 \rightarrow zone4$. The excess trucks are thus idle, which is expected. Figures 4.11 and 4.12 display identical trends in the number of trips and relocation. In Figure 4.13, The relocation costs exhibit fluctuations due to the fact that relocation costs are made up of costs of travelling, which is proportional to the distance and costs of relocation operations, which is proportional to the

number of scooters that are relocated as defined in Section 3.2.1. With more relocation paths passed by multiple trucks incurring more travel costs. Therefore, when truck capacity is sufficiently large, fewer trucks are utilized to reduce travel costs, ultimately resulting in a decrease in total costs.

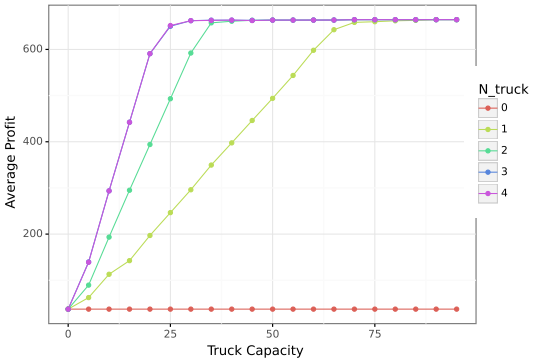

Figure 4.10: Average profit v.s. Nt

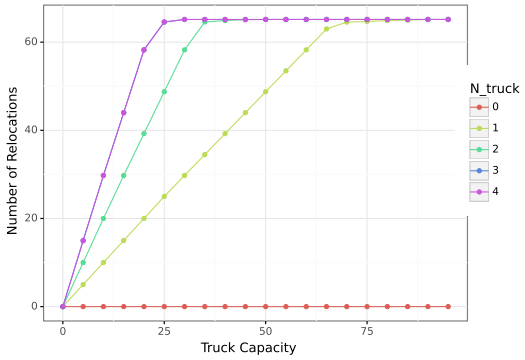

Figure 4.11: Average number of trips v.s. Nt

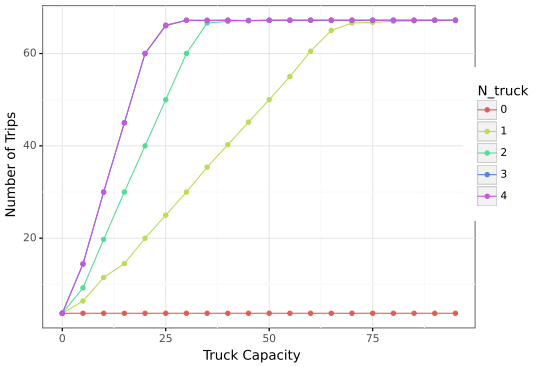

Figure 4.12: Average relocation number v.s. Nt

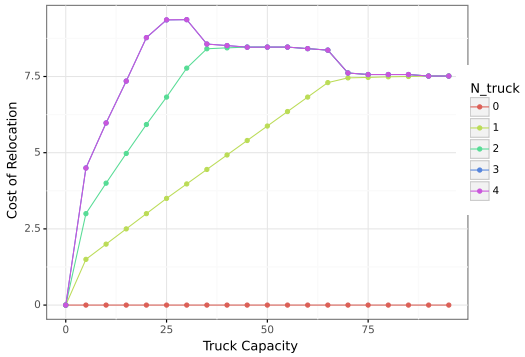

Figure 4.13: Average relocation cost v.s. Nt

Changing unit cost for relocating one scooter The parameter settings for the experiment are shown in Table 4.4:

Several different values are adopted as unit cost per relocation in this experiment. Of course, the higher the unit cost, the lower the profit. It is more interesting to observe how the relocation strategy is impacted by the unit cost. As shown in

N_zones	$N\_scooter$	I	C_m	Ct	Nt	C_n	T	P	$Distance$
4	100	10	1	[0-95]	1	[0.1-10]	2	20	#0

Table 4.4: Parameters setting - with changing unit cost for relocating one scooter

Figure 4.16 and 4.17, the relocation strategy is not affected when the unit cost is 4 or lower, because the benefit gained from relocation is always greater than the cost at this moment. When the unit cost is greater than 4 and less to 10, the number of relocations is restricted to below 50, despite the larger truck capacity, because more relocations would lead to a cost higher than the benefit. While for a unit cost of 10 and higher, the optimal strategy is to not perform relocation, because the revenue per effective trip is exactly 10, at which point the benefit of each relocation is 0 or negative.

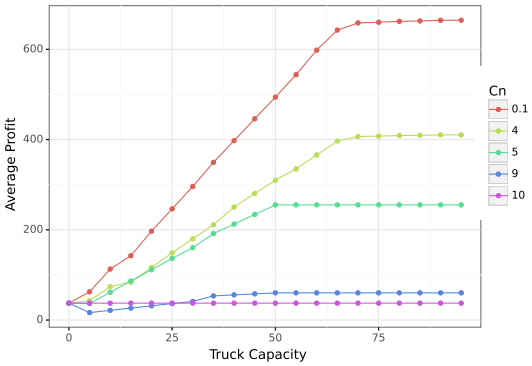

Figure 4.14: Average profit v.s. C_n

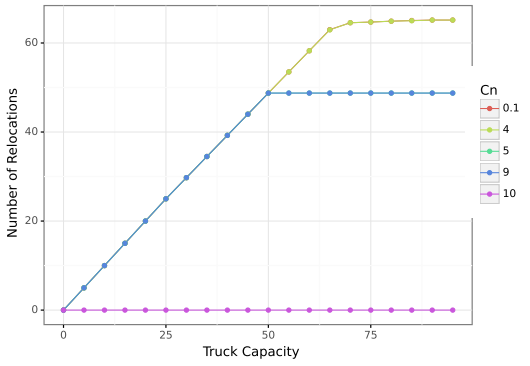

Figure 4.15: Average number of trips v.s. C_n

Changing number of look-ahead horizon With the combination of the constraint 3.9, the specific parameter setting and the particular OD matrix in this scenario, changing look-ahead horizon does not have an obvious impact on the relocation strategy and average profit. The experimental results and detailed explanations are presented in the Appendix A.

Changing number of optimization horizon The optimization horizon is the number of time-frames in which the decision of the relocation strategy is made. In this scenario the OD matrix is identical for each time-frame, so there is no significant change in the relocation strategy and profit in each time-frame. The experimental results are presented in the Appendix A.

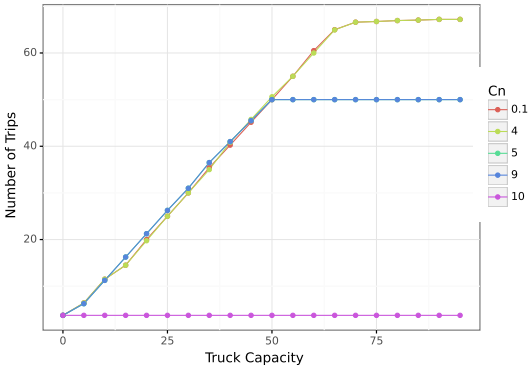

Figure 4.16: Average relocation number v.s. C_n

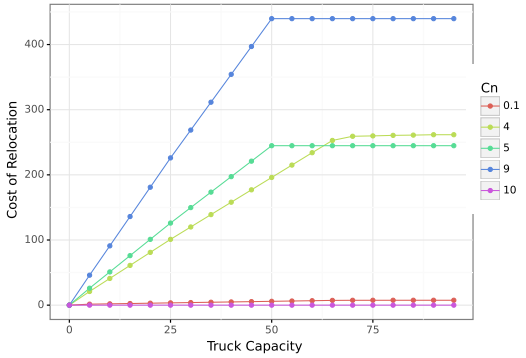

Figure 4.17: Average relocation cost v.s. C_n

Changing distance matrix Changing the distance matrix comes in changing the relocation cost, and similar to the experiment of changing the unit cost per relocated scooter, higher cost leads to lower profit. However, the effect on the relocation strategy in this experiment is reflected in the case of small capacity trucks, when the number of relocated scooters is low, the benefit gained from relocation could be less than the cost of creating the relocation path. Therefore, when the distance between zones increases, relocation operations will only be performed if the capacity of the truck is large enough. The experimental results are presented in the Appendix A.

4.1.3 Sensitivity analysis in scenario 2

In scenario 2, a random OD matrix is adopted, as shown in Figure 4.2. Similar experiments are carried out, with a slightly adjusting on the parameter settings, as shown in Figure 4.18.

Average profit with different truck capacities The parameter settings for this experiment are shown in Table 4.5:

N_zones	$N\_scooter$	I	C_m	Ct	Nt	C_n	T	P	$Distance$
4	100	10	1	[0-50]	1	0.1	2	20	#0

Table 4.5: Parameters setting - with different truck capacities

The trend shows similarity with the first scenario, as shown in Figures 4.19, 4.20, 4.21. Notice that the average number of trips in this scenario is much larger than

Paras	Plotting					
	1	2	3	4	5	6
N_zones	4					
N_scooter	100					
I	10					
Cm	1					
Ct	[0-50]					
Nt	1	[0-5]	1	1	1	1
Cn	0.1	0.1	[0.1-12]	0.1	0.1	0.1
T	2	2	2	[1-4]	2	2
P	20	20	20	20	[5,10,15,20]	20
Distance	#0					#[0-3]

Figure 4.18: Parameters setting under scenario 2

in scenario 1 when the truck capacity is 0, i.e., when no relocation is performed, reaching 22.75 (instead of 3.75), due to the fact that trips in this scenario no longer all have the same destination, but commute between zones, so that a certain number of trips are self-balanced. The curve saturates at a truck capacity of 30, when the average number of relocations is 19.4 and the maximum average profit is 437.06. This is due to the relatively lower total demand in this scenario compared to the previous one, with an average of 44 (instead of 66.6), which allows smaller trucks to meet the total demand.

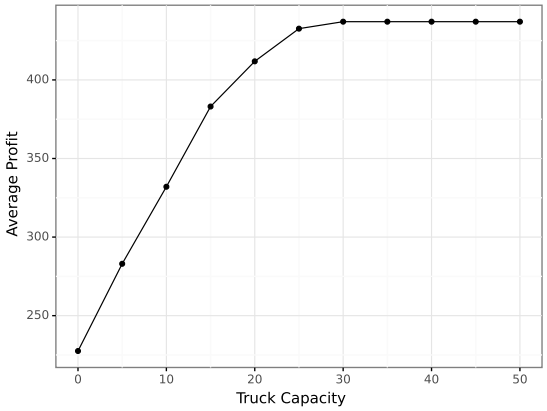

Figure 4.19: Average profit v.s. Ct

Changing look-ahead horizon The parameter settings for this experiment are shown in Table 4.6.

Figure 4.22 shows that increasing the look-ahead horizon raises the average

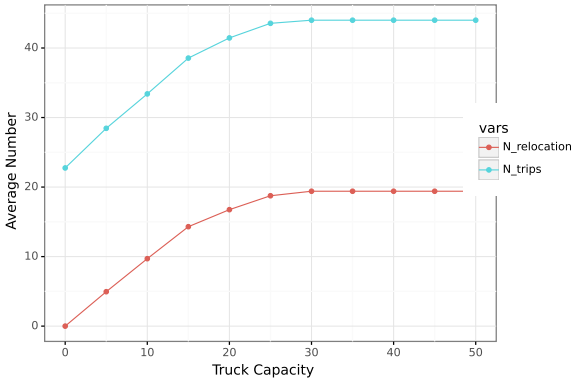

Figure 4.20: Average relocation number v.s. Ct

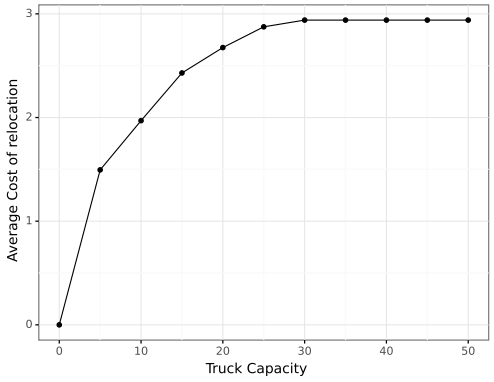

Figure 4.21: Average relocation cost v.s. Ct

N_zones	$N\_scooter$	I	C_m	Ct	Nt	C_n	T	P	$Distance$
4	100	10	1	[0-50]	1	0.1	[1-4]	20	#0

Table 4.6: Parameters setting - changing look-ahead horizon

profit. With a truck capacity of 20, increasing the look-ahead horizon from 2 to 3 raises the average profit by 3.5%, but increasing it to 4 does not further raise profits since the relocation strategy remains unchanged. With a truck capacity of 30 or higher, a look-ahead horizon of 2 or higher has no significant impact on average profit, which is always between 436.98 and 437.06, but both are higher than the average profit of 431.6 with a look-ahead horizon of 1.

