Optimizing the Battery Swap Strategy for e-Cargo Bicycles

Erasmus
University
Rotterdam

Optimizing the Battery Swap Strategy for e-Cargo Bicycles

Master Thesis Project

by

V.M.H. (Veerle) Willemse

to obtain the degree of Master of Science
at the Erasmus University Rotterdam.

Student number:	499185	
Project duration:	June 7, 2022 –	October 31, 2022
Thesis committee:	Dr. P. C. Bouman,	Erasmus University Rotterdam, supervisor
	Dr. R. Spliet,	Erasmus University Rotterdam, second assessor
	Ir. T. A. Nguyen,	Districon Solutions B.V., supervisor

An electronic version of this thesis is available at https://thesis.eur.nl.

Acknowledgements

This report would not have been possible without the constant support and guidance of numerous people.

First, I would like to thank my daily supervisor Ir. Thy Anh Nguyen from Districon, for the guidance and feedback. The weekly meetings contributed greatly to the success of my graduation process. The ability to brainstorm together and exchange ideas were invaluable. Although this report is my own work, our meetings helped to broaden my view on the project. Consecutively, I wish to express my sincere thanks to the rest of the team I was part of at Districon. Thank you for the friendly conversations and for making me feel at home from the first moment I stepped into the office.

Next, I would like to show my gratitude towards my supervisor Dr. P.C. Bouman from the Erasmus University, who guided me through this process with his knowledge. The provided scientific view on the problem helped me to focus on the important aspects of writing a thesis specifically. My further thanks goes out to the second assessor Dr. R. Spliet, for taking the time to read my thesis and provide feedback.

Further, I would like to thank Niels Veenman from Cargoroo for creating the opportunity to perform this research. The clear agreements, fast communication, and pleasant conversations have contributed to a great project. The ability to experience the swap procedure in real life, having a look at the depot and home office really contributed to an enjoyable project.

Lastly, I would like to thank my family and friends for their support during my study period at the Erasmus University. All of this would not have been achievable without them.

V.M.H. (Veerle) Willemse
Rotterdam, October 2022

Abstract

As awareness of society on sustainability increases gradually, new transportation modes have been introduced. Cargoroo contributes to this movement as an e-cargo bicycle rental company. In their system, customers rent bicycles at fixed locations under the condition of sufficient remaining battery. For that reason, the company aims to maximize the battery levels taking operational costs into account. They have so-called battery swappers, who replace low power batteries at the designated locations.

This thesis presents two tour construction methods to schedule battery swappers. We formulate this problem by extending the bi-objective Travelling Salesman Problem with Profits with capacity constraints. In our first solution approach, we linearize the objective resulting in the mono-objective Profitable Tour Problem. This exact mathematical model constructs a tour that maximizes the collected prizes, depending on the battery levels of the visited bicycles, and minimizes the travelled distance. Since this problem is $\mathcal{NP}$-hard, a Local Search is proposed as a second, alternative solution method. This fast greedy heuristic starts with an initial tour to which improvement steps are applied.

The performances of the tour construction methods are evaluated by means of a Discrete Event Simulation. We apply this simulation to real world instances and define a time horizon that is subdivided into time periods of equal duration. Each time period includes two events, the tour construction by our proposed methods, and the simulation of new rental periods based on one year of historical data. In our experiments, we fit distributions to the data about the travelled distances and rental durations, extended with data about the weather conditions. This approach enables us to evaluate the long term effect of the constructed tours. In this problem, the main performance indicators are the average battery levels and travel times of the swappers.

The result of the exact mathematical formulation to construct tours shows high computational times in contrast to the Local Search method. In addition, it does not outperform the heuristic approach based on the performance indicators as the tours constructed by the Local Search show good long term performances.

Ultimately, we observe a trade-off between the average battery levels of the bicycles and the travel time spent to swap the batteries.

Contents

List of Algorithms

List of Figures

List of Tables

Nomenclature

1-$PDTSP$	One-Commodity Pickup-and-Delivery Travelling Salesman Problem
CDF	Cumulative Distribution Function
$CTSP$	Capacitated Travelling Salesman Problem
$CTSP$-PD	Capacitated Traveling Salesman Problem with Pickups and Deliveries
$CVRP$	Capacitated Vehicle Routing Problem
DES	Discrete Event Simulation
$GRASP$	Greedy Randomized Adaptive Search Procedure
LS	Local Search
OP	Orienteering Problem
$PCTSP$	Prize Collecting Travelling Salesman Problem
PLS	Pareto Local Search
PTP	Profitable Tour Problem
$STSP$	Selective Travelling Salesman Problem
TS	Tabu Search
TSP	Travelling Salesman Problem
$TSPP$	Travelling Salesman Problem with Profits
VND	Variable Neighbourhood Descent
VNS	Variable Neighbourhood Search
VRP	Vehicle Routing Problem

1
Introduction

Over the last years, one of the largest challenges in the world is the reduction of CO_2 emissions, which is related to the effect of global warming [42]. This rapid development created a new worldwide priority and many researchers emphasize the importance to combat climate change. The impact can be highlighted by the rising sea level and extreme weather events causing serious concerns. To compromise the needs of future generations, environmental changes have to be made [33]. In face of those increasingly serious environmental problems, green and sustainable transportation modes have been explored and developed.

Vehicle sharing has become increasingly popular as an alternative to personally owned vehicles, due to the necessity for shifting travel modes. It creates the opportunity for more efficient transportation with potential benefits. The reduction in parking space, improved mobility and accessibility for disadvantaged populations are presented. After this emergence, shared electric vehicles are introduced as a promising solution to growing issues of air pollution.

In urban environments, one of the evolved systems is electric bicycle sharing, which can be seen as beneficial in comparison to electric car sharing. It contributes to the ability to avoid traffic congestion and to reach your destination more efficiently. This environmental conscious system can be extended with the ability to carry goods or people by using e-cargo bicycles.

Despite that shared electric vehicles have led to several benefits regarding the environment and efficiency, these systems have a drawback. Electric vehicles are proven to come with range anxiety as people fear running out of battery and strongly avoid these situations [39].

As customers typically prefer to rent high power battery vehicles, bicycles with low power batteries are less likely to be rented and customers may even give up rentals. To prevent missing out on customers, efficient battery swap strategies have been proposed [27, 32, 49].

In this study, e-cargo bicycles are designated to fixed locations and batteries have to be swapped between two rental periods. To perform this operation, a homogeneous capacitated vehicle fleet is used. Each pickup goes along with the delivery of a fully charged battery, which is referred to as *swapping*. If all low power batteries are swapped or the vehicle capacity is reached, the *swapper* returns to the depot to insert the empty batteries into the charging dock. Each route that starts and ends at the depot is called a *tour*. In essence, an efficient strategy for swapping should be established to maintain high battery levels and minimize working hours for swappers. As there is a trade-off, the problem could be formulated as a Travelling Salesman Problem with Profits (TSPP). Due to the computational complexity of this bi-objective problem, we reformulate the TSPP to the mono-objective Profitable Tour Problem (PTP). Prizes can be collected by swapping batteries dependent on their current battery level at the cost of travelling to the designated location of the bicycle.

In comparison to standard Travelling Salesman Problems (TSPs), it is not mandatory to visit all customers. It is allowed to swap part of the batteries to keep the travelled distance to a minimum. In this paper, an exact mathematical model based on the PTP, and a Local Search (LS) heuristic are presented to construct tours. To evaluate the long term performance of our methods, a Discrete Event Simulation (DES) is employed. At the start of each time period, the proposed methods either decide to make a tour satisfying all constraints, or to not make a tour. In addition, we simulate new rental periods based on historical data on bookings.

The approaches are tested on three different instances provided by rental company Cargoroo, corresponding to different cities. Both the exact model and heuristic method succeed in creating tours, and the simulation approach enables us to evaluate the performance by observing the long term effect on battery levels and availability. Based on those key performance indicators (KPIs), the solutions of our methods are similar. However, we observe that the exact model is sensitive for different instance sizes, while the LS provides solutions within seconds.

The remainder of this report is structured as follows. In Section 2 we describe the problem in detail, and in Section 3 we discuss the relevant literature on this problem. Next, in Section 4, we present our formulation, our assumptions, and describe our methodology. Section 5 is devoted to briefly discussing the data. Section 6 presents the outcomes and results of this research. Lastly, Section 7 contains our concluding remarks.

2

Problem Description

In this chapter, we elaborate on the problem and explain the specifics. The research focuses on an optimal strategy for swapping batteries in the shared e-cargo bicycle system of Cargoroo, a rental company. The problem description is of great importance in developing a good strategy. First, we describe the general problem followed by a detailed description of the components and corresponding assumptions. Lastly, the problem is explained from a mathematical point of view.

2.1. General Problem

The main goal of this paper is to establish a good strategy to retain high battery levels in a fleet of shared e-cargo bicycles. We assume that the rental company has full responsibility, and people are employed to take care of picking up empty, and delivering fully charged batteries.

As people tend to prefer renting high power batteries, we aim to maximize the battery levels by swapping timely. If customers cannot rent a bicycle due to a low battery, this will negatively impact user satisfaction. In the long term, the rental company loses customers causing profits to decrease. A drawback of swapping batteries often is that operational costs increase as more employees are required to perform the operation. In this research, we aim to optimize and focus on this trade-off.

Next, we discuss three components included in the problem: the bicycles, the employees, and the customers. We also discuss the assumptions for those three categories.

Bicycles

The shared e-cargo bicycles are assigned to fixed locations, where customers start and end their rental. It is mandatory to return the bicycle to the start location. Consequently, all bicycles are either located at the designated spot or rented.

Based on this assumption, we introduce two different definitions of availability. A bicycle is *swap available* if it is not rented and present at the designated location. Further, to prevent customers from being disappointed due to a small driving range, a minimum battery level is required for a bicycle to be *rental available*. It can be seen as a starting requirement, as customers are allowed to cross the threshold during a rental period.

Lastly, we assume that a low power battery is always replaced by a fully charged battery.

Swappers

The employees swapping batteries are called swappers. They are equipped with a homogeneous vehicle fleet with a fixed capacity. The swappers start at the depot and use a vehicle filled with fully charged batteries. Therefore, the fleet is capacitated to the number of batteries that can be transported. Once the batteries are replaced, the swapper returns to the depot. The collected batteries are put into a charging dock and a new batch of fully charged batteries is taken. We assume that the number of fully charged batteries at the depot is not limited.

Customers

Each bicycle has a designated location, at which customers are obligated to start and end their rental. During the rental period, there are no restrictions on the customer's trip except for the battery level. We assume that time and weather seasonality affect customer demand.

2.2. Mathematical View

In recent years, several researchers investigated the effect of a limited battery capacity on shared electric vehicles, addressed sustainability benefits, and introduced different charging strategies [5, 23].

In our research, employees perform the swap operation and customers do not participate in the charging system. Bicycles have their designated location at which the batteries must be swapped. Based on these properties, we could describe our problem as a TSP in which we have to visit bicycles. In Section 3, we will elaborate further on the TSP and the exact formulations.

3

Literature Review

In this chapter, we discuss relevant literature. First, several papers regarding the concept of shared electric vehicles with limited battery capacities are discussed. Next, a review of TSPs follows, and proposed heuristic approaches are presented. As the performance of these approaches needs to be tested over time, a DES approach is introduced and discussed in the last part of this literature review.

3.1. Limited Battery Capacity

Over the last years, the industry of shared electric vehicles has developed rapidly. An increasing demand for vehicle electrification, sharing, and even automation are considered the three main transportation revolutions over the past half-century [44].

Vehicle automation systems are currently being explored, developed, and tested [10, 24], while electric and shared vehicles are already operating. The first shared electric vehicle was introduced in 1974 [25], although the rapid development occurred mainly in recent years. Initially, the aim was to ensure smooth operation and tackle technical limitations. Meanwhile, the concept of shared electric vehicles was explored and developed further, and the interest and demand for sustainable transportation modes increased. Benefits in the area of less pollution and costs of personally owned vehicles were emphasized and the need for shared electric vehicle systems increased.

