

# Improving the travel compensation plan of Betsy

Bachelor thesis Industrial Engineering and Management

Justin Veldhuis-(2378612)



## UNIVERSITY OF TWENTE.

Behavioural Management and Social Sciences Faculty

Industrial Engineering & Management Department

University of Twente, Enschede, The Netherlands

Supervisors:

Dr. Amin Asadi (1<sup>st</sup> supervisor)

Dr. Alessio Trivella (2<sup>nd</sup> supervisor)

Susanne van Doorn (Betsy)

## Summary

The issue investigated in this thesis was brought to me by home-care company Betsy. Betsy indicated that the money they receive from insurers is decreasing, which is why their large share of travel costs is a problem. Their operations consist of applying compression stockings to elderly and injured people, as well as some small household tasks. There is a set of morning patients, which are less frequent and take 15 minutes to help, and there are evening patients, which occur more often and take 10 minutes to help. And, of course, there are clients that get service in both the morning and the evening. The amount of patients and travel distances also differ per city. This means that a busier city uses more morning and evening routes than a quieter one.

In the current system all employees of Betsy drive their routes by bike, which they own privately. This usage of bikes results in long travel times, which means more salary that needs to be paid as well as less customers that can be helped in a day. Furthermore, all appointment times are fixed which means that there is no wiggle room to shuffle schedules whatsoever. Consequently, employees have to drive a suboptimal route as they are forced to drive the route in customer order.

In this research the effect of two scenarios was evaluated. Firstly, the impact of adding cars and ebikes as vehicle options was measured. Secondly, the usefulness of larger wiggle room, or Scheduling interval, in combination with a scheduling algorithm was checked. There are five KPI's used to evaluate the results which are Average cost per customer, Punctuality, Efficiency, Average Late time per customer, and Average waiting time per customer.

Since the company operates in an uncertain and dynamic atmosphere with regards to customers and travel time a Monte Carlo Simulation was employed to measure the KPI's. This method repeats a single simulation iteration many times to reduce uncertainty in the results. Due to the dynamic nature of operations that Betsy faces this is a viable way to evaluate performance over time.

The simulation operates by generating a patient matrix, as well as a customer base with corresponding appointment times and assigned employee. The generation of these entities was done by using a random number generator along a distribution, of which parameters per city were gathered from data. When these are generated, it allows for making a schedule, which is then solved with the input variables (e.g. 1 car, 5 bikes) indicated at the start of the simulation.

For the investigation itself 4 cities were tested, namely Zutphen, Doetinchem, Weesp, and Cuijk. The simulation for each of these cities were run for 100 iterations, with 10 weeks per iteration. All combinations of vehicles for that city were tested, as well as scheduling intervals of 30, 45, and 60 minutes in combination with the scheduling algorithm.

These simulation results indicated that with the current scheduling interval of 15 minutes it is most beneficial to use 1 car, and fill the empty spaces with ebikes. This would result in an average decrease in cost per customer of €1.09, a punctuality increase of 33.5%-point, an efficiency increase of 8%-point, and a decrease in average waiting time per customer of around 26 minutes.

For a scheduling interval, 45 minutes is the most efficient one to use in combination with the scheduling algorithm, in the current situation where all operations are done by bike. This results in an average decrease in cost per customer of €1.50, a punctuality increase of 28.8%-point, an efficiency increase of 14.25%-point, and a decrease in average waiting time per customer of about 5 minutes.

## Contents

Summary .....	2
1. Problem introduction .....	6
1.1. Defining the problem .....	6
1.2. Problem cluster .....	7
1.3. Core problem .....	7
1.4. Norm and Reality .....	8
1.4.1. Current situation .....	8
1.4.2. Improved situation .....	9
1.5. Problem-solving approach .....	10
1.5.1. Identify constraints .....	10
1.5.2. Map & prepare available data .....	10
1.5.3. Construct model .....	10
1.5.4. Evaluate and Implement .....	10
1.6. Deliverables .....	11
1.7. Research questions .....	11
2. Theory .....	12
2.1. Definitions .....	12
2.1.1. Monte Carlo simulation .....	12
2.1.2. Scheduling interval .....	12
2.1.3. Vehicle routing problem .....	13
2.2. Methodological framework .....	14
2.2.1. Crisp-DM steps .....	14
2.3. Literature review .....	14
2.3.1. Home healthcare routing and scheduling .....	14
2.3.2. Interval scheduling .....	14
2.4. Model horizon .....	15
2.4.1. Generation of patients grid .....	15
2.4.2. Generation of the schedule .....	16
2.4.3. Scheduling .....	16
2.4.4. Simulation output .....	17
2.5. Systematic literature review .....	17
2.6. Research design .....	17
2.6.1. Research type .....	17
2.6.2. Subjects of Research .....	17
2.6.3. Key variables .....	17