The impact of look-ahead horizon on profit and relocation strategy is not as straightforward as the other parameters, it is influenced a lot of other factors such as OD matrix, truck capacity, etc. Therefore, the optimal look-ahead horizon should be customized to the specific scenario and parameter settings.

For the other parameters, number of trucks, unit cost, optimization horizon and distance matrices, the experimental results show similar trends to those in scenario 1, see Appendix A.

4.1.4 Scalability analysis

The scalability are analysed in this section, under both scenario 1 and scenario 2. Specifically, we demonstrate how the elapsed time required to resolve the model is affected by: the number of zones and scooters, the optimization horizon and the look-ahead horizon. The parameter settings are shown in Table 4.7

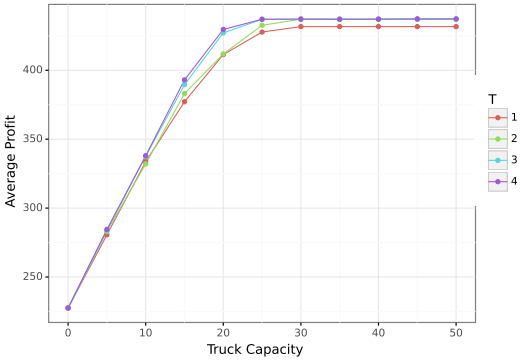

Figure 4.22: Average profit v.s. T

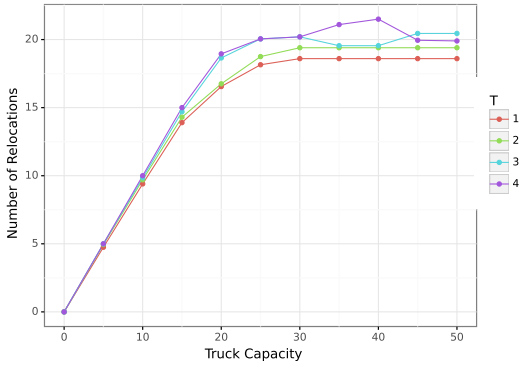

Figure 4.23: Average number of trips v.s. T

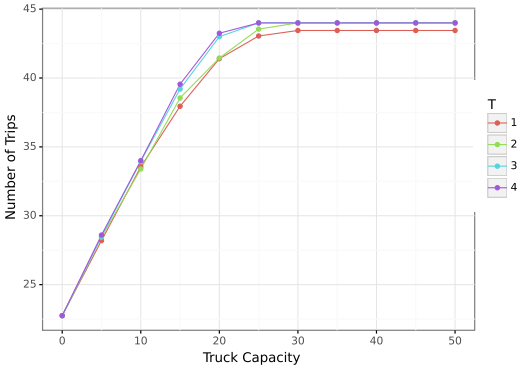

Figure 4.24: Average relocation number v.s. T

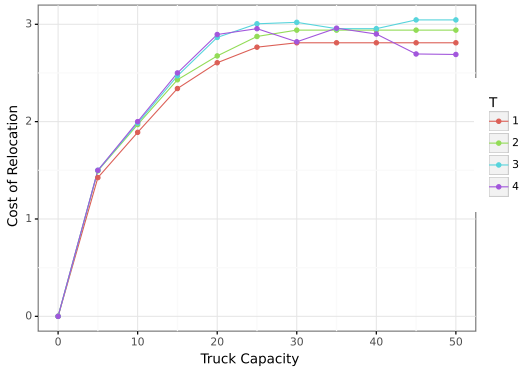

Figure 4.25: Average relocation cost v.s. T

In the first set of parameters, the number of scooters varies with the number of zones, specifically, $N\_scooter = N\_zones \times 25$, to guarantee that in the initial state there are always 25 scooters in each zone. In addition, increasing the number of zones leads to a change in the OD matrix. Fortunately, the deterministic OD matrix considered in scenario 1 is relatively unaffected, regardless of the number of zones, there are always the same 50 trips demand from each zone to one certain zone, from which there is no outgoing demand. Contrary to scenario 1, scenario 2 considers a random OD matrix. In order to obtain a generalized result, for each change in the number of zones, we generate 5 different OD matrices and calculate the average of the 5 elapsed times as the final result.

	N_zones	$N\_scooter$	I	C_m	Ct	Nt	C_n	T	P
Para.1	[4-120]	[100-3000]	10	1	20	1	0.1	[1-4]	20
Para.2	4	100	10	1	20	1	0.1	[1-4]	[1-20]

Table 4.7: Parameters setting for scalability analysis

In Figure 4.26, with the increasing N_zones the running time shows an accelerated increasing trend, and a larger look-ahead horizon will further accelerate the increasing rate of the running time. Ignoring the outliers, solving the optimization for 100 zones with $T = 4$ takes the longest running time, which is 167.262s. However, as shown in Figure 4.27, even though the averaged results are considered, there is considerable fluctuation in the running time, especially when the number of zones and the look-ahead horizon are large. The trend is relatively smoother when $T = 1$, and an accelerated increasing trend in running time, similar to that in scenario 1, is evident in Figure 4.28. It is obvious that for random matrices it is more difficult to find the optimal solution than for a deterministic one. In scenario 2, the average running time for solving the cases with 100 to 120 zones with look-ahead horizon of 1 is 58.22s, which is much larger than the average time for the same cases in scenario 1, 28.27s.

Figures 4.29 and 4.30 show that in both 2 scenarios, the elapsed time increases linearly with the optimization horizon. In scenario 1, performing 20 round of optimization with $T = 4$ can be solved in 0.44 seconds, and also in scenario 2 it takes only 3.49 seconds.

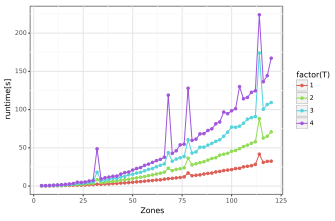

Figure 4.26: Elapsed time v.s. N_Zones in scenario 1

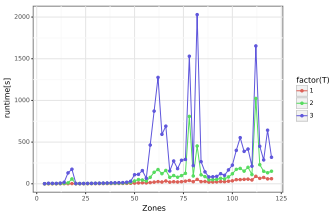

Figure 4.27: Elapsed time v.s. N_Zones in scenario 2

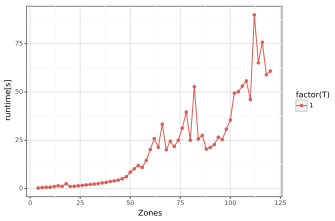

Figure 4.28: Elapsed time v.s. N_Zones with $T = 1$ in scenario 2

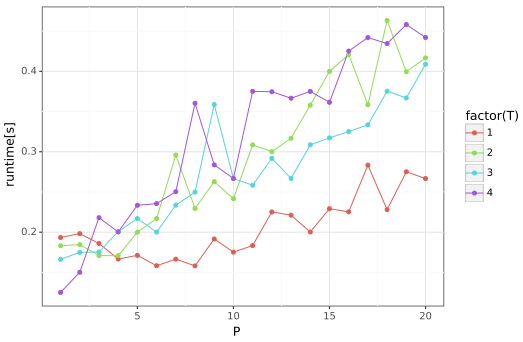

Figure 4.29: Elapsed time v.s. P in scenario 1

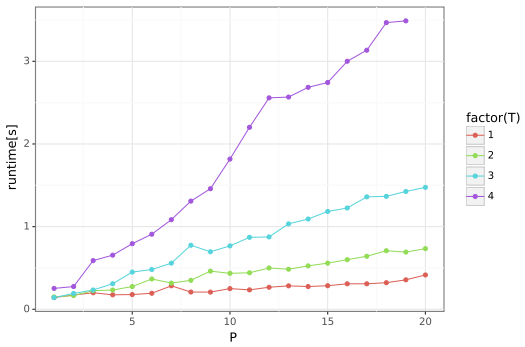

Figure 4.30: Elapsed time v.s. P in scenario 2

4.2 Model verification in toy case 2

In the second phase, we proposed two models 3.2.2 and 3.2.2. Both of them take the optimized relocation strategy proposed by the first phase as input, and find the optimal route of dispatch trucks considering the minimum route cost and maximum total profit respectively. The designed toy case for this phase includes 4 zones and a depot, and they are represented by points in a two-dimensional coordinate system, as shown in Figure 4.31. The input consists of zones indexes and the corresponding relocation tasks, as well as coordinates of each zone, as shown in Figure 4.32. The distance between each zone is given by the Euclidean distance of each point.

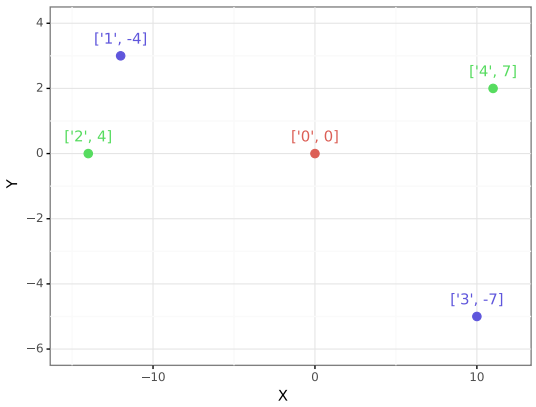

Figure 4.31: 4 zones and a depot

raw_data	X	Y	set
('0', 0)	0	0	depot
('1', -4)	-12	3	source
('2', 4)	-14	0	destination
('3', -7)	10	-5	source
('4', 7)	11	2	destination

Figure 4.32: Inputs of the model

4.2.1 Verification of model M2.1

The model $M2.1$ is adopted firstly to find the optimal route in this toy case, with the specific parameters shown in Table 4.8. Notice that the parameters are consistent with the ones defined in Section 3.2.2: T is the maximum routing time, Sr and Tr refer to service time per scooter and the travel time between zones.

	Ct	Nt	T	Sr	Tr
Para.1	7	2	60	1	1
Para.2	7	2	70	1	1

Table 4.8: Parameters setting for model $M2.1$

The optimal route are shown in Figure 4.33, 4.34. The optimal routes under the two sets of parameters are different, in the second case the optimal route presented has a lower routing cost of 61.85 rather than 86.98 in the first case. This is due to the fact that the maximum routing time in the first case is 60 and one truck is not able to traverse all the zones in the time limit.