The first shared vehicle system started in 1948 in the city of Zurich, motivated by economic reasons. Unfortunately, this attempt was not successful and neither were other public car-sharing systems in the following years [17]. However, the growing attention to reducing CO_2 emissions contributed positively to the development of shared vehicle systems. Koerner and Klopatek [31] evaluated the CO_2 emission of Phoenix (Arizona) and concluded that more than 80% of the CO_2 input into the urban environment was produced by humans and automobile activity. The International Energy Agency (IEA) study [4] found that urban areas account for 67% of energy consumption and 71% of CO_2 emissions worldwide, corresponding to the higher population density. Related to these studies, Chapman [8] states that behavioural changes stimulated by policy are also required to achieve a stabilisation of greenhouse gas emissions from transport. As a consequence, pressure on policymakers to facilitate sustainable transport modes grows, and alternative transportation modes in urban environments are deployed, including shared electric scooters, steps, and bicycles. These means of transport come with more benefits, such as the reduction of parking space needed, leading to more efficient use of space in crowded places. Another often emphasised benefit is the decrease in traffic congestion, due to smaller and more flexible vehicles. In addition to the reduction of CO_2 emissions, the introduction of shared electric vehicles adds value to the development of an improved urban environment.

Despite all the environmental benefits, there is a lot of focus on the drawbacks of shared electric vehicles. One of the main shortcomings is the limited battery capacity, as the usage is restricted to the remaining power level of the vehicle. Customers are proven to be sensitive to this limitation, which is referred to as range anxiety [39]. This fear of running out of battery affects the behaviour of customers, resulting in reduced efficiency of the system [35]. To maximize the utilization, the importance of timely charging the batteries is often featured.

Specifically, in the field of our problem, focused on swapping the batteries of shared e-cargo bicycles timely, little research has been done. This is mainly due to the vehicle type in combination with the charging system in which we swap batteries. For example, Osorio, Lei, and Ouyang [36] explore the effect of timely charging batteries of shared electric scooters, and Brandstätter, Kahr, and Leitner [6] focus on shared electric cars. Unlike our problem, most of those papers propose a solution method combining the reallocation and charging strategy into one model. Since our bicycles are designated to a fixed location, we do not consider the reallocation of one-way trip vehicles. Equivalent to our problem, Zheng et al. [48] propose a strategy for replacing the batteries timely without vehicle reallocation. They use a Vehicle Routing Problem (VRP) to determine optimal routes, visiting the locations with low power batteries. These routes are completed by several vehicles and include multiple depots, to which low power batteries are brought, and fully charged are picked up. In our situation, we limit the problem to one depot and one vehicle. Therefore, we can formulate our problem as a variant of the well-known TSP. Next, relevant research regarding this formulation will be discussed.

3.2. Travelling Salesman Problems

As introduced our problem can be formulated as a TSP in which a vehicle has to supply goods to customers. The general TSP was first described as a mathematical problem by the RAND Corporation in 1949 [41], who desired to find the shortest closed curve containing n different points in the plane. The TSP comprises a large class of problems, which belong to $\mathcal{NP}$-hard problems. Karp [28] showed that the Hamiltonian cycle problem was $\mathcal{NP}$-complete, which implies the $\mathcal{NP}$-hardness of TSP. Since the TSP is extensively studied, many applications led to numerous variations and solution approaches.

If the demand of the visited customers is limited by the salesman's capacity, the problem changes into the Capacitated Travelling Salesman Problem (CTSP), which is often referred to as the Capacitated Vehicle Routing Problem (CVRP) [40]. In this problem, the demands collected on a tour cannot exceed the salesman's vehicle capacity. Due to this capacity constraint, multiple vehicles may be required to serve all customers and ensure feasibility. Therefore, the problem becomes a VRP. In contrast to our problem, in VRPs multiple vehicles are available to serve customers.

A closely related problem is the Capacitated Travelling Salesman Problem with Pickup and Deliveries (CTSP-PD) [34]. In this problem, the salesman needs to pick up goods and/or deliver goods to a given set of locations. The demands of the customers are known and a tour must be created without violating the vehicle's capacity. An important part of this problem is the order in which the pick-ups and deliveries are performed. In addition, the initial load of the vehicle leaving the depot can be any quantity and must be determined by the model. The general CTSP-PD can be used for the distribution of multiple commodities. However, our problem considers the distribution of one commodity corresponding to the batteries. Therefore, the One-Commodity Pickup-and-Delivery Travelling Salesman Problem (1-PDTSP) [22] is more similar to our problem, in which all products are identical and have equal weights. One of the assumptions of the 1-PDTSP is that the products are distributed among customers. So, any amount collected from a pickup can be delivered to any other customer. This assumption defers strongly from our study, as we assume that fully charged batteries are taken from the depot and delivered to the customers.

A general assumption on the discussed TSPs is that it is mandatory to visit all customers. However, due to the capacity constraint, this may not be possible. Alternatively, the Selective Travelling Salesman Problem (STSP) consists of selecting customers to maximize profits, such that the tour length does not exceed a prespecified bound. In this problem, it is not mandatory to visit all customers and the main goal is to maximize profits regardless of any costs. In practice, travelling more comes with higher costs, which negatively impacts profits. Therefore, other solution methods are introduced to minimize travel costs and maximize the total collected prizes.

The TSPP [16] belongs to the class of the STSP and is applied to different subjects. One of the differences with the standard TSPs is that visiting a customer is not associated with a constant value. Further, the TSPP is bi-criterial with two opposite objectives. The first pushes the salesman to collect as much profit as possible, and the second tries to limit the distance to minimize costs. Solving this problem with two objectives results in finding a non-inferior solution set. This is a set of feasible solutions in which neither of the objectives can be improved without deteriorating the other.

An attempt to address the TSPP was made by Keller [29], and later by Keller and Goodchild [30]. The explicit multi-objective form was used and referred to as the multi-objective vending problem. They concluded that heuristics pose the most feasible solution approach, because of computer capacity limitations of exact solution techniques. As this multi-criterion problem is hard to solve, most researchers transform it into one objective.

Although there is an efficiency loss in formulating the problem to a mono-criterion objective, several attempts have been made. According to Ehrgott and Gandibleux [14], a linear scalarization of the objectives using a weighted sum is the most popular solution method. Unfortunately, due to the linearization, an optimal solution is no longer guaranteed. Despite the possibility to vary weights, it is not suited for non-convex solution spaces. As a consequence of the linear combination of different objectives, points on the convex envelope of the objective space can be found. We refer to this problem as the PTP introduced by Dell'Amico, Maffioli, and Värbrand [12], who maximize the difference between the collected prizes and the travel costs.

An alternative solution approach for bi-objective combinatorial optimization problems is the ϵ-constraint method, proposed by Bérubé, Gendreau, and Potvin [2]. In this study, one of the objectives is used as the optimization criterion. The other objective is added to the model as a constraint by forcing it to be bounded by ϵ. For every different value of ϵ, a new optimization problem is formulated, resulting in a high number of subproblems. It is possible to generate the pareto front [47], but in practice, it is difficult to establish an efficient variation scheme for this upper bound ϵ. Therefore, this approach is often used and integrated into heuristic approaches.

One of the solution methods to solve the TSPP using the ϵ-constraint method is the STSP, also referred to as the Orienteering Problem (OP). In this formulation, the travel costs are limited by an upper bound and added as a constraint. The remaining part of the objective is to maximize the total collected profits. Another solution method based on this ϵ-constraint method is to minimize the travelled distance and force the collected prize to be at least ϵ. This is referred to as the Prize Collecting Travelling Salesman Problem (PCTSP). In the formulation of an ϵ-constraint method, it is of great importance to define ϵ such that the number of subproblems is limited. One could either evaluate the model for all values of ϵ or reduce the computation time and accept an optimality gap. In the standard PCTSP, it is assumed that the value of ϵ is known. In our study it is difficult to set this bound as the prizes depend on the current battery status, differing over time.

To solve the bi-objective problem to optimality the pareto global optimum set must be found. Ehrgott [13] states that this is an $\mathcal{NP}$-hard problem and obtaining an exact solution becomes extremely time-consuming with increasing instance sizes. For that reason, the focus on finding the pareto global optimum set shifts to finding a good approximation to this set. Several papers have been written on the approximation. Paquete, Chiarandini, and Stützle [37] have developed a Pareto Local Search (PLS) algorithm based on different neighbourhoods. The results of this study suggest that, once a reasonably good solution is found by this method, improvements can be made by a small number of edge exchanges.

The PCTSP belongs to the $\mathcal{NP}$-hard class as it is a generalization of the standard TSP [1]. Bienstock et al. [3] present an approximation algorithm with constant bounds to solve the PCTSP. In this study, a method is included to round fractional solutions of an LP relaxation to integers, which is feasible for the original problem. Dell'Amico, Maffioli, and Sciomachen [11] also propose a heuristic based on a relaxation. They formulate a Lagrangian heuristic using the subgradient technique to solve the dual problem. First, a feasible solution is created by inserting customers into a tour to collect the minimum required prize. After that, they apply an extension-and-collapse improvement algorithm and obtain the upper bound. Their results show that a Lagrangian heuristic is in favor of an LP relaxation in case of an asymmetric TSP.

Gendreau, Laporte, and Semet [19] analyze the polyhedral structure of the formulation and introduce several classes of valid inequalities that proved to be facet defining. The results of this branch-and-cut procedure based on the LP relaxation show high-quality solutions within a short computation time. Fischetti, González, and Toth [18] also based their method on families of valid inequalities and additionally introduced a family of cuts. These cuts are referred to as conditional cuts and are used within the overall branch-and-cut framework. The results show that the algorithm can solve large-scale instances to optimality within a reasonable time.

One could also solve the problem without using additional software to implement the exact mathematical model. Different researchers propose a step-by-step heuristic, using a constructive heuristic to obtain an initial solution and perform improvement steps. For example, Gomes, Diniz, and Martinhon [20] propose a Greedy Randomized Adaptive Selection Procedure (GRASP) heuristic combined with a Variable Neighbourhood Descent (VND), which is later extended by Chaves et al. [9] with a Variable Neighbourhood Search (VNS) to obtain upper bounds. Pedro, Saldanha, and Camargo [38] formulated a solution method using a constructive insertion method in combination with a LS improvement heuristic. The devised Tabu Search (TS) algorithm included basic features and was able to provide good solutions for larger instances in a short time. They concluded that their TS method with only basic features was able to outperform a more elaborated method such as the GRASP of Chaves et al. [9], for larger instances.

3.3. Simulation Approach

DES models are usually built to understand how systems behave over time and to compare their performance [45]. In these models, state changes occur at discrete points in time and generally, a stochastic nature is incorporated. Based on statistical distributions, randomness is generated and problems at an operational or tactical level are studied [46]. The reason in favor of a DES is the limited number of events. As we assume that the decision to swap is made every fixed time interval, a DES is in line with this study.

Illgen and Höck [25] test the performance of an electric car sharing network using a DES. In this simulation, important operational characteristics of electric vehicles were used. A comparison of different shared vehicle systems is presented and measured by vehicle utilization and service level. In this study, the comparison is made between different charging systems, whereas our study focuses on comparing solution methods to apply to one system. Further, El Fassi, Awasthi, and Viviani [15] present a decision support tool based on a DES that tests different growth strategies for car sharing systems to effectively satisfy the demand. They aim to select the best network growth strategies that meet demand while maximizing customer satisfaction and minimizing the number of vehicles used. Based on the demand distribution, they evaluate the performance of the proposed methods. They use different scenarios in their simulation and assign performance scores.

Ji et al. [26] propose a Monte Carlo simulation in which they assume the probability distributions are known. They simulate user demand and test the system availability of different operational concepts of a fully automated electric bicycle sharing system. In addition, the sensitivity of these results under various demand and supply scenarios is evaluated. The study of Campbell et al. [7] focuses on factors influencing the choice of shared electric bicycles in Beijing. They conclude that the demand for shared bicycles is strongly impacted by temperature, precipitation, and poor air quality. However, they mainly focus on the factors that influence people to rent a bicycle instead of using an alternative transportation mode such as public transport. In line with [7], Zhu et al. [50] concluded that precipitation and temperature suppress the usage of shared bicycles. Further, trips of short duration roughly have a linear relation between duration and distance. This suggests that such trips mainly follow the shortest paths at a constant speed in contrast to trips of long duration.

4
Methodology

In this chapter, we present two tour construction methods of which the performance is evaluated by means of a DES. We use a time horizon divided into time periods of equal duration in which two different events are included. The first event is based on the decision of our model regarding the construction of a tour visiting bicycles with low power batteries. The second event corresponds to new rental periods. Both events affect the battery levels, which are updated accordingly. Although customers prefer high power batteries, travel costs increase significantly by performing a tour often. Single tours do not show the performance of our methods in the long term. Therefore, a DES is proposed to test the quality by evaluating multiple time periods and swap operations. We aim to maximize the profit of the rental company, which is positively related to the amount of time the bicycles are rented, and negatively related to the working hours of a swapper. We face this trade-off and evaluate different system settings.