2.6.4.	Data type.....	18
2.6.5.	Analysis type .....	18
2.6.6.	Limitations of research method.....	18
2.7.	Assessment of validity and reliability.....	18
2.7.1.	Validity .....	19
2.7.2.	Reliability.....	19
3.	Model Description.....	19
3.1.	Assumptions.....	19
3.2.	Adapted Greedy heuristic .....	20
3.2.1.	Model elaboration .....	22
3.3.	Solution Method .....	22
4.	Simulation description .....	24
4.1.	Input Data .....	24
4.2.	City Data.....	25
4.3.	Variables.....	25
4.4.	Generate Matrix.....	26
4.5.	Customer base .....	26
4.5.1.	Customer amount generation.....	27
4.5.2.	Customer base generation.....	28
4.5.3.	Route Division .....	29
4.5.4.	Slot Division.....	30
4.5.5.	Service time generation .....	31
4.6.	Schedule Generation.....	32
4.6.1.	Filling CustomerID's .....	32
4.7.	Results.....	37
4.8.	Limitations.....	38
5.	Simulation setup .....	38
5.1.	Vehicle adaption simulations .....	39
5.2.	Algorithm-scheduling simulations .....	39
5.3.	Parameter extraction .....	40
5.3.1.	Distance distribution.....	40
5.3.2.	Distance Asymmetry distribution .....	40
5.3.3.	Duration Distribution .....	41
5.3.4.	Demand Distribution & Change Distribution .....	41
5.3.5.	Slot Distribution .....	43
5.3.6.	Examined cities .....	44

6.	Results .....	44
6.1.	Vehicle adaptation results .....	45
6.2.	Algorithm-scheduling results .....	48
7.	Discussion and conclusion .....	50
7.1.	Discussion.....	50
7.2.	Conclusion.....	51
7.3.	Limitations and future work .....	52
8.	Appendix .....	52
	Appendix A: Systematic literature review .....	52
	Appendix B: Check on good research .....	55
	Appendix C: CrispDM cycle .....	55
	Appendix D: City distribution comparison .....	56
	Appendix D.1: Duration Distribution per city .....	56
	Appendix D.2: Distance Distribution per city.....	57
	Appendix D.3: Change distribution per city.....	57
	Appendix D.4: Nr. of Clients distribution per city .....	58
	Appendix E: Vehicle adaption simulation graphs .....	59
	Appendix E.1: Vehicles vs cost comparison .....	59
	Appendix E.2: Vehicles vs efficiency and punctuality comparison .....	60
	Appendix E.3: Vehicles vs waiting time and late time comparison .....	61
	Appendix E.4: Vehicles vs driving distance and non-service time comparison .....	62
	Appendix E.5: Vehicles vs Average cost share comparison .....	64
	Appendix F: Algorithm-scheduling simulation graphs .....	65
	Appendix F.1: Scheduling interval vs Efficiency and Punctuality comparison .....	65
	Appendix F.2: Scheduling interval vs Average waiting time and Late time comparison .....	67
	Appendix F.3: Scheduling interval vs Average cost comparison.....	67
	Appendix F.4: Scheduling interval vs Total distance comparison.....	68
	Appendix G: Full result tables .....	68
	Appendix G.1: Vehicle Adaptation simulation tables .....	68
	Appendix G.2: Algorithm-Scheduling simulation tables .....	69
9.	References .....	70

## 1. Problem introduction

### 1.1. Defining the problem

Betsy is a home care provider specialising in the application and undoing of compression stockings. Betsy provides care in 6 different provinces in the Netherlands with each province being split up into smaller regions. Each region serves customers weekly, with every morning care procedure lasting 15 minutes, and the afternoon/evening procedures taking up 10 minutes. These service times are determined by the insurance companies, as that is the time they will reimburse the customers for. Upon arrival the employees apply the compression stockings, sometimes do some household chores, and leave to the next customers. Most people in service depend on Betsy to start their day, or to go to bed. Customers are either elderly people that do not have the ability anymore to put on the stockings themselves or injured people that face the same issue. The individuals Betsy serves either call directly to Betsy themselves for an appointment or are ordered from Betsy by their residential care facility.