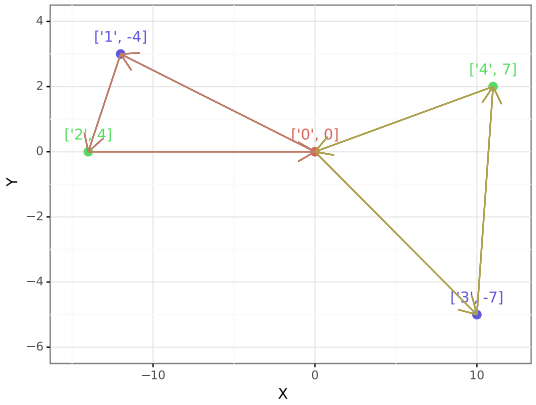

Figure 4.33: Optimal route for $M2.1$ with Para.1

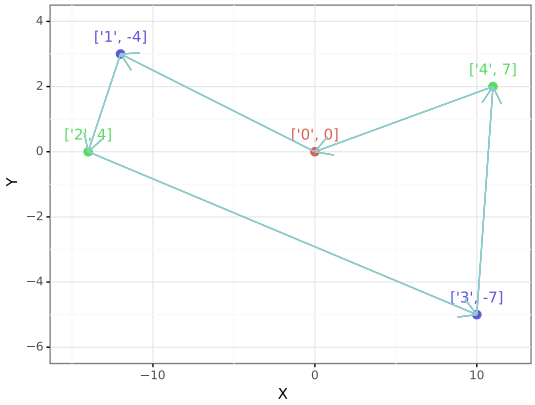

Figure 4.34: Optimal route for $M2.1$ with Para.2

For the model $M2.1$, each zone must be visited exactly once, thus the above two parameter settings are the 'limit case' to guarantee a feasible solution. Trucks with capacity less than 7 would fail to complete the relocation task for zones 3 and 4. Or, a smaller number of trucks or maximum routing time would prevent the more distant zones 3 and 4 from being visited and thus no feasible solution can be found. Nevertheless, the proposed route can always be verified as optimal if there is a sufficient number of available trucks and sufficient truck capacity.

4.2.2 Verification of model M2.2

The model $M2.2$ has a larger feasible region by relaxing this constraint, more experiments with different parameter settings are conducted, as shown in Table 4.9.

	Ct	Nt	T	Sr	Tr
Para.1	5	2	60	1	1
Para.2	10	1	60	1	1
Para.3	10	2	60	1	1
Para.4	10	2	50	1	1
Para.5	10	2	70	1	1
Para.6	15	2	70	1	1
Para.7	15	2	70	2	1

Table 4.9: Parameters setting for model $M2.2$

With the set of parameters of Para.1, the optimal route includes only zones 1 and 2, due to the limitation of the truck capacity (Figure 4.35). In this model, the relocation task in each zone is considered as a whole, and since the relocation tasks in zones 3 and 4 exceed the truck capacity, they are not visited. The routing cost in this case is 36.72, only partial relocation tasks have been completed.

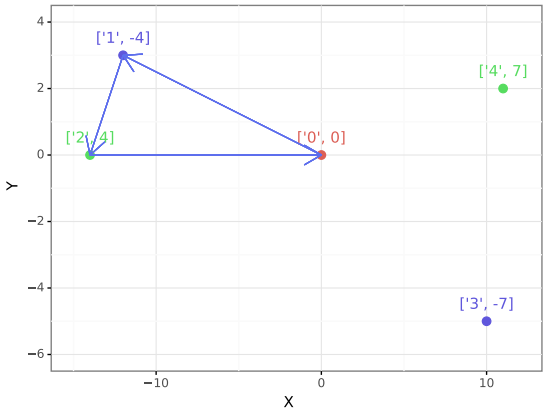

Figure 4.35: Optimal route for $M2.2$ with Para.1

While with Para.2, the truck capacity is large enough, still, only partial relocation needs are satisfied (Figure 4.36). Due to the restrict of maximum routing time, one truck is unable to traverse all zones. Under the premise of not exceeding the maximum routing time, the relocation task of zones 3 and 4 is prioritized as the

higher relocation needs. When two trucks are involved as with Para.3, finally all relocation needs can be met (Figure 4.37).

The solution obtained with Para.4 (Figure 4.38) is same as in the first case, but it is limited by the maximum routing time in this case. The time of 50 is larger than the travel time of visiting zones 1 and 2 (36.72), but smaller than that of visiting zones 3 and 4 (50.26).

By increasing the maximum routing time, the optimal route found with Para. 5 (Figure 4.39) is capable of satisfying all relocation tasks with a relatively low routing cost. And the same solution is proposed with Para.6 (Figure 4.40).

Finally, with Para.7, increasing the unit service time per scooter, zones 3 and 4 are not visited (Figure 4.41). Because a higher number of relocations would require the trucks to stay in the zone too long to return to the depot within the maximum routing time.

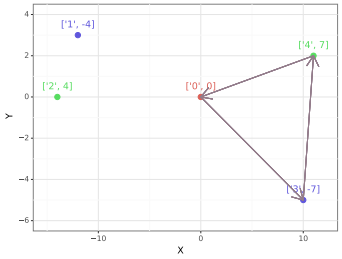

Figure 4.36: $M2.2$ with Para.2

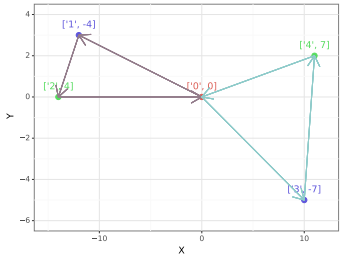

Figure 4.37: $M2.2$ with Para.3

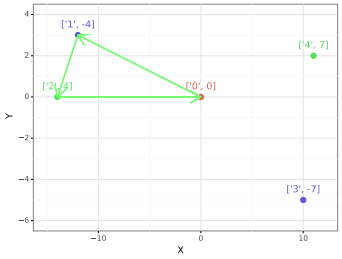

Figure 4.38: $M2.2$ with Para.4

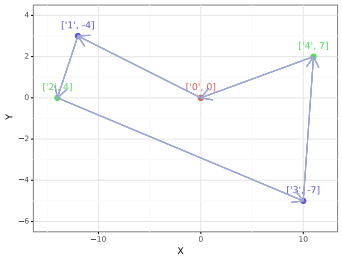

Figure 4.39: $M2.2$ with Para.5

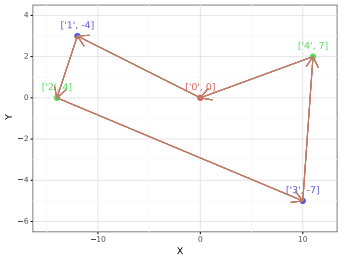

Figure 4.40: $M2.2$ with Para.6

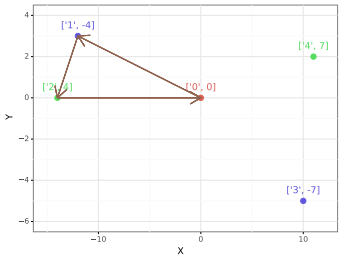

Figure 4.41: $M2.2$ with Para.7

Overall, in any case, the route proposed by model $M2.2$ can always be verified as optimal.

4.3 Joint optimization in toy case 3

In the third case study, we designed a toy case containing 8 zones and a depot, thus a total of 9 zones, whose locations are identified by the coordinates (Figure 4.42). The corresponding distance matrix is calculated and given in Figure 4.43 (presented in a format preserving three decimal digits). Each time frame has identical OD matrix as in Figure 4.44, where the trip demand within zones is not considered and there is no travel demand between the depot and other zones. The number of travel demands between each zone is a random integer within 10, and the total number of trips demand in each time-frame is 255.

The total number of e-scooters in the system is set to 200, and the initial state of the system is set as uniformly distributed, therefore there are 25 e-scooters in each zone at initial. Other parameters are set as in Table 4.10. The number of trucks is set to be 4, which means in each optimization round there are at most 4 couples of zones included in the optimized relocation strategy. Figure 4.45 shows the optimized relocation strategy in each time-frame. Takes that as input, the following phase, employing the model $M2.1$ with parameters as in Table 4.11, it finds the optimal route in each time-frame, as shown from Figure 4.46 to 4.50.

zones	0	1	2	3	4	5	6	7	8
X	0	-6	-11	-15	-6	6	9	13	9
Y	0	-2	5	-4	10	7	-4	-3	-3

Figure 4.42: Zones in scenario 2

Dist.	0	1	2	3	4	5	6	7	8
0	0.000	6.325	12.083	15.524	11.662	9.220	9.849	13.342	9.487
1	6.325	0.000	8.602	9.220	12.000	15.000	15.133	19.026	15.033
2	12.083	8.602	0.000	9.849	7.071	17.117	21.932	25.298	21.541
3	15.524	9.220	9.849	0.000	16.643	23.707	24.000	28.018	24.021
4	11.662	12.000	7.071	16.643	0.000	12.369	20.518	23.022	19.849
5	9.220	15.000	17.117	23.707	12.369	0.000	11.402	12.207	10.440
6	9.849	15.133	21.932	24.000	20.518	11.402	0.000	4.123	1.000
7	13.342	19.026	25.298	28.018	23.022	12.207	4.123	0.000	4.000
8	9.487	15.033	21.541	24.021	19.849	10.440	1.000	4.000	0.000

Figure 4.43: Distance matrix in scenario 2

OD	0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0	0
1	0	0	8	1	2	4	1	5	5
2	0	1	0	0	5	8	9	1	2
3	0	9	5	0	1	6	3	1	4
4	0	8	4	6	0	9	6	2	5
5	0	7	5	1	1	0	5	4	2
6	0	8	7	8	0	8	0	2	0
7	0	6	0	7	5	1	8	0	9
8	0	3	7	7	5	3	8	7	0

Figure 4.44: OD matrix in scenario 2

N_zones	$N\_scooter$	I	C_m	Ct	Nt	C_n	T	P
8	200	10	1	20	4	0.1	2	5

Table 4.10: Parameters setting for $M1$ in scenario 2

Ct	Nt	T	Sr	Tr
40	4	70	1	1

Table 4.11: Parameters setting for $M2.1$ in scenario 2

The average optimized profit on the 5 time-frames is 2516.28, increased by 28.64% relative to 1956, the one whit no relocation strategy performed.

The capacity and number of dispatch trucks of model $M2.1$ are set to be large enough to guarantee the feasible solution. In any case, it can be seen that only 1 truck has been used in phase 2. Moreover, in time-frame 0, the relocation is carried out in 3 couples of zones, and the optimized profit is 2524.82, the dispatch trucks depart from the depot, traverse the 6 zones and complete the relocation tasks, and then return to the depot with a routing cost of 58.34. In time-frame 1, the optimized profit is 2510.29 and the routing cost of 60.06. The relocation strategy includes 4 couples of zones, but in practice only 6 zones are involved. Due to the large outgoing demand in zone 4, which is 80, a large number of scooters are relocated to this zone from multiple source zones. Conversely, zone 5 has a larger incoming demand(39) than outgoing demand(25), so scooters are accumulated in this zone, which allows the relocation from this zone to multiple destination zones. A similar story in next time-frames. Observing the situation in these time-frames, it is clear that the relocation strategy almost always involves zones 4, 5, 7 and 8 and never takes place in zone 3. This is due to the fact that in zones 4, 7 and 8 there is more outgoing demand than the incoming demand, so scooters in these

zones often experience shortages, whereas the opposite is true for zone 5. In zone 3, however, the demand is relatively balanced, 29 trips for outgoing and 30 for incoming, hence no relocation is required.

tf	Relocation operation	Relocation Path
0	[('2', -20.0), ('4', 20.0), ('5', -14.0), ('7', 14.0), ('6', -15.0), ('8', 15.0)]	[('2', '4'), ('5', '7'), ('6', '8')]
1	[('1', -14.0), ('4', 14.0), ('2', -14.0), ('4', 14.0), ('5', -20.0), ('8', 20.0), ('8', -14.0), ('7', 14.0)]	[('1', '4'), ('2', '4'), ('5', '8'), ('8', '7')]
2	[('1', -20.0), ('4', 20.0), ('5', -18.0), ('8', 18.0), ('6', -15.0), ('7', 15.0), ('6', -2.0), ('8', 2.0)]	[('1', '4'), ('5', '8'), ('6', '7'), ('6', '8')]
3	[('1', -20.0), ('8', 20.0), ('2', -20.0), ('4', 20.0), ('8', -13.0), ('7', 13.0)]	[('1', '8'), ('2', '4'), ('8', '7')]
4	[('1', -17.0), ('4', 17.0), ('2', -6.0), ('4', 6.0), ('5', -19.0), ('8', 19.0), ('6', -14.0), ('7', 14.0)]	[('1', '4'), ('2', '4'), ('5', '8'), ('6', '7')]