4.1. Discrete Event Simulation

The tour construction methods we propose are tested by means of a DES. It allows us to evaluate the performance over a specified period of time. In the short term, it is difficult to measure the impact of the decision made by the method(s). Therefore, the simulation uses a time horizon divided into time periods of equal duration, designed to iteratively check for events. The events correspond to the decision to construct a tour visiting low power batteries, and simulating new rentals that start within the next time period. If a new rental is simulated, the duration and travelled distance are also predicted.

At the beginning of each time period, we first evaluate all currently swap available bicycles. For this set, we apply a tour construction method and determine whether to swap batteries. After this decision, we simulate new rentals for all bicycles that are rental available. The simulation process is given in Algorithm 1.

Algorithm 1 Discrete Event Simulation

$b_0 \leftarrow$ Initialize the battery levels.
$ar_0 \leftarrow$ Initialize the availability for rentals.
$as_0 \leftarrow$ Initialize the availability for swaps.
for $t \leftarrow 1$ to T **do**
  tourConstruction()
  if tour constructed by the solution method **then**
    $b_t \leftarrow$ *updateBattery()*
  rentalSimulation()
  if any rental simulated **then**
    $b_t \leftarrow$ *updateBattery()*
    $ar_t \leftarrow$ *updateRentalAvailability()*
    $as_t \leftarrow$ *updateSwapAvailability()*

First, we have to set the initial battery levels and availability. The battery levels are based on historical data and we assume that all bicycles are available at the start of the simulation. The construction of a tour is based on the result of our methods which are discussed next. Note here, that our methods are able to decide not to construct a tour. In this case, no batteries are swapped corresponding to no change in battery level or availability.

Next, we describe the mathematical formulation of our tour construction problem, followed by two solution approaches. Lastly, we discuss the rental prediction and the update procedures on the bicycle statuses.

4.1.1. Tour Construction

We developed a bi-objective Mixed Integer Linear Program (MILP) formulation to construct a tour visiting low power batteries. In Table 4.1 an overview of the used notation is given. The problem can be formulated as an event-activity network, characterized by a directed graph $G = (V, E)$, where V represents the set of nodes or vertices, and E represents the set of arcs or edges. First, the relevant sets, parameters, variables, and functions are defined. Next, the exact mathematical model is presented, followed by a detailed description of the objective and constraints.

Table 4.1: Mathematical Notation

Set	**Description**
V	Set of nodes
V'	Set of nodes without the depot, $V \setminus \{0\}$
E	Set of arcs
T	Set of time periods

Parameter	**Description**
d_{ij}	Distance between node i and j
h_{ij}	Travel time between node i and j
h_{service}	Constant service time at a node
w_{prod}	Constant product weight
Q_{veh}	Constant vehicle capacity
as_i^t	Availability of node i in V' for the swap operation at time period t
ar_i^t	Availability of node i in V' for rentals at time period t
b_i^t	Battery level of node i in V' at time period t
r_i^t	Prize associated with visiting node i in V' at time period t
π_i^t	Penalty associated with visiting node i in V' at time period t
H_{max}^t	Maximum travel time of a tour at time period t
D_{max}	Maximum distance of a tour
v_{min}	Minimum amount of bicycles visited in a tour
b_{max}	Maximum battery of bicycles included in a tour

Variable	**Description**
x_{ij}	Binary variable indicating if arc (i, j) is used
y_i	Binary variable indicating if node i is visited

Function	**Description**
$f_r^t(\cdot)$	Function to calculate the prize associated with visiting node i
$f_\pi^t(\cdot)$	Function to calculate the penalty associated with not visiting node i
$f_q^t(\cdot)$	Function to calculate the quality of a constructed tour

In our research, set V is defined as the set of bicycle locations and the depot. The depot, at which the batteries are charged, is indicated by index 0. In total, we have n bicycle locations $i \in \{0, ..., n\}$, all connected by an arc. Therefore, set E includes an arc (i, j) for every node i and j. Those arcs are associated with a travel cost, corresponding to the distance, denoted by d_{ij}. In addition, we introduce travel time h_{ij} between node i and j, based on the distance and average speed.

Since we evaluate the performance of our model over multiple time periods, we define set $T = \{1, ..., t_{\text{max}}\}$. The swap availability as_i^t, rental availability ar_i^t, battery level b_i^t, prize r_i^t and penalty π_i^t of a node depend on time period t, to which the model is applied.

The tour constructed by the model is limited by a maximum duration H_{max}^t and distance D_{max}. The distance requirement is based on the driving range of the vehicle used to perform the tour. The duration is based on the working hours of the driver and therefore dependent on time period t. This requirement ensures that the end time of the tour is within working hours. The total duration is calculated as the sum of travel times plus the service times h_{service} of all visited nodes. Due to the vehicle capacity, the total transported weight can be at most Q_{veh}. In our problem, the product weights w_{prod} are constant and independent of node i.

The decision variables are used to obtain the optimal tour. Binary variable y_i represents whether node i is visited, and binary variable x_{ij} indicates whether arc (i, j) is used in the tour.

The aim is to obtain a solution that minimizes the travelled distance and maximizes the total collected prize, taking the constraints into account. As the prizes depend on the current status of every node individually, we calculate all prizes at the start of every time period as $r_i^t = f_r^t(b_i^t)$. Lastly, we impose two additional restrictions on the tour. We force the tour to visit a minimum amount of nodes v_{min}, and we do not include nodes with a remaining capacity b_i^t greater than b_{max}. The following MILP can be defined:

$$\text{Minimize:} \quad \sum_{(i,j)\in V} d_{ij}x_{ij} \tag{4.1}$$

$$\text{Maximize:} \quad \sum_{i\in V'} r_i^t y_i \tag{4.2}$$

$$\text{s.t.} \quad \sum_{j\in V} x_{ij} = y_i, \qquad i \in V, i \neq j \tag{4.3}$$

$$\sum_{i\in V} x_{ij} = y_j, \qquad j \in V, j \neq i \tag{4.4}$$

$$\sum_{i\in V'} r_i^t y_i - \sum_{(i,j)\in E} d_{ij}x_{ij} \geq 0 \tag{4.5}$$

$$\sum_{(i,j)\in E} h_{ij}x_{ij} + \sum_{i\in V'} h_{\text{service}} y_i \leq H_{\max}^t \tag{4.6}$$

$$\sum_{(i,j)\in E} d_{ij}x_{ij} \leq D_{\max} \tag{4.7}$$

$$\sum_{(i,j)\in S} x_{ij} \leq \sum_{j\in S\setminus k} y_j, \qquad S \subset V', k \in S \tag{4.8}$$

$$\sum_{i\in V'} v_{\min} x_{0i} \leq \sum_{i\in V'} y_i \tag{4.9}$$

$$\sum_{i\in V'} w_{\text{prod}} y_i \leq Q_{\text{veh}} \tag{4.10}$$

$$\sum_{i\in V'} y_i \leq M y_0 \tag{4.11}$$

$$y_i \leq as_i^t, \qquad i \in V' \tag{4.12}$$

$$b_i^t y_i \leq b_{\max}, \qquad i \in V' \tag{4.13}$$

$$y_i \in \{0,1\}, \qquad i \in V' \tag{4.14}$$

$$x_{ij} \in \{0,1\}, \qquad (i,j) \in E \tag{4.15}$$

The objective consists of two functions, referred to as a bi-objective formulation. The first (4.1) minimizes the total travelled distance and the second (4.2) maximizes the total collected prize. In addition, we add constraint (4.5) to ensure that the collected prizes exceed the travel costs. These prizes are calculated using step-wise function $f_r^t(\cdot)$ for every available node i assigning higher prizes to nodes corresponding to a low battery level.

Next, we discuss all constraints. The first constraints (4.3) and (4.4) ensure that the decision variables are correctly related. If an in- or outgoing arc is used for node i, it is visited and y_i must be equal to 1, else y_i must be equal to 0. These constraints also ensure that a node is visited at most once. Additionally, constraints (4.12) ensure that a node can be visited if it is available in the current time period. If not, the tour does not include this node and y_i is forced to be 0. Constraint (4.13) ensures that nodes with a remaining capacity above $b_{\max}$ are not included in the tour.

Constraints (4.8) force every visited node to be connected to the depot. These represent the subtour elimination constraints, proposed by Feillet, Dejax, and Gendreau [16]. If an edge $i \in V'$ is visited, it is necessarily directly or indirectly connected to the depot. For n locations, the number of possible sets S adds up to 2^n, i.e. the number of constraints grows exponentially. Therefore, we start by relaxing these constraints and the resulting problem is solved to integer optimality. Then, we test for subtours and if the solution contains subtours, the violated constraints are added and the process is repeated until a feasible solution is found. Instead of adding constraints for all the possible sets, only some constraints are added. In addition, constraint (4.11) is used to ensure that the tour includes the depot if any node is visited. In this constraint, M refers to a large value.

Further, constraint (4.6) limits the tour duration, calculated as the total travel time plus the service times of the visited nodes. In addition, constraint (4.7) limits the distance of the tour. As the vehicle has a limited driving range, the total travelled distance is bounded. Constraint (4.9) ensures that we either visit zero nodes, or at least the minimum amount $v_{\min}$. Lastly, constraint (4.10) imposes the vehicle capacity. As the product weights are all equal, this constraint limits the maximum number of nodes we visit.

Constraints (4.14) and (4.15) are the variable restrictions imposing binary variables.

4.1.2. Solution Approach 1: Mono-objective Mathematical Model

In this section, we discuss the most common reformulation of the bi-objective model, a linear scalarization of the objectives. In our literature review (3), we introduced the PTP. We adapt our objective as follows:

$$\text{Minimize:} \quad \sum_{(i,j)\in E} d_{ij}x_{ij} + \sum_{i\in V'} \pi_i^t(1-y_i) \tag{4.16}$$

As can be seen in equation (4.16), the travel costs and the penalties incurred by not visiting a node are minimized. In contrast to the original formulation (4.1-4.15), we use penalties for not visiting instead of prizes for visiting a node. These penalties are calculated similarly: $\pi_i^t = f_\pi^t(as_i^t, b_i^t) = f_r^t(as_i^t, b_i^t)$.

In our first solution approach, the linear objective (4.16) is implemented and constraints (4.3-4.15) remain in the formulation.

4.1.3. Solution Approach 2: Local Search

In Section 3 it was stated that the bi-objective TSP, PTP, and PCTSP are $\mathcal{NP}$-hard. Therefore, it is unlikely that an exact solution can be found for large instances. To obtain solutions in a reasonable time, we propose an alternative solution method based on a LS. This method does not use additional computational software. The initial solution is constructed in a greedy way by iteratively adding nodes to the tour while taking the constraints into account. The second step is to perform improvement steps to tighten the upper bound on the problem. In this section, more details on this LS heuristic are given.

Initialization Heuristic

The initialization heuristic is based on the insertion method. Note that the insertion of a node y_i to the tour corresponds to an increase in both collected prizes and costs. To measure the quality of the constructed tour, we define a formula assigning a score based on the prizes that are currently collected. Further, the costs of the tour are taken into account. Assume that $y_i = 1$ for all nodes included in the tour, and $x_{ij} = 1$ for all arcs (i, j). The resulting formula to measure the quality is stated in equation (4.17).

$$f_q^t(\cdot) = \sum_{i \in V'} r_i^t y_i - \sum_{(i,j) \in E} d_{ij} x_{ij} \tag{4.17}$$

The constructed tour depends on the first inserted node, which we randomly choose from a sorted priority list. The list contains nodes that have the highest priority and prizes, independent of the travel costs. Once the first node is set, we start adding new nodes based on the cheapest insertion method. The tour construction stops if adding new nodes does not yield improvement. Then, the constructed tour is added to the list of initial tours if no constraints are violated. In Algorithm 2 an overview of the initialization heuristic is given.