Figure 4.45: Relocation strategies in scenario 2

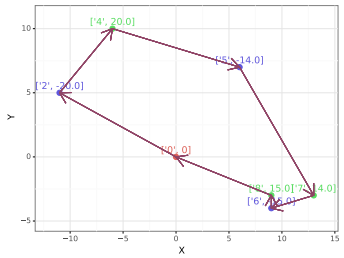

Figure 4.46: Optimal route over time-frame 0 in scenario 2

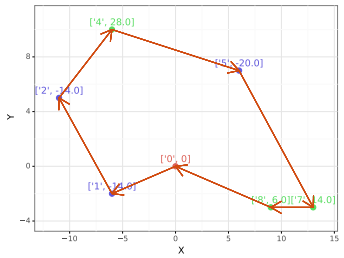

Figure 4.47: Optimal route over time-frame 1 in scenario 2

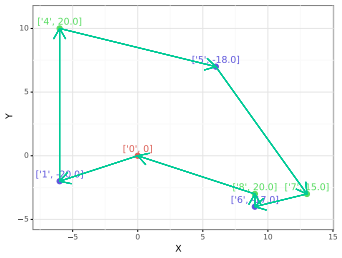

Figure 4.48: Optimal route over time-frame 2 in scenario 2

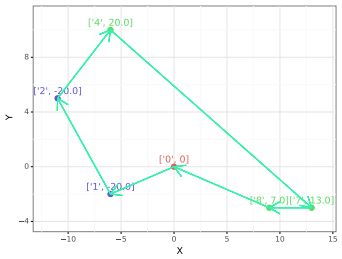

Figure 4.49: Optimal route over time-frame 3 in scenario 2

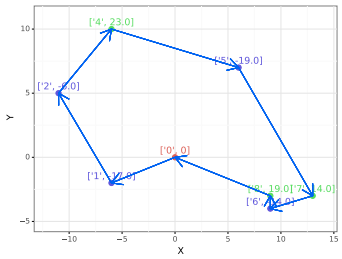

Figure 4.50: Optimal route over time-frame 4 in scenario 2

Chapter 5

Numerical results in practical case

In this chapter, the developed joint optimization algorithm is applied to a practical case, in which a trace of recorded e-scooter trips is collected from the city of Louisville. The dataset is organized and provided by the group SmartData@PoliTO.

5.1 Data processing

5.1.1 Analysis on dataset

The datasets include information about zones and trips which are saved in 3 *csv* files. They are processed using the python packages pandas[1], numpy[2] and scipy[3].

Grid matrix The city of Louisville was discretized into zones by $200m \times 200m$ grids. A total of 1476 zones were obtained and indexed sequentially as shown in Figure 5.1, each zone can be located by the location index, e.g. the location of zone 124 is $(1, 3)$

[1]https://pandas.pydata.org/docs/

[2]https://numpy.org/doc/stable/

[3]https://docs.scipy.org/doc/scipy/

	0	1	2	3	4	5	···	35	36	37	38	39	40
0	0	1	2	3	4	5		35	36	37	38	39	40
1	41	42	43	44	45	46		76	77	78	79	80	81
2	82	83	84	85	86	87		117	118	119	120	121	122
3	123	124	125	126	127	128		158	159	160	161	162	163
4	164	165	166	167	168	169		199	200	201	202	203	204
...	...												...
31	1271	1272	1273	1274	1275	1276		1306	1307	1308	1309	1310	1311
32	1312	1313	1314	1315	1316	1317		1347	1348	1349	1350	1351	1352
33	1353	1354	1355	1356	1357	1358		1388	1389	1390	1391	1392	1393
34	1394	1395	1396	1397	1398	1399		1429	1430	1431	1432	1433	1434
35	1435	1436	1437	1438	1439	1440	···	1470	1471	1472	1473	1474	1475

Figure 5.1: Grid matrix in practical case

Distance matrix The distance matrix represents the calculated geometric distances of each zone, which is a matrix of size 1476×1476. Figure 5.2 shows a partial distance matrix.

	0	1	2	3	4	5	6	7	8	9	10
0	0	200	400	600	800	1000	1200	1400	1600	1800	2000
1	200	0	200	400	600	800	1000	1200	1400	1600	1800
2	400	200	0	200	400	600	800	1000	1200	1400	1600
3	600	400	200	0	200	400	600	800	1000	1200	1400
4	800	600	400	200	0	200	400	600	800	1000	1200
5	1000	800	600	400	200	0	200	400	600	800	1000
6	1200	1000	800	600	400	200	0	200	400	600	800
7	1400	1200	1000	800	600	400	200	0	200	400	600
8	1600	1400	1200	1000	800	600	400	200	0	200	400
9	1800	1600	1400	1200	1000	800	600	400	200	0	200
10	2000	1800	1600	1400	1200	1000	800	600	400	200	0

Figure 5.2: Partial distance matrix in practical case

Trips count The basic format of the trips count data is shown in Figure 5.3. Where the *original_id* and *destination_id* are the indexes of the trips' departure and arrival zones, from 0 to 1475, *hour* is the time at which the trips occurred, from 0 to 23, and *start_time* indicates the number of trips started at this time.

	origin_id	destination_id	hour	start_time
0	4	218	10	1
1	13	216	6	1
2	16	264	20	1
3	17	103	21	1
4	18	58	8	1
5	18	145	22	1

Figure 5.3: Trips count in practical case

In order to understand the data in a more intuitive way, we plot the line graphs

using the plotnine[4] package in python. Figures 5.4 and 5.5 are representing the number of outgoing and incoming trips in each area during the day, as well as a graph representing the number of trips per hour in Figure 5.6.

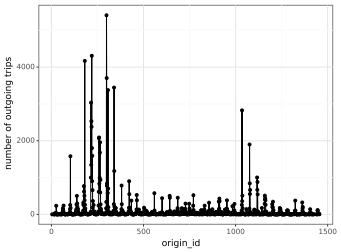

Figure 5.4: Outgoing demand in each zone

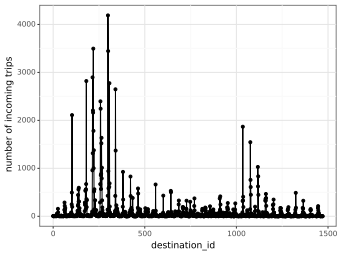

Figure 5.5: Incoming demand in each zone

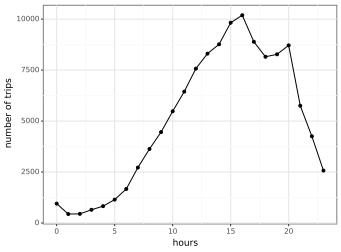

Figure 5.6: Number of trips per hour

It can be noticed that the origins and destinations of trips have a strong consistency, the highly active zones of e-scooter usage are mostly concentrated in zones indexed 200 to 300, and in zones 1100 to 1200 also have relatively high usage. The number of outgoing trips among these zones is around 2000 and more, within zone 299 the largest number of trips even reaches 5408. However the use of shared scooters is rather unevenly distributed spatially. Including the zones mentioned here, there are only 22 zones counting number of outgoing trips more than 1,000 and only 164 zones with outgoing trips more than 100, while more than a half of the zones (799) counts outgoing trip less than 10.

Besides, a certain trend of (in total of outgoing and incoming) trips can be seen in the temporal distribution. From 0:00 to 5:00 the number of trips remaining very small, as people rarely travel during the night, which is intuitive. The number of trips continues to rise during the daytime and peaks at 16:00, after a slight decrease, it reaches a sub-peak at 20:00 and continues to decrease for the rest of the day.

According to the analysed characteristics of the trips, We have performed aggregation of the data over space and time, in order to improve the performance of the algorithm.

5.1.2 Zones aggregation

Spatially the zones are aggregated by 2×2. Starting from zone 0, the four geometrically neighboring zones are aggregated into a new zone and re-assigned

[4]https://plotnine.readthedocs.io/en/stable/index.html

sequentially, as shown in Figure]5.7. Note that due to the odd number of zones in one row, in the last column, 2 zones are aggregated into a new zone. After aggregation, a total of 378 new zones are obtained, assigned the index numbers 0 to 377.

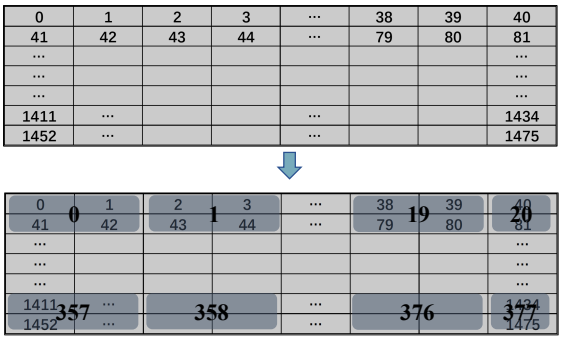

Figure 5.7: Zones aggregation

Due to the presence of many ineffective zones (i.e., in which no or very few trips present), a further filtering step is performed. We first sort the zones according to the number of trips in descending order, and show the relationship of number of zones involved versus the ratio of covered trip demand to the total demand, as in Figure 5.8, and the same step has also been employed on the origin zones, as in Figure 5.9. For the original region, 308 zones need to be involved to cover 90% of the trip demand, while the aggregated zones can cover 91.64% of the trip demand by involving only the first 100 zones.

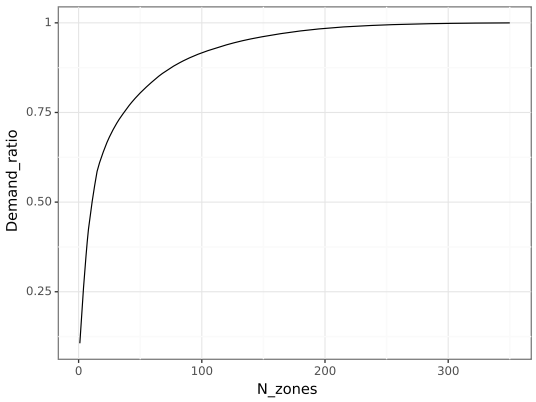

Figure 5.8: Demand ratio v.s. Number of zones – Origin zones

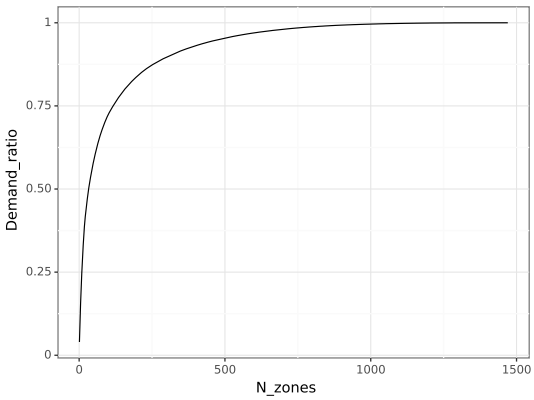

Figure 5.9: Demand ratio v.s. Number of zones – Aggregated zones

5.1.3 OD matrix aggregation

In temporal, we aggregate the 24 hours into 7 time-frames. The number of trips during night hours are very low, hence hours from 0:00 to 5:00 are aggregated as time-frame 0, the trips starts within these hours are simply summed up with their origin and destination zones unchanged. The other hours are aggregated by 3, as shown in Figure 5.10. The trips then are represented by 7 different OD matrices, each with size of 100×100.

time-frame	0	1	2	3	4	5	6
hour	[0,1,2,3,4,5]	[6,7,8]	[9,10,11]	[12,13,14]	[15,16,17]	[18,19,20]	[21,22,23]

Figure 5.10: OD aggregation

5.2 Optimization results

To improve the computing efficiency, the joint optimization for the practical case will involve only the first 100 aggregated zones in the shared e-scooter system, and adopt the 7 aggregated OD matrices.