Algorithm 2 Initialization Heuristic Algorithm

```
sorted ← Generate a sorted priority list of nodes.
initialTours← Create an object to store the initial tours.
for i ← 1 to #Starts do
    n_0 ← Select a random node from sorted
    sorted← Delete n_0 from sorted
    currTour ← Construct a new tour starting and ending at the depot, visiting n_0
    while Improvement and tour constraints not violated do
        currTour ← Add node based on Cheapest Insertion
    initialTours ← Add currTour to initialTours

Return: initialTours
```

Next, we discuss the cheapest insertion method in more detail. Each time we insert a new node, the profit/cost ratio is maximized. We denote the profit of adding a node as δ_i^p and costs as δ_i^c. The costs are calculated as the additional travel costs of visiting node i. Once we decide to insert a node, we choose the best spot in the current tour, for which δ_i^c is minimized. The profit is equivalent to the corresponding prize r_i^t. In Algorithm 3 an overview of the cheapest insertion method is given.

Algorithm 3 Cheapest Insertion Algorithm

bestRatio $\leftarrow$ Create an object to store the best profit/cost ratio of inserting a node.
bestNode $\leftarrow$ Create an object to store the node corresponding to the *bestRatio*.
bestSpot $\leftarrow$ Create an object to store the best spot to insert the *bestNode*.
for $i \leftarrow 1$ to #AvailableNodes **do**
    for all spots in the tour between node r and s **do**
        $\delta_i^p \leftarrow r_i^t$
        $\delta_i^c \leftarrow d_{ri} + d_{is} - d_{rs}$
        if $\delta_i^p/\delta_i^c \geq$ *bestRatio* **then**
            bestRatio $\leftarrow \delta_i^p/\delta_i^c$
            bestNode $\leftarrow$ Node i
            bestSpot $\leftarrow$ Between node r and node s

Return: *bestNode, bestSpot*

The initialization heuristic is performed multiple times due to the greedy component of adding the first node from a priority list. As we use a multi-start mechanism, we obtain multiple initial solutions to which we apply the improvement heuristic.

Improvement Heuristic

The solutions of the initialization heuristics can be enhanced using the improvement heuristic. In most cases, changing the tour leads to an improvement of one of the objectives at the expense of the other. Therefore, during the improvement phase, we use equation (4.17) to measure the quality of the solution. Four operations are included in our heuristic to improve the current tour by resequencing, replacing, deleting, or adding nodes. The heuristic is designed to avoid the solution from being stuck at a local optimum and prevent cycling.

During the procedure, many potential tours are constructed. As some of the operations cause a score decrease to avoid local optima, we store interim solutions. Note that a solution is stored if it belongs to the class of best solutions found. The stopping criterion is based on the scores of two successive iterations. The algorithm terminates if the score does not increase, and no improvement has been found. If the best found solution yields a positive quality score based on equation (4.17), the tour will be performed. In Algorithm 4 an overview is given.

Algorithm 4 Improvement Heuristic

```
currBest ← Create an object to store the best solution found
Operations ← {Resequence, Replace, Deletion, Addition}
for i ← 1 to #InitialTours do
    currTour ← Initial tour i
    while Improvement between two successive iterations do
        for Op in Operations do
            while Improvement do
                currTour ← Op
            if currTour outperforms currBest then
                currBest ← currTour

Return: currBest
```

Resequencing

The first step of the improvement heuristic is resequencing the tour. This operation cannot deteriorate the solution, as costs can decrease and the collected prize does not change. It is the most widely used heuristic method for classic TSPs and is referred to as k-opt [21]. It can be described as an attempt to improve the solution by changing the sequence of k nodes in the tour. In our research, we limit ourselves to the 2-opt with regard to the computation time.

We resequence two nodes not necessarily directly connected by an arc and carry out the operation corresponding to the highest score increase. If changing the sequence of two nodes does not yield improvement, we continue with the replacement operation.

Replacement

The second step attempts to improve the solution by replacing a node in the tour with a node outside the tour. We consider the variant of replacement in which we decide to replace a node if and only if the score calculated by equation (4.17) increases. We replace the node corresponding to the largest score increase and repeat this step until no improvement is possible.

Deletion

The third step considers the deletion of nodes. In this step, we delete at least one node. This can lead to an increase or a decrease, considering equation (4.17). Apart from the first, we delete nodes corresponding to a score increase until no improvements are possible.

Addition

The last step evaluates whether it is profitable to add nodes that are currently not in the tour. Based on the profit/cost ratio, we insert nodes. Using equation (4.17), we evaluate whether adding the best-found candidate leads to an improvement. Nodes are added until no improvements are possible or the vehicle capacity is reached.

4.1.4. Rental Simulation

In the DES, we attempt to accurately simulate new rental periods based on historical data. The rentals of all bicycles are assumed to be independent and generated separately. We describe a rental by the start time, the end time, the duration, and the travelled distance.

The first evaluation of our data showed seasonality. We assume that demand for bicycles depends on two factors related to time: the day of the week, and the time of the day. Further, we take weather conditions into account, as people tend to prefer cycling in good conditions. We distinguish between *Bad* and *Good* weather, which is described later. In Table 4.2 an overview is given. In total, we distinguish between 56 situations.

Table 4.2: Demand Situation Specification

Factor	**Categories**
Day	Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday
Time	[6am, 10am], [10am, 2pm], [2pm, 6pm], [6pm, 10pm]
Weather	Bad, Good

Each time period t corresponds to one of the specified situations on which the probability of a rental depends. Next, we address the three main questions: What is the probability that bicycle i is rented in time period t? Given that is it rented, what is the duration of the rental? Given that it is rented, what is the travelled distance?

Probability of Rentals

For each bicycle, we need to estimate the probability of a rental in time period t. As introduced, we distinguish between different situations based on the day, time, and weather. To calculate p^t_{rental}, we evaluate all rentals in our historical data and formulate this probability as follows. Note that a situation occurs more than once in our historical data. For each situation, we evaluate the number of times a bicycle was actively participating in the network, and the number of times a rental period was started. By dividing those values, we determine the probability of a rental.

Prediction Duration and Travelled Distance

The rental duration is also dependent on the specified situations, and we estimate the probability distribution based on our historical data. Then, samples from the learned probability distribution can be drawn in the DES. Similarly, we estimate the probability distribution of the travelled distance.

In our initial data analysis, we tested for a correlation between duration and distance. As the results showed a significant correlation, we introduce a method to draw a sample from correlated random variables.

The joint cumulative distribution function (CDF) is approximated using the Gaussian copula. The key property of a copula correlation model is preserving the original marginal distributions while defining a correlation between them, first described by Sklar [43]. Based on the marginal CDFs and the correlation coefficient, the joint CDF is approximated. The general procedure can be explained in three steps and an overview is given in Algorithm 5.

Algorithm 5 Sampling Correlated Variables using Gaussian Copula

1: Draw $Z = (Z_1, Z_2) \sim \mathcal{N}(\mu, \Sigma)$, where μ is the mean and $\Sigma(\cdot)$ is the correlation matrix.
2: Set $U_i = \Phi(Z_i)$ for $i = 1, 2$, where Φ is the standard normal CDF.
3: Set $Y_i = F_i^{-1}(U_i)$ for $i = 1, 2$, where $F_i^{-1}(\cdot)$ is the inverse of the marginal CDF of variable i.

The first step is to draw a sample from a normal distribution using the constant correlation coefficient ρ between the duration and distance. The mean is given by $\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, and the correlation matrix by $\Sigma = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$.

The resulting correlated normal variables $Z = (Z_1, Z_2)$ are transformed into correlated standard normal variables using the standard normal CDF ($\Phi(\cdot)$). The resulting random variables U_i are dependent and uniformly distributed over $[0, 1]$. Lastly, we perform inverse transformation sampling using the inverse of the marginal CDF of both variables separately.

The resulting Y_i variables are drawn from the original marginal distributions and correlated. Given that a bicycle is rented in the considered time period, we predict the duration and travelled distance by this procedure. The end time follows from the start time and duration. In Section 6 we discuss the estimation of the marginal distributions and elaborate further on this procedure.

4.1.5. Update Procedures

An important part of the DES is the status of all bicycles. In this section, we describe the update procedures regarding the availability (as_i^t, ar_i^t) and battery levels (b_i^t). The updates are based on constructed tours and rental periods. Variable as_i^t is equal to 1, if bicycle i is available for the swap procedure in time period t. Variable ar_i^t is equal to 1, if bicycle i is available for rentals in time period t. Variable b_i^t indicates the remaining battery level in kilometers. Note that ar_i^t is equal to 0, if b_i^t is less than the minimum required battery level b_{low}.

The construction of a tour is the first event resulting in a status update. We determine at which time period a bicycle is visited due to the travel time, and update the battery level to the maximum once it is visited.

The second event changing the status of a bicycle is a rental period. Each time period we consider the rental available bicycles and simulate new rentals. In the update procedure, we include the bicycles rented in period t. During the rental period, bicycles are not available for swaps or new rentals and the statuses are adapted accordingly. The remaining battery level is reduced with the simulated travelled distance.

5
Data

In this chapter, a detailed description of the available data is given. First, a dataset regarding the fixed locations is described, and second, a dataset on all bookings is explained. The datasets are provided by Cargoroo, which is a rental company of e-cargo bicycles. Lastly, we discuss an additional dataset on weather conditions to make our data complete.

5.1. Fixed Locations

Each bicycle has a fixed location, explained by longitude and latitude, at which the battery must be swapped. In Table 5.1 an overview of the dataset is given. Each row corresponds to a location, described by the Station ID, City, Longitude, and Latitude.

Table 5.1: Data Sample Locations

	Station ID	City	Longitude	Latitude
1	100	Amsterdam	4.889184	52.356905
2	101	Amsterdam	4.888288	52.356273
3	14	The Hague	4.289448	52.083572
4	15	The Hague	4.290488	52.080733
5	206	Utrecht	5.117606	52.063138
6	207	Utrecht	5.116863	52.064704

In this research, we consider three different cities. In total, we have 293 bicycle locations, of which 121 are in Amsterdam, 78 in The Hague, and 94 in Utrecht.

5.2. Bookings

The dataset that contains all bookings is structured as follows. Each row corresponds to a booking, which includes: the unique Booking ID, the Start time, the End time, the Duration, the Distance travelled, and the Station ID. The data provides inside into the historical rental periods and customer behaviour. In Table 5.2 a sample of the dataset is shown.

Table 5.2: Data Sample Bookings

	Booking ID	Start	End	Dur. (sec)	Dist. (km)	Station ID
1	6bcc5-0d60	6/26/21, 09:42	6/26/21, 12:51	11352	22.49	212
2	cd138-d569	6/26/21, 13:19	6/26/21, 16:19	10827	12.83	212
3	4b218-7053	6/26/21, 19:55	6/26/21, 20:21	1573	1.10	212
4	d3bc3-1bda	7/01/21, 17:43	7/01/21, 18:17	2072	6.26	212
5	75849-4c26	7/02/21, 15:08	7/02/21, 16:07	3532	1.37	212
6	302c5-f619	7/03/21, 09:27	7/03/21, 12:19	10296	9.56	212

Before processing the dataset, it contains 130537 recorded bookings, of which 38977 in Amsterdam, 58008 in The Hague, and 33552 in Utrecht. The data includes exactly one year of recorded bookings, between July 2021 and June 2022. In Appendix A, an overview of our filters regarding a valid booking is given. The data is cleaned and erroneous observations, or observations not matching our assumptions, are excluded.

The duration of the filtered bookings ranges from 598 seconds (10.0 minutes) to 51862 seconds (15.5 hours), and the travelled distance from 0.5 kilometers to 49.9 kilometers.

5.3. Weather Conditions

In this section, we discuss an additional dataset, which is not provided by Cargoroo. This open source dataset is obtained from the Koninklijk Nederlands Meteorologisch Instituut (KNMI[1]) and contains information on the weather condition of a specific day. Based on this data, we evaluate whether a booking was made on a day with bad or good weather. In Table 5.3, part of the dataset is shown.

Table 5.3: Data Sample Weather

	City	Date	Temp. (°C)	Precipitation dur. (h)	Precipitation (mm)
1	Utrecht	20/10/2020	11.7	1.2	0.3
2	Utrecht	21/10/2020	15.4	9.5	8.6
3	Utrecht	22/10/2020	13.9	0.5	0.5
4	Utrecht	23/10/2020	12.0	0.0	0.0
5	Utrecht	24/10/2020	13.9	0.6	0.2
6	Utrecht	25/10/2020	12.1	1.4	4.1

We use three different measures to describe the weather, one regarding temperature and two regarding precipitation. The mean temperature over one year is 10.8 °C. Daily, the average precipitation duration is 1.8 hours and the precipitation is 2.4 millimeters. A booking is considered to be made on a *Good* weather day if the temperature is greater or equal than 7.5 °C, the precipitation duration is less than 3 hours, and the precipitation is less than 2 millimeters. If one of those criteria does not hold, a booking is made on a *Bad* weather day.

[1] https://www.knmi.nl/nederland-nu/klimatologie/daggegevens

6
Results

In this chapter, the computational results for our introduced methods are evaluated on three different instances. We evaluate the performances and analyze the influence of the parameters. Further, we consider the managerial trade-off between the battery levels and the travel time to swap the batteries.

The mathematical models are coded in Java (Eclipse IDE for Java Developers version 4.24.0) and CPLEX (version 20.1.0), and run on a single thread of a 11th Gen Intel(R) Core(TM) i5-1135G7 CPU with 32GB of RAM. We solve the instances until an optimality gap of 0.01% is reached. Furthermore, we impose a time limit of 5 minutes per time period on each instance. If we are not able to solve one of the time periods within our time horizon, we do not obtain solutions on performance indicators as the method is terminated. Next, we describe the instances and discuss the results.

6.1. Test Instances

The data discussed in Section 5 describe the rentals in Amsterdam, The Hague, and Utrecht. The rental systems operate independently, and we evaluate the performance of our methods on each city separately. Each city contains a different amount of bicycles, and the instance sizes vary. First, we address the instances followed by the probability distributions of the travelled distance and duration.

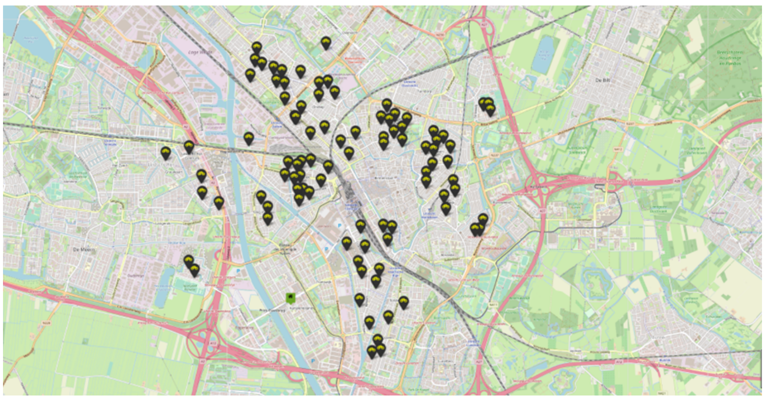
6.1.1. Locations

Each instance has a specific spread of bicycles over the city. In Figure 6.1 a geographic overview is given. In this figure, the depot is marked with a green square. We observe a varying density, as the bicycles in some regions are sparse. It is likely that swapping those batteries is relatively costly, as a lot of travelling comes along. An interesting remark is that no bicycles are located in the centre of all cities due to governmental policies. To prevent the street scene from being compromised, there is a prohibition to locate the bicycles in the heart of the centre. In The Hague, the prohibition does not hold for the depot as can be seen in Figure 6.1b. The depot of Utrecht is located on the outskirts of the city, requiring the swapper to travel a lot. The average, minimum, and maximum distances of the instances are given in Appendix B.

Figure 6.1: Locations of e-cargo bicycles

(a) Amsterdam

(b) The Hague

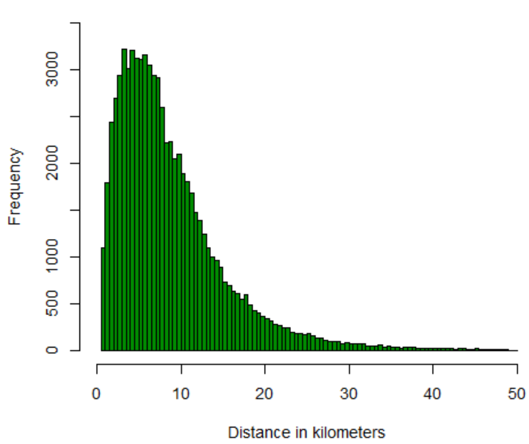


(c) Utrecht

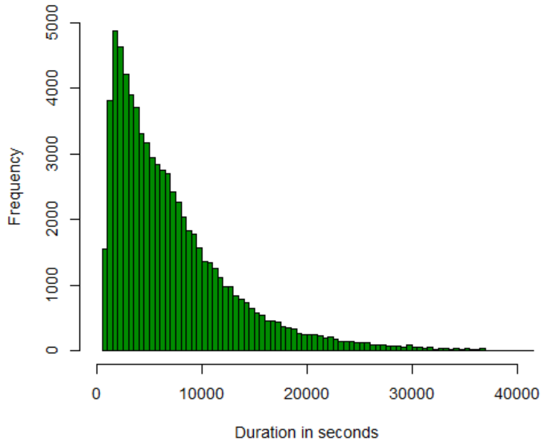
6.1.2. Probability Distributions

The DES depends on the probability distribution of the rental duration and travelled distance, and we assume that the demand depends on the situations described in Section 4.1.4. Figure 6.2 shows the histograms of the travelled distance (6.2a) and the duration (6.2b) aggregated over all situations and bicycles. Both histograms are right skewed and long tailed.

Figure 6.2: Histograms of travelled distance and rental duration based on historical bookings



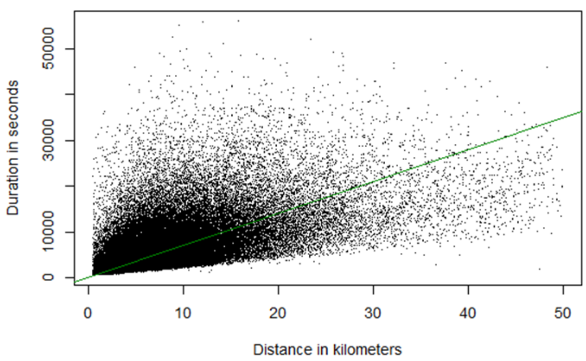
(a) Travelled distance



(b) Rental duration

In our analysis, we evaluated different probability distribution functions to fit our data. The results show that the lognormal distribution fits the travelled distance and duration best. Therefore, for each situation, we estimated the means $\mu_{\log}^{\text{dis}}$ and $\mu_{\log}^{\text{dur}}$, and standard deviations $\sigma_{\log}^{\text{dis}}$ and $\sigma_{\log}^{\text{dur}}$ (Appendix C). The distribution is continuous and takes only positive real values. These properties are suited for our simulation. In Figure 6.3 a scatter plot of the distance and duration is shown. We observe a positive correlation, highlighted by the regression line. The correlation coefficient used in the Gaussian copula method is $\rho = 0.561$.

Figure 6.3: Scatter plot of travelled distance and rental duration with a regression line through the origin



6.2. Numerical Results

In this section, we discuss the results on the introduced instances. First, the parameter settings are given followed by the corresponding results. The results of the exact mathematical model and the heuristic approach are discussed separately.

6.2.1. Parameter Settings

In this research, the aim is to create an efficient tour for a battery swapper. To evaluate the performance of our methods, we use a DES with a time horizon of 10 weeks. We consider 3360 time periods of 30 minutes each. Furthermore, bicycles can be rented by customers between 7am and 10pm, and swappers work from 6am to 10pm.

In our formulation, several parameters are included as given in Table 4.1. The service time (h_{service}) is equal to 3 minutes, the product weight (w_{prod}) is equal to 1 and the vehicle capacity (Q_{veh}) is equal to 16. Further, we calculate the maximum duration of a tour based on the current time period. For example, starting a tour at 9pm corresponds to a $H^t_{\max}$ of one hour. If the working hours are not limiting, we impose a maximum of 3 hours. Further, the distances (d_{ij}) are calculated using the Distance Matrix Service of Google Maps Platform[1]. Based on those distances, we calculate the travel times (h_{ij}) using an average speed of 15km/h. Lastly, the maximum tour length ($D_{\max}$) is 100 kilometers.

The prizes (r^t_i) and penalties (π^t_i) depend on the battery statuses in each time period, and are scaled to the travel distances in meters. In Appendix D the definition and plot of stepwise function $f^t_r(\cdot) = f^t_\pi(\cdot)$ can be found.

Parameters $v_{\min}$ and $b_{\max}$ are used to formulate different strategies. Varying over these parameters affects the tour lengths and the average battery levels over time. The results are based on the settings for which $v_{\min} \in \{10, 13, 16\}$ and $b_{\max} \in \{40, 50, 60, 70\}$. As rental periods depend on the weather conditions, we distinguish between good and bad weather conditions as defined in Section 5. In total, we evaluate the results of 24 settings.

Further, the results are based on 100 runs to correct for the randomness of the rental periods. For example, the number of tours is calculated as the average over 100 simulations of 10 weeks.

Lastly, we define the number of starts for the initialization heuristic of the LS. The method has a multi-start mechanism and constructs at most 16 initial tours. Then, the improvement heuristic is applied to those tours.

[1] https://developers.google.com/maps/documentation/distance-matrix

6.2.2. Profitable Tour Problem Results

First, we evaluate the results of the proposed exact mathematical method. As introduced in Section 4.1.4, we distinguish between different weather conditions. Table 6.1 shows the results in times of bad weather and Table 6.2 in times of good weather.

Table 6.1: Overview of the PTP model results with bad weather conditions averaged over 100 complete DES runs. It includes the average battery levels of all rental available bicycles, the average time spent to swap one battery, the average number of batteries replaced per tour, the number of tours, the number of times a battery level crosses the threshold b_{low}, and the number of times a bicycle is empty

Min. Tour Length (v_{min})	**10**		**13**	
Max. Battery Level (b_{max})	**40**	**50**	**40**	**50**
City: Amsterdam				
Avg. Battery (%)[1]	67.5	**75.3**	-	-
Avg. Trav. time (min)[2]	9.9	**9.7**	-	-
Avg. Visits per tour[3]	**10.2**	10.1	-	-
Tot. Trav. time (h)[3]	**168**	201	-	-
Nr. Tours[3]	**77**	93	-	-
Nr. Below 30%[3]	259	**106**	-	-
Nr. Empty 0%[3]	11	**6**	-	-
Comp. time (sec)[4]	**25.0**	33.7	-	-
City: The Hague				
Avg. Battery (%)[1]	66.4	**74.6**	-	-
Avg. Trav. time (min)[2]	7.0	**7.1**	-	-
Avg. Visits per tour[3]	**10.2**	10.1	-	-
Tot. Trav. time (h)[3]	**82**	100	-	-
Nr. Tours[3]	**48**	59	-	-
Nr. Below 30%[3]	181	**76**	-	-
Nr. Empty 0%[3]	7	**4**	-	-
Comp. time (sec)[4]	**10.7**	13.7	-	-
City: Utrecht				
Avg. Battery (%)[1]	67.1	**75.0**	66.5	74.3
Avg. Trav. time (min)[2]	8.4	8.4	**7.1**	7.2
Avg. Visits per tour[3]	10.2	10.2	**13.3**	13.2
Tot. Trav. time (h)[3]	114	138	**100**	122
Nr. Tours[3]	59	71	**45**	54
Nr. Below 30%[3]	209	**86**	227	101
Nr. Empty 0%[3]	9	**5**	9	6
Comp. time (sec)[4]	**9.7**	13.2	23.1	33.7

[1] of all rental available bicycles over 10 weeks
[2] average time to swap one battery over 10 weeks
[3] over/in 10 weeks [4] to solve one complete DES (10 weeks)

The results of the PTP show that our model is not able to obtain solutions within our time limit of 5 minutes per time period on each instance. We observe that the model reaches the time limit with the parameter settings $v_{\text{min}} = 16$, $b_{\text{max}} = 60\%$, or $b_{\text{max}} = 70\%$.

First, we discuss the results corresponding to $v_{\text{min}} = 10$. Recall that b_{max} corresponds to the maximum battery level allowed to be included in the tour, and b_{low} to the minimum required battery level to start a rental.

For each instance, we observe similar results on the average battery levels, which are approximately 67% and 75% for $b_{\text{max}} = 40\%$ and $b_{\text{max}} = 50\%$ respectively. The effect of changing b_{max} is that batteries of higher levels are allowed to be swapped if it is beneficial relative to the extra travel costs. In addition, we observe that the number of bicycles reaching a battery level below $b_{\text{low}} = 30\%$ decreases considerably by 58.6% averaged over the instances if we increase b_{max}. This is at the expense of an average increase in the number of tours of 20.2%. Per tour, the average number of visited bicycles does not change for a different b_{max}. Hence, more batteries are swapped over the time horizon.