The parameter setting in this expreiment is shown as in Figure 5.11. In the first phase, the total number of shared scooters in the system is assumed to be 2000, and they are uniformly distributed in the initial state of the system, so that each zone has 20 scooters at initial. The total number of dispatch trucks available within the system is 7, each with a capacity of 20. The average revenue per effective trip is 10€. The unit mileage cost is 0.25€ per kilometer and the unit service cost per relocated scooter is 0.2€.

The algorithm runs for 5 optimization rounds, considering the demand of 2 time-frames for each optimization round. The proposed relocation strategy is shown in Figure 5.12. The average optimized profit on the 5 time-frames is 26926.41€, increased by 6.6% relative to 25260€, the one whit no relocation strategy performed.

paras_relocation
{'N_scooter': 2000, 'N_truck': 7, 'C_truck': 20, 'Incoming': 10, 'Cm': 0.25, 'Cn': 0.2, 'P': 5, 'T': 2}
paras_vrp
{'C_trucks': 120, 'N_trucks': 7, 'T': 3600, 'Sr': 30, 'Tr': 0.18}

Figure 5.11: Parameter setting in practical case

tf #	Relocation operation
0	[('257', -20.0), ('258', 20.0), ('279', -20.0), ('258', 20.0), ('29', -20.0), ('51', 20.0), ('30', -18.0), ('51', 18.0), ('32', -20.0), ('31', 20.0), ('250', -14.0), ('251', 14.0), ('159', -13.0), ('160', 13.0)]
1	[('69', -20.0), ('70', 20.0), ('50', -20.0), ('29', 20.0), ('48', -20.0), ('27', 20.0), ('90', -20.0), ('89', 20.0), ('90', -20.0), ('91', 20.0), ('215', -20.0), ('194', 20.0), ('138', -20.0), ('139', 20.0)]
2	[('48', -20.0), ('27', 20.0), ('90', -20.0), ('89', 20.0), ('90', -20.0), ('91', 20.0), ('73', -20.0), ('52', 20.0), ('111', -20.0), ('132', 20.0), ('236', -20.0), ('215', 20.0), ('182', -13.0), ('183', 13.0)]
3	[('73', -20.0), ('94', 20.0), ('278', -20.0), ('257', 20.0), ('278', -20.0), ('277', 20.0), ('72', -20.0), ('93', 20.0), ('279', -20.0), ('280', 20.0), ('111', -20.0), ('132', 20.0), ('299', -20.0), ('300', 20.0)]
4	[('69', -20.0), ('89', 20.0), ('48', -20.0), ('27', 20.0), ('90', -20.0), ('89', 20.0), ('90', -20.0), ('112', 20.0), ('73', -20.0), ('94', 20.0), ('47', -20.0), ('26', 20.0), ('72', -20.0), ('94', 20.0)]
tf #	Relocation Path
0	[('257', '258'), ('279', '258'), ('29', '51'), ('30', '51'), ('32', '31'), ('250', '251'), ('159', '160')]
1	[('69', '70'), ('50', '29'), ('48', '27'), ('90', '89'), ('90', '91'), ('215', '194'), ('138', '139')]
2	[('48', '27'), ('90', '89'), ('90', '91'), ('73', '52'), ('111', '132'), ('236', '215'), ('182', '183')]
3	[('73', '94'), ('278', '257'), ('278', '277'), ('72', '93'), ('279', '280'), ('111', '132'), ('299', '300')]
4	[('69', '89'), ('48', '27'), ('90', '89'), ('90', '112'), ('73', '94'), ('47', '26'), ('72', '94')]

Figure 5.12: Relocation strategy in practical case

In the second phase, the model $M2.1$ is adopted thus the number and capacity of trucks are set to be sufficiently large. As indicated in Section 5.1.3, each time-frame comprised 3 hours or 5 hours, so a reasonable maximum routing time is set to 3600s, i.e., the relocation tasks need to be completed within 1 hour. The service time per relocated scooter is 30s. Assuming that the average speed of the dispatch truck is 20 $\mathrm{km\,h^{-1}}$, which is around 5.56 $\mathrm{m\,s^{-1}}$, Tr is set to the inverse of this value, i.e., 0.18 $\mathrm{s\,m^{-1}}$. The proposed optimal routes are shown in Figures 5.13 to 5.17.

In time-frame 0, the designated relocation zones are gathered in three orientations far apart. Due to the maximum routing time constraint, four dispatch trucks are employed to complete all the relocation tasks, and the optimal routes are:

Truck 0: 0 -> 32 -> 31 -> 30 -> 29 -> 51 -> 0
Truck 1: 0 -> 257 -> 279 -> 258 -> 0
Truck 2: 0 -> 159 -> 160 -> 0
Truck 3: 0 -> 250 -> 251 -> 0

The total routing length is 14103.87m. Zones 250 and 251 are the farthest from the depot, although they are in the same direction with zones 159 and 160, the relocation time required to visit them in the same path would result in exceeding the maximum time limit, so these two couples of zones are visited by two trucks separately.

In time-frame 1, there are also four dispatch trucks employed. The optimal routes are:

Truck 0: 0 -> 50 -> 29 -> 48 -> 27 -> 0
Truck 1: 0 -> 90 -> 89 -> 69 -> 70 -> 91 -> 0

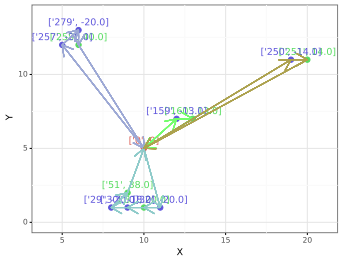

Figure 5.13: Optimal route over time-frame 0 in practical case

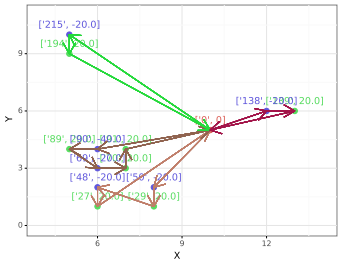

Figure 5.14: Optimal route over time-frame 1 in practical case

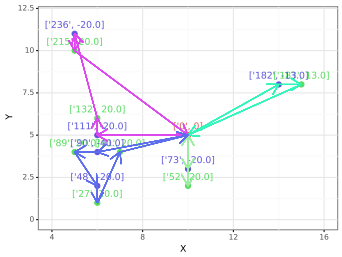

Figure 5.15: Optimal route over time-frame 2 in practical case

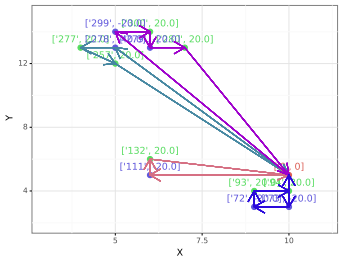

Figure 5.16: Optimal route over time-frame 3 in practical case

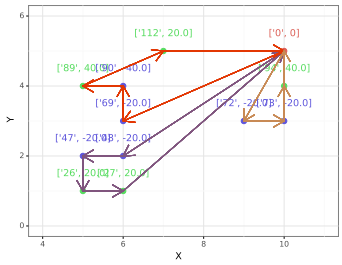

Figure 5.17: Optimal route over time-frame 4 in practical case

Truck 2: 0 -> 215 -> 194 -> 0
Truck 3: 0 -> 138 -> 139 -> 0

The total routing length is 11344.18m.
In time-frame 2, the optimal routes having the total length of 13436.43m, are:

Truck 0: 0 -> 182 -> 183 -> 0
Truck 1: 0 -> 111 -> 132 -> 236 -> 215 -> 0
Truck 2: 0 -> 73 -> 52 -> 0
Truck 3: 0 -> 90 -> 89 -> 48 -> 27 -> 91 -> 0

All the designated 13 zones are visited.
In time-frame 3, the optimal routes with a total length of 13365.50m, are:

Truck 0: 0 -> 278 -> 277 -> 257 -> 0
Truck 1: 0 -> 111 -> 132 -> 0
Truck 2: 0 -> 299 -> 300 -> 279 -> 280 -> 0
Truck 3: 0 -> 73 -> 72 -> 93 -> 94 -> 0

In time-frame 4, 11 zones are involved, served by 3 dispatch trucks. In this case the total length is only 7967.44m.

Truck 0: 0 -> 48 -> 47 -> 26 -> 27 -> 0
Truck 1: 0 -> 69 -> 90 -> 89 -> 112 -> 0
Truck 2: 0 -> 72 -> 73 -> 94 -> 0

Chapter 6

Conclusions

This thesis proposes a joint optimization of carrier-based relocation and routing strategies for e-scooter sharing system. We develop a mixed integer linear programming (MILP) model to optimize the relocation strategy. The objective function of the model is to maximize total profit, which is obtained by subtracting the total revenue from satisfied trips from the mileage cost and the repositioning cost, considering predicted demand, number and capacity limitation of dispatched trucks, look-ahead horizon and optimization horizon. A toy case including 4 zones and 100 shared scooters was used for verification, sensitivity and scalability analysis. In most cases, a larger number and capacity of dispatch trucks leads to higher profits. Using a truck with a capacity of 30 for relocation increases expected profits by 92.3%. The unit cost of relocation directly affects the relocation strategy and the profit. The higher the cost, the lower the profit. In the scenario which considers random OD matrix, the number of relocations decreases from 19 to 8 as unit costs increases, until the relocation stops when the unit cost reaches as much as the revenue per trip. The impact of look-ahead horizons on profitability and relocation strategies is not as straightforward as other parameters, it is influenced by many other factors such as OD matrix, truck capacity etc. Therefore, the optimal look-ahead horizon should be customized to the specific scenario and parameter settings. In this case, With a truck capacity of 20, increasing the look-ahead horizon from 2 to 3 raises the average profit by 3.5%, but increasing it to 4 does not further raise profits.

We have also developed two MILP models for the VRP. The first model is based on the classical pickup and delivery travel salesman problem (PDTSP). where each area is visited exactly once. The objective function is to minimize the total routing cost with an additional constraint of the maximum routing time. While the other model relaxes the traversal constraint to allow trucks to ignore unfavourable zones, accordingly we introduce unit relocation revenue to incentivize trucks to meet as many relocation demands as is profitable. The two models are verified in a toy case

containing two couples of zone with a total number of 11 scooters to be relocated. Moreover, the joint optimization for relocation and routing strategies was applied in the third toy case, which included 8 zones and 200 scooters. The results of the 5 rounds of optimization are presented and the optimized average profit is increased by 28.64% compared the one without optimization.

We analyzed the dataset of recorded e-scooter trip traces collected from Louisville. The highly active zones of e-scooter usage are mostly concentrated in zones indexed 200 to 300, but the use of shared scooters is rather unevenly distributed spatially. Besides, temporally, during night hours the number of trips remains very small, then it continues to rise during the daytime and peaks at 16:00. According to the analyzed characteristics, we have performed the aggregation of zones by 2*2, and the aggregation of hours by 3(daytime) and 5(night hours) to obtain 378 aggregated zones and 7 time-frames. By which, the aggregated OD matrices are further obtained. Taking them as parameters, the joint optimization is applied to the practical case, and the results of the 5 optimization rounds are presented, with the optimized average profit increasing by 6.6% compared to the one before optimization.

In the future, further improvements can be made in the following aspects: Firstly, an improvement can be made to the optimization method, by constructing a single model for the joint optimization rather than in two phases. Secondly, the models proposed are all solved by an exact algorithm, which shows limitations on the scalability performance, an attempt to develop heuristic or meta-heuristic algorithms could lead to improvements in computational performance. Finally, when constructing the routing optimization model, only one depot is set up in the study area and each dispatch truck must depart from the depot and return to the depot after completing relocation tasks. In future research, the design of a model for the case of multiple depots working together to provide collaborative services can be considered.

Appendix A

Sensitivity analysis in toy case 1

A.1 Scenario 1

Changing number of look-ahead horizon Intuitively, when $T = 1$, the relocation strategy should only consider the demand for the current time-frame, so the number of relocations should be less than 50, however, since in constraint 3.9, only half of the number of relocations is considered to be effectively used. Hence, there is tolerance for the maximum number of relocations to be greater than the current demand. In this case, it is reaching about 62. And when T is greater than 1, theoretically, the number of relocations should be higher, considering that the future demand is not less than 100. In practice the number of relocations is limited by the average number of scooters presenting in zone 2, which happens to be around 62 as well. In summary, due to the combination of this particular OD matrix, the total number of scooters and the parameter settings, the corner situation emerges where the look-ahead horizon has no effect on the relocation strategy as shown in the Figures A.1 to A.4.

A.2 Scenario 2

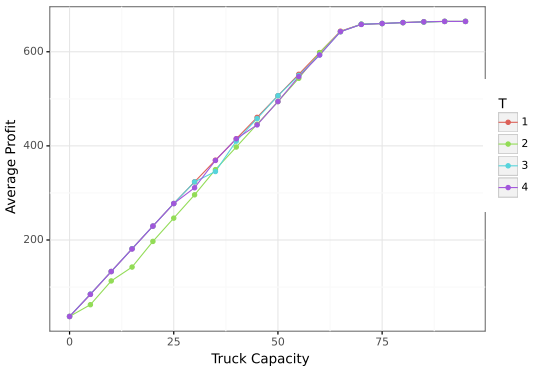

Figure A.1: Average profit v.s. T

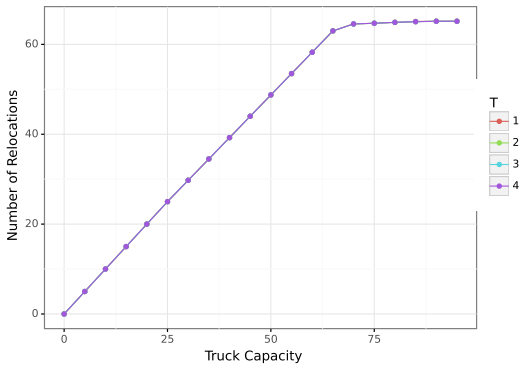

Figure A.2: Average number of trips v.s. T

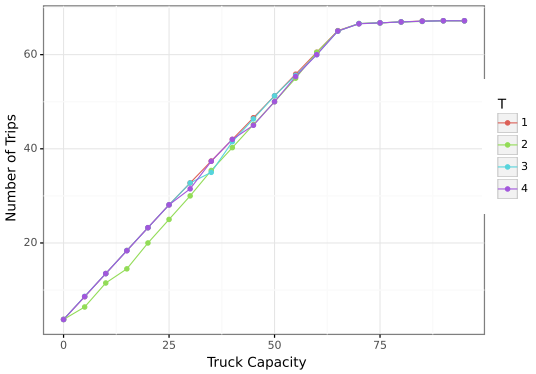

Figure A.3: Average relocation number v.s. T

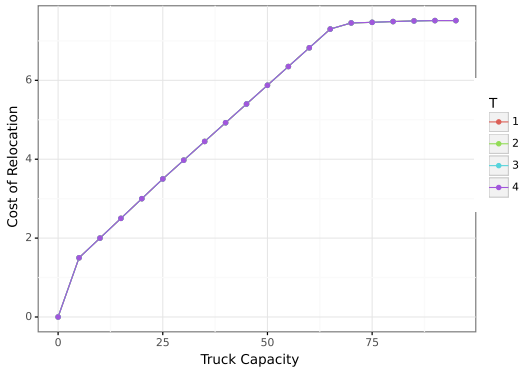

Figure A.4: Average relocation cost v.s. T

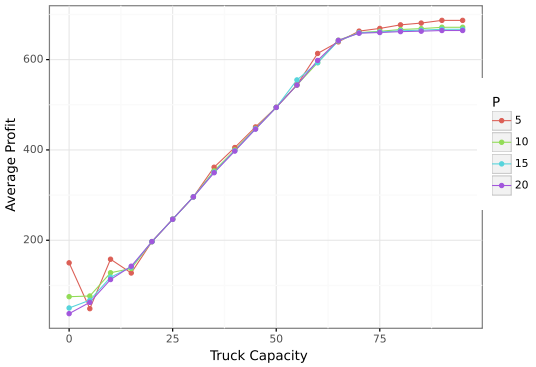

Figure A.5: Average profit v.s. P

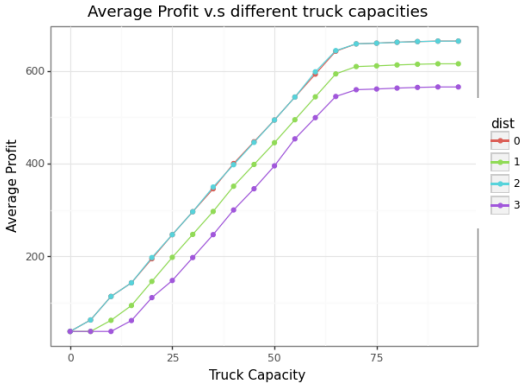

Figure A.6: Average profit v.s. distance matrix

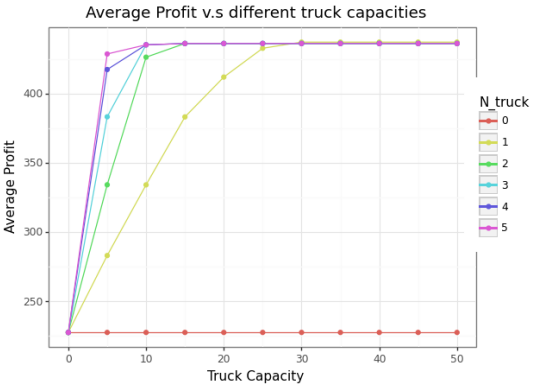

Figure A.7: Average profit v.s. Nt

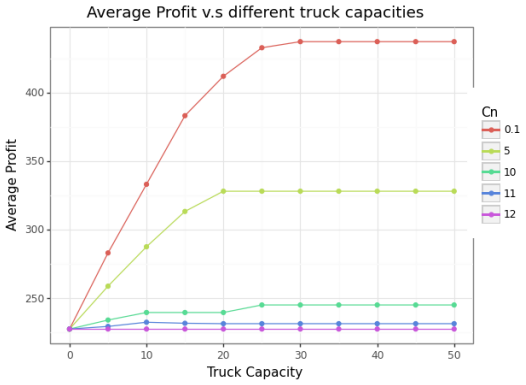

Figure A.8: Average profit v.s. Cn

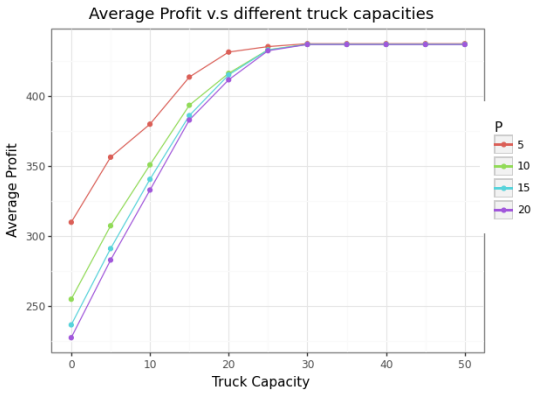

Figure A.9: Average profit v.s. P

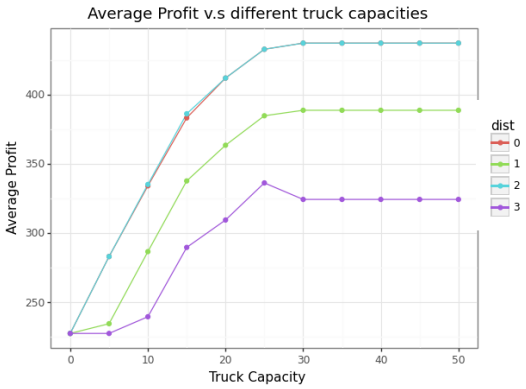

Figure A.10: Average profit v.s. distance matrix

Bibliography

[1] Fan Zhou, Zuduo Zheng, Jake Whitehead, Robert Perrons, Lionel Page, and Simon Washington. «Projected prevalence of car-sharing in four Asian-Pacific countries in 2030: What the experts think». In: *Transportation Research Part C: Emerging Technologies* 84 (2017), pp. 158–177. ISSN: 0968-090X. DOI: https://doi.org/10.1016/j.trc.2017.08.023. URL: https://www.sciencedirect.com/science/article/pii/S0968090X17302334 (cit. on p. 1).

[2] Cláudia A.Soares Machado, Nicolas Patrick Marie de Salles Hue, Fernando Tobal Berssaneti, and José Alberto Quintanilha. *An overview of shared mobility*. Nov. 2018. DOI: 10.3390/su10124342 (cit. on pp. 1, 2).

[3] Elliot Fishman. «Bikeshare: A Review of Recent Literature». In: *Transport Reviews* 36 (1 Jan. 2016). doi: 10.1080/01441647.2015.1033036, pp. 92–113. ISSN: 0144-1647. DOI: 10.1080/01441647.2015.1033036. URL: https://doi.org/10.1080/01441647.2015.1033036 (cit. on p. 1).

[4] Rusul L Abduljabbar, Sohani Liyanage, and Hussein Dia. «The role of micromobility in shaping sustainable cities: A systematic literature review». In: *Transportation Research Part D: Transport and Environment* 92 (2021), p. 102734. ISSN: 1361-9209. DOI: https://doi.org/10.1016/j.trd.2021.102734. URL: https://www.sciencedirect.com/science/article/pii/S1361920921000389 (cit. on p. 1).

[5] Susan A Shaheen, Stacey Guzman, and Hua Zhang. «Bikesharing in Europe, the Americas, and Asia: Past, Present, and Future». In: *Transportation Research Record* 2143 (1 Jan. 2010). doi: 10.3141/2143-20, pp. 159–167. ISSN: 0361-1981. DOI: 10.3141/2143-20. URL: https://doi.org/10.3141/2143-20 (cit. on p. 1).

[6] M D Keall, C Shaw, R Chapman, and P Howden-Chapman. «Reductions in carbon dioxide emissions from an intervention to promote cycling and walking: A case study from New Zealand». In: *Transportation Research Part D: Transport and Environment* 65 (2018). Export Date: 14 March 2023; Cited By: 34, pp. 687–696. DOI: 10.1016/j.trd.2018.10.004.

URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85055266310%5C&doi=10.1016%5C%2fj.trd.2018.10.004%5C&partnerID=40%5C&md5=1198a4e6b4e49f0ae91f65aa90d634a7 (cit. on p. 1).

[7] Helmut Brunner, Mario Hirz, Wolfgang Hirschberg, and Kurt Fallast. «Evaluation of various means of transport for urban areas». In: *Energy, Sustainability and Society* 8 (1 2018), p. 9. ISSN: 2192-0567. DOI: 10.1186/s13705-018-0149-0. URL: https://doi.org/10.1186/s13705-018-0149-0 (cit. on p. 1).