The instance corresponding to the network of Utrecht shows results for $v_{\text{min}} = 13$. The average travel time spent to swap one battery decreases by 15% due to extended tours. However, this comes with an increase in the number of battery levels crossing threshold $b_{\text{low}} = 30\%$ as we postpone the performance of a tour. Further, we construct 23% fewer tours and the total travel times decrease by 13%, emphasizing the efficiency gain of tours visiting more bicycles.

The results show different computation times and increase with the instance size and parameter v_{min}. The largest instance (Amsterdam) shows the highest computation times, and the smallest instance (The Hague) shows the lowest computation times. Increasing the instance size and minimum tour length both lead to more participating bicycles in the tour construction method. Therefore, it comes with more possible subtours which mainly cause the increase in computation time of our exact model. For every violated subtour, the elimination constraints are added and the model is resolved. A minimum tour length of $v_{\text{min}} = 16$ causes the time limit to be reached for each instance due to this time consuming iterative procedure.

Table 6.2 shows the results of our PTP model in times of good weather. Since customers rent a bicycle more often, the number of rental periods increases. In addition, both the average travelled distance and rental duration increase. As a result, the total travelled distance in our time horizon of 10 weeks increases.

First, we evaluate the number of tours performed by the swapper. On average, 44% more tours are constructed and the total travel time shows an increase of 44%. It follows that the average travel time per tour is similar. Further, the average time spent to visit one bicycle is not affected.

Next, we consider the effect on the average battery levels. The results show an absolute difference of at most 0.5%, corresponding to the instance of Utrecht with $v_{\text{min}} = 13$ and $b_{\text{max}} = 50$. Constructing more tours thus succeeds in compensating for the increase in demand.

Despite the unchanged average battery levels, we observe a difference in the number of

Table 6.2: Overview of the PTP model results with good weather conditions averaged over 100 complete DES runs. It includes the average battery levels of all rental available bicycles, the average time spent to swap one battery, the average number of batteries replaced per tour, the number of tours, the number of times a battery level crosses the threshold b_{low}, and the number of times a bicycle is empty

Min. Tour Length (v_{min})	**10**		**13**	
Max. Battery Level (b_{max})	**40**	**50**	**40**	**50**
City: Amsterdam				
Avg. Battery (%)[1]	67.4	**75.1**	-	-
Avg. Trav. time (min)[2]	9.8	**9.7**	-	-
Avg. Visits per tour[3]	**10.3**	10.2	-	-
Tot. Trav. time (h)[3]	**243**	313	-	-
Nr. Tours[3]	**111**	134	-	-
Nr. Below 30%[3]	430	**197**	-	-
Nr. Empty 0%[3]	24	**15**	-	-
Comp. time (sec)[4]	44.9	**44.2**	-	-
City: The Hague				
Avg. Battery (%)[1]	66.3	**74.3**	-	-
Avg. Trav. time (min)[2]	**7.0**	**7.0**	-	-
Avg. Visits per tour[3]	**10.2**	**10.2**	-	-
Tot. Trav. time (h)[3]	**119**	143	-	-
Nr. Tours[3]	**70**	84	-	-
Nr. Below 30%[3]	295	**141**	-	-
Nr. Empty 0%[3]	16	**10**	-	-
Comp. time (sec)[4]	**25.2**	28.9	-	-
City: Utrecht				
Avg. Battery (%)[1]	67.0	**74.7**	66.1	73.8
Avg. Trav. time (min)[2]	8.3	8.4	**7.1**	**7.1**
Avg. Visits per tour[3]	10.3	10.3	**13.4**	13.3
Tot. Trav. time (h)[3]	166	199	**145**	175
Nr. Tours[3]	85	102	**64**	78
Nr. Below 30%[3]	344	**161**	372	181
Nr. Empty 0%[3]	19	**11**	19	12
Comp. time (sec)[4]	**12.3**	15.8	27.8	50.7

[1] of all rental available bicycles over 10 weeks
[2] average time to swap one battery over 10 weeks
[3] over/in 10 weeks [4] to solve one complete DES (10 weeks)

(near) empty batteries. For example, the instance corresponding to Amsterdam shows an increase of 75.9% in the number of batteries below 30% and shows 134% more empty batteries. As bicycles with a remaining battery level below 30% are not rental available, we want to prevent this from occurring. We observe that these situations occur often for each instance, so we want to swap the batteries earlier. Increasing the battery participation level b_{max} allows for earlier battery swaps and avoids risk. Unfortunately, due to the computational complexity of the subtour elimination constraints, we do not obtain those results within our time limit.

Compared to the results with bad weather conditions, we observe higher computation times to solve one complete DES which is in line with the increased number of tours and rentals.

6.2.3. Local Search Results

Next, we evaluate the results of the proposed heuristic method. Table 6.3 corresponds to the results in times of bad weather conditions and Table 6.4 in times of good weather conditions. The LS is able to find a solution within the time limit for all parameter settings in contrast to our exact model.

Table 6.3: Overview of the LS results with bad weather conditions averaged over 100 complete DES runs. It includes the average battery levels of all rental available bicycles, the average time spent to swap one battery, the average number of batteries replaced per tour, the number of tours, the number of times a battery level crosses the threshold b_{low}, and the number of times a bicycle is empty

Min. Tour Length (v_{min})	**10**				**13**				**16**			
Max. Battery Level (b_{max})	**40**	**50**	**60**	**70**	**40**	**50**	**60**	**70**	**40**	**50**	**60**	**70**
City: Amsterdam												
Avg. Battery (%)[1]	67.1	74.5	78.4	**81.0**	66.3	73.4	77.3	80.1	64.3	70.5	74.3	76.9
Avg. Trav. time (min)[2]	11.8	11.0	9.8	8.7	10.9	10.2	9.3	8.5	10.2	9.7	9.1	**8.4**
Avg. Visits per tour[3]	11.3	11.4	11.6	12.2	14.2	14.3	14.4	14.5	**16.0**	**16.0**	**16.0**	**16.0**
Tot. Trav. time (h)[3]	152	168	174	178	137	153	161	169	**122**	135	146	154
Nr. Tours[3]	69	81	92	101	53	63	72	82	**44**	52	60	68
Nr. Below 30%[3]	278	134	92	**77**	312	161	108	89	393	239	174	148
Nr. Empty 0%[3]	11	7	6	**5**	12	8	6	5	13	9	7	6
Comp. Time (sec)[4]	**0.4**	0.5	0.9	1.8	0.5	0.6	1.1	2.2	1.0	1.3	2.0	3.1
City: The Hague												
Avg. Battery (%)[1]	66.2	74.1	78.5	**81.6**	65.5	73.2	77.7	81.1	64.6	72.1	76.7	80.2
Avg. Trav. time (min)[2]	9.6	9.3	8.6	7.8	8.7	8.5	8.0	7.4	8.2	8.0	7.6	**7.2**
Avg. Visits per tour[3]	10.7	10.8	11.1	11.7	13.8	13.9	14.0	14.1	**16.0**	**16.0**	**16.0**	**16.0**
Tot. Trav. time (h)[3]	78	91	99	104	70	81	90	98	**64**	75	84	92
Nr. Tours[3]	46	54	63	69	35	42	48	56	**29**	35	41	48
Nr. Below 30%[3]	188	84	51	**40**	205	96	59	44	229	113	69	50
Nr. Empty 0%[3]	7	4	**2**	3	8	5	3	3	8	5	4	3
Comp. time (sec)[4]	**0.2**	0.5	0.4	1.6	0.3	0.4	0.5	2.0	0.5	0.4	0.6	2.0
City: Utrecht												
Avg. Battery (%)[1]	67.0	74.7	78.7	**81.2**	66.3	73.8	78.0	80.8	65.5	72.9	77.2	80.1
Avg. Trav. time (min)[2]	11.0	10.7	9.8	8.8	9.9	9.7	9.1	8.4	9.2	9.1	8.7	**8.2**
Avg. Visits per tour[3]	11.0	11.1	11.6	12.9	14.0	14.1	14.2	14.6	**16.0**	**16.0**	**16.0**	**16.0**
Tot. Trav. time (h)[3]	110	129	136	139	97	115	125	132	**88**	105	116	126
Nr. Tours[3]	55	65	72	74	42	50	58	65	**36**	43	50	58
Nr. Below 30%[3]	214	92	56	**44**	237	106	63	49	262	124	74	53
Nr. Empty 0%[3]	8	6	4	**3**	9	5	4	**3**	10	6	4	4
Comp. time (sec)[4]	**0.2**	0.3	0.5	1.2	0.3	0.3	0.6	1.4	0.4	0.4	0.7	1.6

[1] of all rental available bicycles over 10 weeks [2] average time to swap one battery over 10 weeks [3] over/in 10 weeks [4] to solve one complete DES (10 weeks)

First, we discuss the average battery levels for the three instances and different parameter settings. For a minimum tour length of $v_{min} = 16$, the results corresponding to Amsterdam differ from The Hague and Utrecht. Independent of b_{max}, this instance shows lower battery levels. Further, we observe high values for the average time spent on swapping one battery. These results suggest that it is more costly to swap one battery due to the travel costs. To compensate for these costs, the collected prize must be larger to construct a profitable tour. Therefore, batteries of lower levels are observed compared to the other instances. In addition, fewer tours are constructed in Amsterdam relative to the instance size due to these higher travel costs.

Next, we evaluate the effect of parameter v_{min}. Each instance shows that the number of tours, and average time spent to swap one battery decrease if the minimum tour length increases. This result emphasizes the efficiency of visiting more bicycles in one tour, as less time is spent travelling up and down to the depot. Although efficiency increases with regard to the travel time, we observe that, mainly in Amsterdam, the number of (near) empty bicycles is related to the minimum tour length. Due to the construction of fewer tours, it is more likely that a battery level drops below 30% before it is swapped.

As discussed, the average travel time to swap one battery is related to the length of the constructed tours and depends on the spread of bicycles over the city. In The Hague, we observe smaller values, consistent with the smaller distances between the bicycles. In addition to the minimum tour length, the participation level b_{max} is also related. As we allow more bicycles to be visited if we increase the b_{max}, the total travel time increases and more tours are constructed. Further, the average travel times per visited bicycle decreased. Therefore, we observe that v_{min} and b_{max} are both negatively related to the average travel time spent to swap one battery.

As bicycles cannot be rented with a battery level below 30%, we want (near) empty batteries to prevent from occurring. The results show that b_{max} is related to these performance indicators, and this parameter has to be set carefully. Although increasing this parameter leads to fewer empty batteries, we have to take the operational costs into account corresponding to the number of tours and the total travel time. Both the number of tours and the total travel time increase if we allow more bicycles to be visited by increasing b_{max}. Further, the minimum tour length v_{min} is negatively related to the total travel time. As a result of large tours, the total time spent travelling from the depot to the first nodes, and from the last nodes to the depot decreases.

Lastly, the computational times are positively related to the instance size and the values of v_{min} and b_{max}. Despite the increase, the LS method is solves all instances within 4 seconds for each parameter setting.