[8] Susan A Shaheen, Adam P Cohen, and Elliot W Martin. «Public Bikesharing in North America: Early Operator Understanding and Emerging Trends». In: *Transportation Research Record* 2387 (1 Jan. 2013). doi: 10.3141/2387-10, pp. 83–92. ISSN: 0361-1981. DOI: 10.3141/2387-10. URL: https://doi.org/10.3141/2387-10 (cit. on p. 2).

[9] Tianqi Gu, Inhi Kim, and Graham Currie. «To be or not to be dockless: Empirical analysis of dockless bikeshare development in China». In: *Transportation Research Part A: Policy and Practice* 119 (2019), pp. 122–147. ISSN: 0965-8564. DOI: https://doi.org/10.1016/j.tra.2018.11.007. URL: https://www.sciencedirect.com/science/article/pii/S0965856418309844 (cit. on p. 2).

[10] Giulia Oeschger, Páraic Carroll, and Brian Caulfield. «Micromobility and public transport integration: The current state of knowledge». In: *Transportation Research Part D: Transport and Environment* 89 (2020), p. 102628. ISSN: 1361-9209. DOI: https://doi.org/10.1016/j.trd.2020.102628. URL: https://www.sciencedirect.com/science/article/pii/S1361920920308130 (cit. on p. 2).

[11] Yuanyuan Guo and Sylvia Y He. «Perceived built environment and dockless bikeshare as a feeder mode of metro». In: *Transportation Research Part D: Transport and Environment* 92 (2021), p. 102693. ISSN: 1361-9209. DOI: https://doi.org/10.1016/j.trd.2020.102693. URL: https://www.sciencedirect.com/science/article/pii/S1361920920308774 (cit. on p. 2).

[12] Roberto Nocerino, Alberto Colorni, Federico Lia, and Alessandro Luè. «E-bikes and E-scooters for Smart Logistics: Environmental and Economic Sustainability in Pro-E-bike Italian Pilots». In: *Transportation Research Procedia* 14 (2016), pp. 2362–2371. ISSN: 2352-1465. DOI: https://doi.org/10.1016/j.trpro.2016.05.267. URL: https://www.sciencedirect.com/science/article/pii/S2352146516302733 (cit. on p. 2).

[13] Authors Shaheen. «Shared Micromoblity Policy Toolkit: Docked and Dockless Bike and Scooter Sharing». In: (2019). DOI: 10.7922/G2TH8JW7. URL: https://escholarship.org/uc/item/00k897b5 (cit. on p. 2).

[14] Tim Jones, Lucas Harms, and Eva Heinen. «Motives, perceptions and experiences of electric bicycle owners and implications for health, wellbeing and mobility». In: *Journal of Transport Geography* 53 (2016), pp. 41–49. ISSN: 0966-6923. DOI: https://doi.org/10.1016/j.jtrangeo.2016.04.006. URL: https://www.sciencedirect.com/science/article/pii/S0966692316301934 (cit. on p. 2).

[15] Graeme Lindsay, Alexandra Macmillan, and Alistair Woodward. «Moving urban trips from cars to bicycles: impact on health and emissions». In: *Australian and New Zealand Journal of Public Health* 35 (1 Feb. 2011). https://doi.org/10.1111/j.1753-6405.2010.00621.x, pp. 54–60. ISSN: 1326-0200. DOI: https://doi.org/10.1111/j.1753-6405.2010.00621.x. URL: https://doi.org/10.1111/j.1753-6405.2010.00621.x (cit. on p. 2).

[16] Taekwan Yoon, Christopher R Cherry, and Luke R Jones. «One-way and round-trip carsharing: A stated preference experiment in Beijing». In: *Transportation Research Part D: Transport and Environment* 53 (2017), pp. 102–114. ISSN: 1361-9209. DOI: https://doi.org/10.1016/j.trd.2017.04.009. URL: https://www.sciencedirect.com/science/article/pii/S1361920917303139 (cit. on p. 2).

[17] Susan A Shaheen, Nelson D Chan, and Helen Micheaux. «One-way carsharing's evolution and operator perspectives from the Americas». In: *Transportation* 42 (3 2015), pp. 519–536. ISSN: 1572-9435. DOI: 10.1007/s11116-015-9607-0. URL: https://doi.org/10.1007/s11116-015-9607-0 (cit. on p. 2).

[18] Sarah M Kaufman and Luke Buttenwieser. *The State of Scooter Sharing in United States Cities*. 2018. URL: https://www.limebike.com/press (cit. on p. 2).

[19] Charilaos Latinopoulos, Agathe Patrier, and Aruna Sivakumar. «Planning for e-scooter use in metropolitan cities: A case study for Paris». In: *Transportation Research Part D: Transport and Environment* 100 (2021), p. 103037. ISSN: 1361-9209. DOI: https://doi.org/10.1016/j.trd.2021.103037. URL: https://www.sciencedirect.com/science/article/pii/S1361920921003357 (cit. on p. 2).

[20] Grant McKenzie. «Spatiotemporal comparative analysis of scooter-share and bike-share usage patterns in Washington, D.C.» In: *Journal of Transport Geography* 78 (2019), pp. 19–28. ISSN: 0966-6923. DOI: https://doi.org/10.1016/j.jtrangeo.2019.05.007. URL: https://www.sciencedirect.com/science/article/pii/S0966692319302741 (cit. on p. 3).

[21] Rui Zhu, Xiaohu Zhang, Dániel Kondor, Paolo Santi, and Carlo Ratti. «Understanding spatio-temporal heterogeneity of bike-sharing and scooter-sharing mobility». In: *Computers, Environment and Urban Systems* 81 (2020), p. 101483. ISSN: 0198-9715. DOI: https://doi.org/10.1016/j.compenvurbsys.2020.101483. URL: https://www.sciencedirect.com/science/article/pii/S0198971519305812 (cit. on p. 3).

[22] Jiangtao Zhu, Ningke Xie, Zeen Cai, Wei Tang, and Xiqun (Michael) Chen. «A comprehensive review of shared mobility for sustainable transportation systems». In: *International Journal of Sustainable Transportation* (Mar. 2022). doi: 10.1080/15568318.2022.2054390, pp. 1–25. ISSN: 1556-8318. DOI: 10.1080/15568318.2022.2054390. URL: https://doi.org/10.1080/15568318.2022.2054390 (cit. on p. 3).

[23] *E-scooters in Europe: legal status, usage and safety Results of a survey in FERSI countries*. URL: www.fersi.org (cit. on p. 4).

[24] Joseph Hollingsworth, Brenna Copeland, and Jeremiah X. Johnson. «Are e-scooters polluters? the environmental impacts of shared dockless electric scooters». In: *Environmental Research Letters* 14 (8 Aug. 2019). ISSN: 17489326. DOI: 10.1088/1748-9326/ab2da8 (cit. on p. 4).

[25] Jianbin Huang, Qinglin Tan, He Li, Ao Li, and Longji Huang. «Monte carlo tree search for dynamic bike repositioning in bike-sharing systems». In: *Applied Intelligence* 52 (4 2022), pp. 4610–4625. ISSN: 1573-7497. DOI: 10.1007/s10489-021-02586-x. URL: https://doi.org/10.1007/s10489-021-02586-x (cit. on pp. 5, 7).

[26] Jie Zhang, Meng Meng, Yiik Diew Wong, Petros Ieromonachou, and David Z W Wang. «A data-driven dynamic repositioning model in bicycle-sharing systems». In: *International Journal of Production Economics* 231 (2021), p. 107909. ISSN: 0925-5273. DOI: https://doi.org/10.1016/j.ijpe.2020.107909. URL: https://www.sciencedirect.com/science/article/pii/S0925527320302693 (cit. on pp. 5, 7).

[27] Stefano Carrese, Fabio D'andreagiovanni, Tommaso Giacchetti, Antonella Nardin, Leonardo Zamberlan, and Leonardo Zam. *Night makes you beautiful: an optimization approach to overnight joint beautification and relocation in e-scooter sharing*. URL: https://hal.science/hal-03053339 (cit. on pp. 5, 8).

[28] Gianvito Losapio, Federico Minutoli, Viviana Mascardi, and Angelo Ferrando. *Smart Balancing of E-scooter Sharing Systems via Deep Reinforcement Learning*. 2021. URL: http://ceur-ws.org (cit. on pp. 5, 8).

[29] S Sasaki and T Namerikawa. «Optimal Incentive Design for One-Way EV Car Sharing System». In: 2022, pp. 851–856. ISBN: 2768-0770. DOI: 10.1109/CCTA49430.2022.9966009 (cit. on pp. 5, 8, 9).

[30] Patrick Stokkink and Nikolas Geroliminis. «Predictive user-based relocation through incentives in one-way car-sharing systems». In: *Transportation Research Part B: Methodological* 149 (2021), pp. 230–249. ISSN: 0191-2615. DOI: https://doi.org/10.1016/j.trb.2021.05.008. URL: https://www.sciencedirect.com/science/article/pii/S0191261521000874 (cit. on pp. 5, 9).

[31] Giulio Cerruto. «Modelling relocation strategies for shared mobility system management». PhD thesis. Politecnico di Torino, 2022 (cit. on pp. 5, 15).

[32] R Iacobucci, R Bruno, and C Boldrini. «A Multi-Stage Optimisation Approach to Design Relocation Strategies in One-Way Car-Sharing Systems With Stackable Cars». In: *IEEE Transactions on Intelligent Transportation Systems* 23 (10 2022), pp. 17048–17061. ISSN: 1558-0016. DOI: 10.1109/TITS.2022.3164989 (cit. on p. 8).

[33] Yu Jin, Cesar Ruiz, and Haitao Liao. «A simulation framework for optimizing bike rebalancing and maintenance in large-scale bike-sharing systems». In: *Simulation Modelling Practice and Theory* 115 (2022), p. 102422. ISSN: 1569-190X. DOI: https://doi.org/10.1016/j.simpat.2021.102422. URL: https://www.sciencedirect.com/science/article/pii/S1569190X21001209 (cit. on p. 8).

[34] Hyunsoo Yun, Eui-Jin Kim, Seung Woo Ham, and Dong-Kyu Kim. «Price incentive strategy for the E-scooter sharing service using deep reinforcement learning». In: *Journal of Intelligent Transportation Systems* (Oct. 2022). doi: 10.1080/15472450.2022.2135437, pp. 1–15. ISSN: 1547-2450. DOI: 10.1080/15472450.2022.2135437. URL: https://doi.org/10.1080/15472450.2022.2135437 (cit. on p. 8).

[35] Ning Wang, Qiaoqian Liu, Jiahui Guo, and Tong Fang. «A user-based adaptive joint relocation model combining electric car-sharing and bicycle-sharing». In: *Transportmetrica B: Transport Dynamics* 10 (1 Dec. 2022). doi: 10.1080/21680566.2021.2007174, pp. 1046–1069. ISSN: 2168-0566. DOI: 10.1080/21680566.2021.2007174. URL: https://doi.org/10.1080/21680566.2021.2007174 (cit. on p. 9).

[36] H Yang, W Qin, and Y Zhou. «User-based Battery Swapping Strategy in an Electric Bike-sharing System». In: 2022, pp. 236–240. DOI: 10.1109/IEEM55944.2022.9989569 (cit. on p. 9).

[37] Yuhan Zhang, Yichang Shao, Hui Bi, Li Aoyong, and Zhirui Ye. «Bike-sharing systems rebalancing considering redistribution proportions: A user-based repositioning approach». In: *Physica A: Statistical Mechanics and its Applications* 610 (2023), p. 128409. ISSN: 0378-4371. DOI: https://doi.org/10.1016/j.physa.2022.128409. URL: https://www.sciencedirect.com/science/article/pii/S0378437122009670 (cit. on p. 9).

[38] Giovanni Pantuso. «Exact solutions to a carsharing pricing and relocation problem under uncertainty». In: *Computers & Operations Research* 144 (2022), p. 105802. ISSN: 0305-0548. DOI: https://doi.org/10.1016/j.cor.2022.105802. URL: https://www.sciencedirect.com/science/article/pii/S0305054822000892 (cit. on p. 10).

[39] Yang Liu, Jiaohong Xie, and Nan Chen. «Stochastic one-way carsharing systems with dynamic relocation incentives through preference learning». In: *Transportation Research Part E: Logistics and Transportation Review* 166 (2022), p. 102884. ISSN: 1366-5545. DOI: https://doi.org/10.1016/j.tre.2022.102884. URL: https://www.sciencedirect.com/science/article/pii/S1366554522002617 (cit. on p. 10).

[40] Guoxun Xu, Ting Xiang, Yanfeng Li, Jun Li, and Qiang Guo. «A mixed rebalancing strategy in bike sharing systems». In: *Engineering Optimization* 54 (7 July 2022). doi: 10.1080/0305215X.2021.1916008, pp. 1160–1177. ISSN: 0305-215X. DOI: 10.1080/0305215X.2021.1916008. URL: https://doi.org/10.1080/0305215X.2021.1916008 (cit. on p. 10).

[41] G Guo and Y Hou. «Rebalancing of One-Way Car-Sharing Systems Considering Elastic Demand and Waiting Time». In: *IEEE Transactions on Intelligent Transportation Systems* 23 (12 2022), pp. 23295–23310. ISSN: 1558-0016. DOI: 10.1109/TITS.2022.3208215 (cit. on p. 10).

[42] Grigorios D Konstantakopoulos, Sotiris P Gayialis, and Evripidis P Kechagias. «Vehicle routing problem and related algorithms for logistics distribution: a literature review and classification». In: *Operational Research* 22 (3 2022), pp. 2033–2062. ISSN: 1866-1505. DOI: 10.1007/s12351-020-00600-7. URL: https://doi.org/10.1007/s12351-020-00600-7 (cit. on p. 11).

[43] Asma M Altabeeb, Abdulqader M Mohsen, and Abdullatif Ghallab. «An improved hybrid firefly algorithm for capacitated vehicle routing problem». In: *Applied Soft Computing* 84 (2019), p. 105728. ISSN: 1568-4946. DOI: https://doi.org/10.1016/j.asoc.2019.105728. URL: https://www.sciencedirect.com/science/article/pii/S1568494619305095 (cit. on p. 11).

[44] Asma M Altabeeb, Abdulqader M Mohsen, Laith Abualigah, and Abdullatif Ghallab. «Solving capacitated vehicle routing problem using cooperative firefly algorithm». In: *Applied Soft Computing* 108 (2021), p. 107403. ISSN: 1568-4946. DOI: https://doi.org/10.1016/j.asoc.2021.107403. URL: https://www.sciencedirect.com/science/article/pii/S1568494621003264 (cit. on p. 11).

[45] İlhan İLHAN. «An improved simulated annealing algorithm with crossover operator for capacitated vehicle routing problem». In: *Swarm and Evolutionary Computation* 64 (2021), p. 100911. ISSN: 2210-6502. DOI: https://doi.org/10.1016/j.swevo.2021.100911. URL: https://www.sciencedirect.com/science/article/pii/S2210650221000729 (cit. on p. 11).

[46] Mohammad Sajid, Jagendra Singh, Raza Abbas Haidri, Mukesh Prasad, Vijayakumar Varadarajan, Ketan Kotecha, and Deepak Garg. «Article a novel algorithm for capacitated vehicle routing problem for smart cities». In: *Symmetry* 13 (10 Oct. 2021). ISSN: 20738994. DOI: 10.3390/sym13101923 (cit. on p. 12).

[47] Vinícius R Máximo and Mariá C V Nascimento. «A hybrid adaptive iterated local search with diversification control to the capacitated vehicle routing problem». In: *European Journal of Operational Research* 294 (3 2021), pp. 1108–1119. ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2021.02.024. URL: https://www.sciencedirect.com/science/article/pii/S037722172100117X (cit. on p. 12).

[48] Daqing Wu and Chenxiang Wu. «Research on the Time-Dependent Split Delivery Green Vehicle Routing Problem for Fresh Agricultural Products with Multiple Time Windows». In: *Agriculture (Switzerland)* 12 (6 June 2022). ISSN: 20770472. DOI: 10.3390/agriculture12060793 (cit. on p. 12).

[49] Binbin Pan, Zhenzhen Zhang, and Andrew Lim. «Multi-trip time-dependent vehicle routing problem with time windows». In: *European Journal of Operational Research* 291 (1 2021), pp. 218–231. ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2020.09.022. URL: https://www.sciencedirect.com/science/article/pii/S0377221720308262 (cit. on p. 12).

[50] R J Kuo, Shih-Hao Lu, Pei-Yu Lai, and Setyo Tri Windras Mara. «Vehicle routing problem with drones considering time windows». In: *Expert Systems with Applications* 191 (2022), p. 116264. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2021.116264. URL: https://www.sciencedirect.com/science/article/pii/S0957417421015736 (cit. on p. 12).

[51] Huizhen Zhang, Qinwan Zhang, Liang Ma, Ziying Zhang, and Yun Liu. «A hybrid ant colony optimization algorithm for a multi-objective vehicle routing problem with flexible time windows». In: *Information Sciences* 490 (2019), pp. 166–190. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2019.03.070. URL: https://www.sciencedirect.com/science/article/pii/S002002551632120X (cit. on p. 12).

[52] M Salehi Sarbijan and J Behnamian. «Real-time collaborative feeder vehicle routing problem with flexible time windows». In: *Swarm and Evolutionary Computation* 75 (2022), p. 101201. ISSN: 2210-6502. DOI: https://doi.org/10.1016/j.swevo.2022.101201. URL: https://www.sciencedirect.com/science/article/pii/S2210650222001675 (cit. on p. 13).

[53] Xinrui Sun, Kunpeng Li, and Wenli Li. «The vehicle routing problem with release dates and flexible time windows». In: *Engineering Optimization* 54 (12 Dec. 2022). doi: 10.1080/0305215X.2021.1974853, pp. 2123–2139. ISSN: 0305-215X. DOI: 10.1080/0305215X.2021.1974853. URL: https://doi.org/10.1080/0305215X.2021.1974853 (cit. on p. 13).

[54] Gerardo Berbeglia, Jean-François Cordeau, Irina Gribkovskaia, and Gilbert Laporte. «Static pickup and delivery problems: a classification scheme and survey». In: *TOP* 15 (1 2007), pp. 1–31. ISSN: 1863-8279. DOI: 10.1007/s11750-007-0009-0. URL: https://doi.org/10.1007/s11750-007-0009-0 (cit. on p. 13).

[55] Gábor Nagy and SaıHERE!HERE!d Salhi. «Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries». In: *European Journal of Operational Research* 162 (1 2005), pp. 126–141. ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2002.11.003. URL: https://www.sciencedirect.com/science/article/pii/S0377221703008361 (cit. on p. 13).

[56] Suna Cetin and Cevriye Gencer. «A Heuristic Algorithm for Vehicle Routing Problems with Simultaneous Pick-Up and Delivery and Hard Time Windows». In: *Open Journal of Social Sciences* 03 (03 2015), pp. 35–41. ISSN: 2327-5952. DOI: 10.4236/jss.2015.33008 (cit. on p. 13).

[57] H Zhang, Z Wang, M Tang, X Lv, H Luo, and Y Liu. «Dynamic Memory Memetic Algorithm for VRPPD With Multiple Arrival Time and Traffic Congestion Constraints». In: *IEEE Access* 8 (2020), pp. 167537–167554. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.3023090 (cit. on p. 14).

[58] Qingfeng Chen, Kunpeng Li, and Zhixue Liu. «Model and algorithm for an unpaired pickup and delivery vehicle routing problem with split loads». In: *Transportation Research Part E: Logistics and Transportation Review* 69 (2014), pp. 218–235. ISSN: 1366-5545. DOI: https://doi.org/10.1016/

j.tre.2014.06.010. URL: https://www.sciencedirect.com/science/article/pii/S1366554514001057 (cit. on p. 14).

[59] Gabriel Gutiérrez-Jarpa, Guy Desaulniers, Gilbert Laporte, and Vladimir Marianov. «A branch-and-price algorithm for the Vehicle Routing Problem with Deliveries, Selective Pickups and Time Windows». In: *European Journal of Operational Research* 206 (2 2010), pp. 341–349. ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2010.02.037. URL: https://www.sciencedirect.com/science/article/pii/S0377221710001700 (cit. on p. 14).

[60] Michele Cocca, Douglas Teixeira, Luca Vassio, Marco Mellia, Jussara M. Almeida, and Ana Paula Couto da Silva. «On Car-Sharing Usage Prediction with Open Socio-Demographic Data». In: *Electronics* 9.1 (2020). ISSN: 2079-9292. DOI: 10.3390/electronics9010072. URL: https://www.mdpi.com/2079-9292/9/1/72 (cit. on p. 14).

[61] Michele Cocca, Danilo Giordano, Marco Mellia, and Luca Vassio. «Free Floating Electric Car Sharing Design: Data Driven Optimisation». In: *Pervasive Mob. Comput.* 55.C (Apr. 2019), pp. 59–75. ISSN: 1574-1192. DOI: 10.1016/j.pmcj.2019.02.007. URL: https://doi.org/10.1016/j.pmcj.2019.02.007 (cit. on p. 14).

[62] Michele Cocca, Danilo Giordano, Marco Mellia, and Luca Vassio. «Data Driven Optimization of Charging Station Placement for EV Free Floating Car Sharing». In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. 2018, pp. 2490–2495. DOI: 10.1109/ITSC.2018.8569256 (cit. on p. 14).

[63] Alessandro Ciociola, Dena Markudova, Luca Vassio, Danilo Giordano, Marco Mellia, and Michela Meo. «Impact of charging infrastructure and policies on electric car sharing systems». In: *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE. 2020, pp. 1–6 (cit. on p. 14).

[64] Michelangelo Barulli, Alessandro Ciociola, Michele Cocca, Luca Vassio, Danilo Giordano, and Marco Mellia. «On Scalability of Electric Car Sharing in Smart Cities». In: *2020 IEEE International Smart Cities Conference (ISC2)*. IEEE. 2020, pp. 1–8 (cit. on p. 15).

[65] Edoardo Fassio, Alessandro Ciociola, Danilo Giordano, Michel Noussan, Luca Vassio, and Marco Mellia. «Environmental and Economic Comparison of ICEV and EV in Car Sharing». In: *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. 2021, pp. 1621–1626. DOI: 10.1109/ITSC48978.2021.9564578 (cit. on p. 15).

[66] Alessandro Ciociola, Michele Cocca, Danilo Giordano, Luca Vassio, and Marco Mellia. «E-Scooter Sharing: Leveraging Open Data for System Design». In: *2020 IEEE/ACM 24th International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*. 2020, pp. 1–8. DOI: 10.1109/DS-RT50469.2020.9213514 (cit. on p. 15).

[67] Leonardo Tolomei, Stefano Fiorini, Alessandro Ciociola, Luca Vassio, Danilo Giordano, and Marco Mellia. «Benefits of Relocation on E-scooter Sharing-a Data-Informed Approach». In: *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE. 2021, pp. 3170–3175 (cit. on p. 15).

[68] Sophie N. Parragh, Karl F. Doerner, and Richard F. Hartl. «A survey on pickup and delivery problems: Part II: Transportation between pickup and delivery locations». In: *Journal fur Betriebswirtschaft* 58 (2 June 2008), pp. 81–117. ISSN: 03449327. DOI: 10.1007/s11301-008-0036-4 (cit. on p. 23).

[69] Laurent Perron and Vincent Furnon. *OR-Tools*. Version v9.6. Google, Mar. 13, 2023. URL: https://developers.google.com/optimization/ (cit. on p. 25).