Table 6.4: Overview of the LS results with good weather conditions averaged over 100 complete DES runs. It includes the average battery levels of all rental available bicycles, the average time spent to swap one battery, the average number of batteries replaced per tour, the number of tours, the number of times a battery level crosses the threshold b_{low}, and the number of times a bicycle is empty

Min. Tour Length (v_{min})	**10**				**13**				**16**			
Max. Battery Level (b_{max})	**40**	**50**	**60**	**70**	**40**	**50**	**60**	**70**	**40**	**50**	**60**	**70**
City: Amsterdam												
Avg. Battery (%)[1]	66.9	74.1	78.1	**80.8**	65.8	72.8	76.8	79.6	63.0	68.9	72.2	75.3
Avg. Trav. time (min)[2]	11.7	10.9	9.8	8.9	10.9	10.2	9.4	8.6	10.1	9.7	9.2	**8.6**
Avg. Visits per tour[3]	11.6	11.7	11.9	12.4	14.5	14.6	14.6	14.7	**16.0**	**16.0**	**16.0**	**16.0**
Tot. Trav. time (h)[3]	219	241	252	259	199	219	233	244	**171**	191	205	218
Nr. Tours[3]	96	113	129	141	76	89	102	116	**63**	74	84	95
Nr. Below 30%[3]	459	243	168	**139**	507	278	197	163	645	427	339	280
Nr. Empty 0%[3]	23	17	13	**11**	26	17	14	12	30	22	19	17
Comp. time (sec)[4]	**0.6**	0.8	1.3	2.4	0.8	1.0	1.1	2.6	1.5	2.0	2.2	3.2
City: The Hague												
Avg. Battery (%)[1]	66.0	73.8	78.1	**81.4**	65.0	72.7	77.3	80.7	63.9	71.3	76.0	79.6
Avg. Trav. time (min)[2]	9.6	9.2	8.6	7.9	8.7	8.5	8.0	7.5	8.2	8.0	7.7	**7.2**
Avg. Visits per tour[3]	10.9	11.1	11.3	11.8	14.0	14.1	14.2	14.3	**16.0**	**16.0**	**16.0**	**16.0**
Tot. Trav. time (h)[3]	113	130	141	150	101	117	129	140	**92**	107	121	132
Nr. Tours[3]	65	76	87	97	50	59	69	79	**42**	50	59	68
Nr. Below 30%[3]	303	153	97	**74**	326	171	111	83	360	198	129	94
Nr. Empty 0%[3]	15	10	8	**6**	16	11	8	7	17	12	9	7
Comp. time (sec)[4]	**0.3**	0.4	0.6	0.9	0.5	0.5	0.7	1.1	0.5	0.6	0.8	1.4
City: Utrecht												
Avg. Battery (%)[1]	66.7	74.3	78.4	**81.0**	65.8	73.3	77.6	80.5	64.7	72.1	76.5	79.6
Avg. Trav. time (min)[2]	10.9	10.7	9.8	8.9	9.8	9.6	9.1	8.5	9.2	9.1	8.7	**8.3**
Avg. Visits per tour[3]	11.2	11.4	11.8	12.7	14.3	14.4	14.5	14.7	**16.0**	**16.0**	**16.0**	**16.0**
Tot. Trav. time (h)[3]	158	184	196	202	139	164	179	191	**127**	150	168	180
Nr. Tours[3]	77	91	102	107	60	71	82	92	**52**	62	72	82
Nr. Below 30%[3]	347	171	108	**85**	376	193	121	93	418	219	141	104
Nr. Empty 0%[3]	19	12	9	**7**	20	12	9	8	20	14	10	8
Comp. time (sec)[4]	**0.4**	0.5	0.7	1.2	0.5	0.6	0.9	1.4	0.6	0.7	1.0	1.7

[1] of all rental available bicycles over 10 weeks [2] average time to swap one battery over 10 weeks [3] over/in 10 weeks [4] to solve one complete DES (10 weeks)

The good weather conditions come with more rental periods during our time horizon of 10 weeks. In addition, the travelled distance and duration per rental increase. As a result, we observe a change in the number of constructed tours in comparison to the results in case of bad weather conditions. For each instance, we observe for identical parameter settings that the number of constructed tours increases by 37% on average. The total travel time of the swapper increases by 44%, implying that the average tour duration increases.

Next, we evaluate the average number of bicycles visited per tour. As $v_{\text{min}} = 16$ corresponds to the maximum tour length, all tours are of length 16. Therefore, we consider the results corresponding to the minimum tour length of $v_{\text{min}} = 10$ and $v_{\text{min}} = 13$. We observe for each instance and parameter settings that slightly more batteries are swapped per tour in times of good weather conditions. In addition to this increase, we observe that less time is spent to swap one battery. These results emphasize that the constructed tours are more efficient.

Despite the increase in the number of constructed tours, we observe that the number of bicycles with a battery level below $b_{\text{low}} = 30\%$ and empty batteries increase. As the travelled distances of customers increase, higher battery levels are required to meet the demand. Therefore, in times of good weather conditions, we have to swap more often to reach a high level of customer satisfaction.

Lastly, the computational times increase at most 1 second for all instances and parameter settings although more tours are constructed.

7 Conclusion

The main goal of this report is to develop a tour construction method for the battery swapper of shared e-cargo bicycles. The bicycles are located at fixed locations and batteries must be swapped between two rental periods. The aim is to develop a tour construction method that leads to high battery levels and low operational costs.

Two tour construction methods are applied to three test instances, each with a different instance size and spread of bicycles over the city. First, we formulate our problem as a PTP with additional capacity constraints. The proposed exact mathematical model was not able to solve the problem for all parameter settings due to the time consuming subtour elimination constraints. The results showed high computational times in contrast to the second method, a heuristic approach. The LS resulted in a solution for each instance and parameter setting, and based on this approach we evaluated the effect of different strategies. We varied over different minimum tour lengths and battery levels that were allowed to be included in the tour. It was concluded that larger tour lengths lead to a more efficient swapping strategy in which operational costs are kept to a minimum. The average time spent on swapping one battery decreased significantly. We observe a trade-off between the number of constructed tours and the average battery levels. A larger minimum tour length leads to less constructed tours and to lower average battery levels. Lastly, the maximum battery level which we allow to swap affects the average battery levels. We observe a non linear relation as allowing to swap high power batteries do not increase the average battery level over time a lot. In our study, we conclude that a tour visiting 16 bicycles and a maximum battery level allowed to be swapped of 60% correspond to the best strategy.

To conclude, the heuristic approach constructs tours within seconds and the long term performance of a strategy depends on the tour length and batteries allowed to swap.

7.1. Future Research

In this chapter, remarks on this study are discussed and recommendations for the future development of the tour construction method are given. As discussed in the previous chapters, the DES in our study includes two events. First, we give some considerations on improving the tour construction methods. Then, the simulation of rental periods is considered.

As discussed, the demand for rentals differs over time and depends on the weather conditions. During the construction of a tour, we do not correct for this varying demand. The decision is based on the current battery levels and travel costs but does not include a forecasting component. From our data analysis, we observe seasonality and have insight into the situations corresponding to high demands. Therefore, one could anticipate future rentals. For example, we observed that rental periods often occur on weekends. Based on this information, we want the battery levels to be as high as possible at the start of the weekend to prevent missing out on customers. For that reason, it can be beneficial to swap more batteries than in periods of low demand. By adding a forecasting component, one could anticipate future demands.

Further, the prizes and costs of the tour are independent of the instance in this study. Our evaluation shows that the results of each instance with equal parameters differ. For example, fewer tours are constructed in Amsterdam relative to The Hague and Utrecht due to the average distances between bicycles. In general, the travel costs are higher, forcing the method to collect more prizes corresponding to low power batteries. To obtain similar results, prizes can be scaled to the distances of the considered instance.

Ultimately, we want to give some recommendations for the simulation of new rental periods. In this study, we assume that customers do not have range anxiety, and do not anticipate the remaining battery. Therefore, we observe a relatively high number of empty batteries. However, if customers rent a bicycle they adjust their trip to the battery level. If the remaining battery level does not satisfy the needs for their trip, they will decide not to rent a bicycle, or rent another bicycle. The quality of our evaluation by means of a DES can be improved by adding this component to the rental simulation.

A

Data Filters

Table A.1: Filters applied to the Bookings dataset

Property	Minimum value	Maximum value
Travelled distance (km)	0.5	50
Duration (min)	10	960
Start time	6am	22pm

B

Travel Distances

Table B.1: Travel Distances

City	Avg. distance (m)	Min. distance (m)	Max. distance (m)
Amsterdam	4111	99	17709
The Hague	2950	44	9227
Utrecht	3441	25	9340

C

Parameters Probability Distributions

Table C.1: Mean Duration ($\mu_{\log}^{\text{dur}}$) values in case of bad weather conditions

	[6am, 10am]	[10am, 2pm]	[2pm, 6pm]	[6pm, 10pm]
Monday	8.260810	8.435077	8.059138	8.016994
Tuesday	8.264027	8.398587	8.071291	8.036988
Wednesday	8.405646	8.543719	8.182139	8.16782
Thursday	8.278570	8.397970	8.080584	8.136701
Friday	8.442834	8.454481	8.269826	8.166757
Saturday	8.814560	8.780043	8.44628	8.142291
Sunday	8.874101	8.791601	8.413546	8.13105

Table C.2: Mean Duration ($\mu_{\log}^{\text{dur}}$) values in case of good weather conditions

	[6am, 10am]	[10am, 2pm]	[2pm, 6pm]	[6pm, 10pm]
Monday	8.584947	8.630505	8.182656	8.101798
Tuesday	8.483255	8.584724	8.223148	8.133498
Wednesday	8.657630	8.745326	8.346959	8.310820
Thursday	8.613593	8.628987	8.234450	8.235183
Friday	8.699351	8.677748	8.462096	8.247460
Saturday	9.040236	8.971133	8.586354	8.172179
Sunday	9.184582	9.045665	8.521736	7.921578

Table C.3: Mean Distance ($\mu_{\log}^{\text{dis}}$) values in case of bad weather conditions

	[6am, 10am]	[10am, 2pm]	[2pm, 6pm]	[6pm, 10pm]
Monday	1.77577	1.84172	1.60023	1.65759
Tuesday	1.81973	1.80164	1.57141	1.63661
Wednesday	1.76354	1.81882	1.67226	1.60702
Thursday	1.76294	1.83090	1.58666	1.66574
Friday	1.82619	1.80208	1.71527	1.65347
Saturday	1.95201	2.01004	1.72890	1.72359
Sunday	2.02195	2.07315	1.81362	1.61273

Table C.4: Mean Distance ($\mu_{\log}^{\mathrm{dis}}$) values in case of good weather conditions

	[6am, 10am]	[10am, 2pm]	[2pm, 6pm]	[6pm, 10pm]
Monday	2.00913	2.02360	1.71631	1.68702
Tuesday	1.93062	1.96491	1.69957	1.71665
Wednesday	1.96182	2.01023	1.75357	1.8177588
Thursday	1.97626	1.96920	1.65511	1.76981
Friday	2.00687	1.93408	1.77862	1.71911
Saturday	2.16209	2.16784	1.88342	1.75311
Sunday	2.32874	2.24900	1.89557	1.61836

Table C.5: Standard deviation Duration ($\sigma_{\log}^{\mathrm{dur}}$) values in case of bad weather conditions

	[6am, 10am]	[10am, 2pm]	[2pm, 6pm]	[6pm, 10pm]
Monday	0.8777186	0.7309603	0.6519099	0.7115335
Tuesday	0.8814643	0.7436526	0.7043419	0.8525847
Wednesday	0.9154163	0.7466301	0.6856938	0.797513
Thursday	0.9219994	0.7792053	0.7108434	0.7686418
Friday	0.8960269	0.7672247	0.7555489	0.8708647
Saturday	0.6931357	0.6990294	0.7145455	0.8200183
Sunday	0.7113786	0.6740978	0.6412977	0.7060338

Table C.6: Standard deviation Duration ($\sigma_{\log}^{\mathrm{dur}}$) values in case of good weather conditions

	[6am, 10am]	[10am, 2pm]	[2pm, 6pm]	[6pm, 10pm]
Monday	0.961026	0.7441905	0.6974019	0.7536564
Tuesday	0.9475594	0.7653082	0.7471943	0.7856797
Wednesday	0.9315624	0.7306246	0.7406049	0.7435374
Thursday	0.9549705	0.7762999	0.7548935	0.6995032
Friday	0.9079749	0.7748342	0.7531534	0.8217652
Saturday	0.7230317	0.7087304	0.7664482	0.8241688
Sunday	0.7008612	0.6576641	0.6831707	0.7351085

Table C.7: Standard deviation Distance ($\sigma_{\log}^{\mathrm{dis}}$) values in case of bad weather conditions

	[6am, 10am]	[10am, 2pm]	[2pm, 6pm]	[6pm, 10pm]
Monday	0.8064631	0.768421	0.7139459	0.7922288
Tuesday	0.7118459	0.7527316	0.7486595	0.8924656
Wednesday	0.7950033	0.7323659	0.7358424	0.8877815
Thursday	0.8095153	0.7524787	0.7459038	0.8634252
Friday	0.7704208	0.7491417	0.7018804	0.8311908
Saturday	0.7635206	0.7588446	0.7615624	0.8000778
Sunday	0.7707023	0.7499099	0.752386	0.8614943

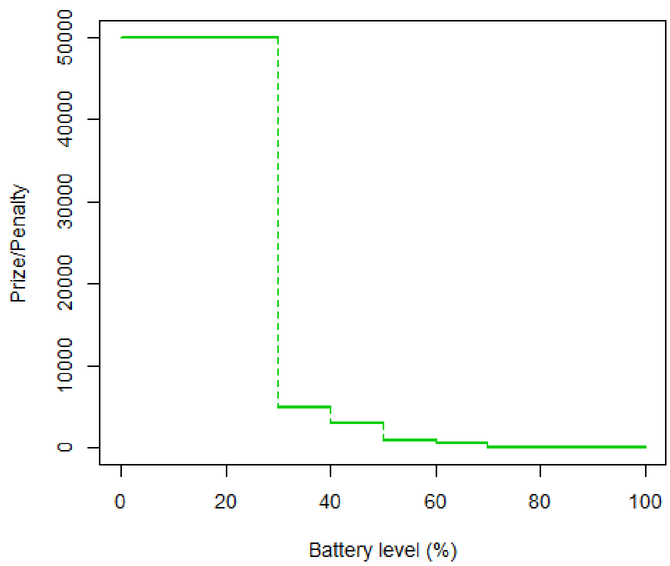
Table C.8: Standard deviation Distance ($\sigma_{\log}^{\mathrm{dis}}$) values in case of good weather conditions

	[6am, 10am]	[10am, 2pm]	[2pm, 6pm]	[6pm, 10pm]
Monday	0.8602686	0.7530236	0.7256035	0.8069414
Tuesday	0.7926841	0.7573008	0.7293453	0.829873
Wednesday	0.8028486	0.6980331	0.7294053	0.8009468
Thursday	0.8034361	0.7523778	0.7631304	0.7069137
Friday	0.7962065	0.738105	0.7098402	0.8645619
Saturday	0.7295599	0.7303775	0.7489465	0.8147839
Sunday	0.7397632	0.7399064	0.751997	0.8028605

D
Prize/Penalty Function

$$
f_r^t(b_i^t) = f_\pi^t(b_i^t) = \begin{cases} 0, & \text{if} \quad 70 \leq b_i^t \leq 100 \\ 500, & \text{if} \quad 60 \leq b_i^t < 70 \\ 1000, & \text{if} \quad 50 \leq b_i^t < 60 \\ 3000, & \text{if} \quad 40 \leq b_i^t < 50 \\ 5000, & \text{if} \quad 30 \leq b_i^t < 40 \\ 50000, & \text{if} \quad 0 \leq b_i^t < 30 \end{cases} \tag{D.1}
$$

Figure D.1: Function of prizes/penalties dependent on battery level



Bibliography

[1] Ross Anderson et al. “Finding long chains in kidney exchange using the traveling salesman problem”. In: *Proceedings of the National Academy of Sciences* 112.3 (2015), pp. 663–668.

[2] Jean-François Bérubé, Michel Gendreau, and Jean-Yves Potvin. “An exact ϵ-constraint method for bi-objective combinatorial optimization problems: Application to the Traveling Salesman Problem with Profits”. In: *European journal of operational research* 194.1 (2009), pp. 39–50.

[3] Daniel Bienstock et al. “A note on the prize collecting traveling salesman problem”. In: *Mathematical programming* 59.1 (1993), pp. 413–420.

[4] Fatih Birol. “World energy outlook 2006”. In: *International Energy Agency* (2009).

[5] Georg Brandstätter, Michael Kahr, and Markus Leitner. “Determining optimal locations for charging stations of electric car-sharing systems under stochastic demand”. In: *Transportation Research Part B: Methodological* 104 (2017), pp. 17–35.

[6] Georg Brandstätter, Michael Kahr, and Markus Leitner. “Determining optimal locations for charging stations of electric car-sharing systems under stochastic demand”. In: *Transportation Research Part B: Methodological* 104 (2017), pp. 17–35.

[7] Andrew Campbell et al. “Factors influencing the choice of shared bicycles and shared electric bikes in Beijing”. In: *Transportation research part C: emerging technologies* 67 (2016), pp. 399–414.

[8] Lee Chapman. “Transport and climate change: a review”. In: *Journal of transport geography* 15.5 (2007), pp. 354–367.

[9] Antonio Augusto Chaves et al. “Metaheuristicas hibridas para resolução do problema do caixeiro viajante com coleta de prêmios”. In: *Production* 17 (2007), pp. 263–272.

[10] Donna Chen, Kara M Kockelman, and Josiah P Hanna. “Operations of a shared, autonomous, electric vehicle fleet: Implications of vehicle & charging infrastructure decisions”. In: *Transportation Research Part A: Policy and Practice* 94 (2016), pp. 243–254.

[11] Mauro Dell’Amico, Francesco Maffioli, and Anna Sciomachen. “A lagrangian heuristic for the prize collectingtravelling salesman problem”. In: *Annals of Operations Research* 81 (1998), pp. 289–306.

[12] Mauro Dell’Amico, Francesco Maffioli, and Peter Värbrand. “On prize-collecting tours and the asymmetric travelling salesman problem”. In: *International Transactions in Operational Research* 2.3 (1995), pp. 297–308.

[13] Matthias Ehrgott. “Approximation algorithms for combinatorial multicriteria optimization problems”. In: *International Transactions in Operational Research* 7.1 (2000), pp. 5–31.

[14] Matthias Ehrgott and Xavier Gandibleux. “Multiobjective combinatorial optimization-theory, methodology, and applications”. In: *Multiple criteria optimization: State of the art annotated bibliographic surveys*. Springer, 2003, pp. 369–444.

[15] Ahmed El Fassi, Anjali Awasthi, and Marco Viviani. "Evaluation of carsharing network's growth strategies through discrete event simulation". In: *Expert Systems with Applications* 39.8 (2012), pp. 6692–6705.

[16] Dominique Feillet, Pierre Dejax, and Michel Gendreau. "Traveling salesman problems with profits". In: *Transportation science* 39.2 (2005), pp. 188–205.

[17] Francesco Ferrero et al. "Car-sharing services: An annotated review". In: *Sustainable Cities and Society* 37 (2018), pp. 501–518.

[18] Matteo Fischetti, Juan José Salazar González, and Paolo Toth. "Solving the Orienteering Problem through Branch-and-Cut". In: *INFORMS Journal on Computing* 10.2 (1998), pp. 133–148.

[19] Michel Gendreau, Gilbert Laporte, and Frederic Semet. "A branch-and-cut algorithm for the undirected selective traveling salesman problem". In: *Networks: An International Journal* 32.4 (1998), pp. 263–273.

[20] Leonardo Gomes, Viviane Diniz, and Carlos Martinhon. "An Hybrid GRASP+VND Metaheuristic for the Prize-Collecting Traveling Salesman Problem". In: *XXXII Simpósio Brasileiro de Pesquisa Operacional* (2000), pp. 1657–1665.

[21] Keld Helsgaun. "General k-opt submoves for the Lin–Kernighan TSP heuristic". In: *Mathematical Programming Computation* 1.2 (2009), pp. 119–163.

[22] Hipólito Hernández-Pérez and Juan-José Salazar-González. "A branch-and-cut algorithm for a traveling salesman problem with pickup and delivery". In: *Discrete Applied Mathematics* 145.1 (2004), pp. 126–139.

[23] Riccardo Iacobucci, Benjamin McLellan, and Tetsuo Tezuka. "Optimization of shared autonomous electric vehicles operations with charge scheduling and vehicle-to-grid". In: *Transportation Research Part C: Emerging Technologies* 100 (2019), pp. 34–52.

[24] Riccardo Iacobucci, Benjamin McLellan, and Tetsuo Tezuka. "Optimization of shared autonomous electric vehicles operations with charge scheduling and vehicle-to-grid". In: *Transportation Research Part C: Emerging Technologies* 100 (2019), pp. 34–52.

[25] Stefan Illgen and Michael Höck. "Electric vehicles in car sharing networks–Challenges and simulation model analysis". In: *Transportation Research Part D: Transport and Environment* 63 (2018), pp. 377–387.

[26] Shuguang Ji et al. "Electric bike sharing: simulation of user demand and system availability". In: *Journal of Cleaner Production* 85 (2014), pp. 250–257.

[27] Qi Kang et al. "Centralized charging strategy and scheduling algorithm for electric vehicles under a battery swapping scenario". In: *IEEE Transactions on Intelligent Transportation Systems* 17.3 (2015), pp. 659–669.

[28] Richard M Karp. "Reducibility among combinatorial problems". In: *Complexity of computer computations*. Springer, 1972, pp. 85–103.

[29] Carl Peter Keller. "Multiobjective routing through space and time: The MVP and TDVP problems". In: (1985).

[30] Carl Peter Keller and Michael Frank Goodchild. "The multiobjective vending problem: a generalization of the travelling salesman problem". In: *Environment and Planning B: Planning and Design* 15.4 (1988), pp. 447–460.

[31] Brenda Koerner and Jeffrey Klopatek. "Anthropogenic and natural CO2 emission sources in an arid urban environment". In: *Environmental Pollution* 116 (2002), S45–S51.

[32] Ho-Yin Mak, Ying Rong, and Zuo-Jun Max Shen. "Infrastructure planning for electric vehicles with battery swapping". In: *Management science* 59.7 (2013), pp. 1557–1575.

[33] John Morelli. "Environmental sustainability: A definition for environmental professionals". In: *Journal of environmental sustainability* 1.1 (2011), p. 2.

[34] Gur Mosheiov. "The travelling salesman problem with pick-up and delivery". In: *European Journal of Operational Research* 79.2 (1994), pp. 299–310.

[35] Jeremy Neubauer and Eric Wood. "The impact of range anxiety and home, workplace, and public charging infrastructure on simulated battery electric vehicle lifetime utility". In: *Journal of power sources* 257 (2014), pp. 12–20.

[36] Jesus Osorio, Chao Lei, and Yanfeng Ouyang. "Optimal rebalancing and on-board charging of shared electric scooters". In: *Transportation Research Part B: Methodological* 147 (2021), pp. 197–219. ISSN: 0191-2615.

[37] Luis Paquete, Marco Chiarandini, and Thomas Stützle. "Pareto local optimum sets in the biobjective traveling salesman problem: An experimental study". In: *Metaheuristics for multiobjective optimisation*. Springer, 2004, pp. 177–199.

[38] Odivaney Pedro, Rodney Saldanha, and Ricardo Camargo. "A tabu search approach for the prize collecting traveling salesman problem". In: *Electronic Notes in Discrete Mathematics* 41 (2013), pp. 261–268.

[39] Nadine Rauh, Thomas Franke, and Josef F Krems. "Understanding the impact of electric vehicle driving experience on range anxiety". In: *Human factors* 57.1 (2015), pp. 177–187.

[40] Gerhard Reinelt. "TSPLIB—A traveling salesman problem library". In: *ORSA journal on computing* 3.4 (1991), pp. 376–384.

[41] Julia Jean Robinson. "On the Hamiltonian Game (A Traveling Salesman Problem)". In: 1949.

[42] Richard Schmalensee, Thomas M Stoker, and Ruth A Judson. "World carbon dioxide emissions: 1950–2050". In: *Review of Economics and Statistics* 80.1 (1998), pp. 15–27.

[43] Martin Sklar. "Fonctions de repartition a n dimensions et leurs marges". In: 1959.

[44] Daniel Sperling. *Three revolutions: Steering automated, shared, and electric vehicles to a better future*. Island Press, 2018.

[45] Albert Sweetser. "A comparison of system dynamics (SD) and discrete event simulation (DES)". In: *17th International Conference of the System Dynamics Society*. 1999, pp. 20–23.

[46] Antuela Tako and Stewart Robinson. "The application of discrete event simulation and system dynamics in the logistics and supply chain context". In: *Decision support systems* 52.4 (2012), pp. 802–815.

[47] David Van Veldhuizen, Gary B Lamont, et al. "Evolutionary computation and convergence to a pareto front". In: *Late breaking papers at the genetic programming 1998 conference*. Citeseer. 1998, pp. 221–228.

[48] Yu-Jun Zheng et al. "Evolutionary Algorithm for Vehicle Routing for Shared E-Bicycle Battery Delivery and Pickup with Recharging". In: *Available at SSRN 4045818* (2022).

[49] Yu Zheng et al. "Electric vehicle battery charging/swap stations in distribution systems: comparison study and optimal planning". In: *IEEE transactions on Power Systems* 29.1 (2013), pp. 221–229.

[50] Rui Zhu et al. "Understanding spatio-temporal heterogeneity of bike-sharing and scooter-sharing mobility". In: *Computers, Environment and Urban Systems* 81 (2020), p. 101483.

DISTRICON
a company of Royal HaskoningDHV

ERASMUS UNIVERSITEIT ROTTERDAM

cargoroo