Over time, Betsy has expanded due to rising demand, and are now struggling with some obstacles regarding the travel time of their employees. Currently employees arrive late, inaccurate schedules are presented, and compensation for their services is in decline. Combining this with an everlasting employee shortage, it is clear that to maintain profitability and an ordered schedule, something needs to change. Their program within their ERP (Enterprise Resource planning) system only accounts for the distance an employee has to travel and the average speed of a vehicle. This regularly results in employees arriving too late, upsetting the schedule for the rest of the day, or way earlier, costing the company money. Furthermore, there are some factors of which Betsy is unsure of how to implement, such as appointment time windows (e.g. arrival time must be within 15 minutes of the appointed time) and the decision to acquire new vehicles.

This means that the current travel time compensation plan is lacking and I have therefore been employed to figure out what the efficient policies would be for Betsy to optimize their travel costs. Therefore an action problem can be defined as: How can Betsy improve the present inefficient travel compensation plan, to a more efficient one?

## 1.2. Problem cluster

### Problem Cluster Betsy's travel compensation plan

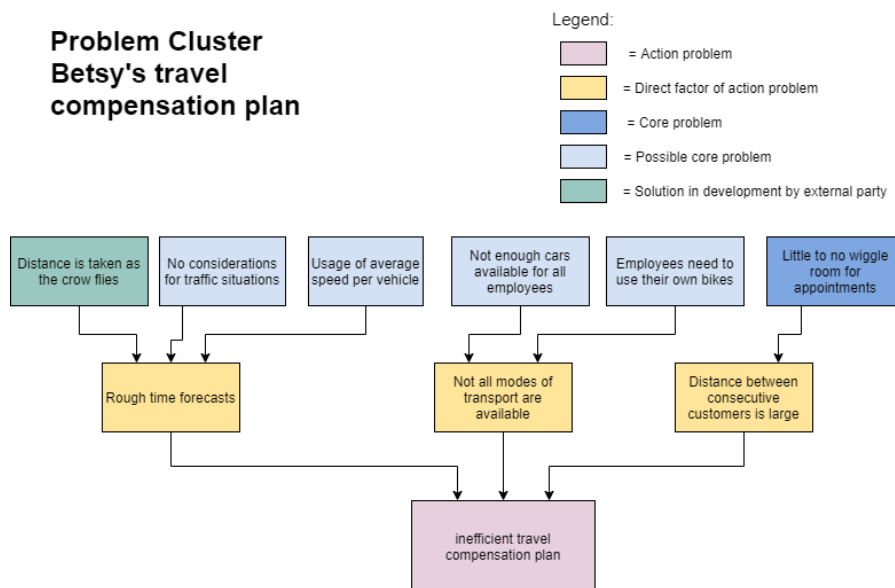


Figure 1: Problem Cluster of Betsy's travel plan

When looking at the problem cluster (Figure 1) many issues make up the action problem. Some of these are difficult to implement due to the ERP system Betsy uses, such as the lack of consideration of traffic and blind trust in the average speed per vehicle. This means that all variations that occur in real life are essentially ignored. The ERP system maintainer is already working to express the distance along the road instead of “as the crow flies”. This would allow the manager to express distances over a path that the employee travels on, in place of the current distance between two points. Two other problems, namely the available cars and own bike usage, are highly dependent on the length of the routes that an employee has to travel. If there are shorter routes, the effectiveness of buying extra cars or electrical bikes is diminished. Furthermore, these investments require capital, which is not always available to healthcare companies.

Moreover, a problem that needs to be researched, is the small wiggle room when making appointments. Currently Betsy promises to be at the customer within 15 minutes of the appointed time, which, with a service time of 10 to 15 minutes, means that there is no room for restructuring the order in which patients are served if one wants to be on time. This lack of flexibility means that employees need to travel all around town, as consecutive customers might not be close to each other, whereas customer reordering could result in a smaller total travel distance. Its resolution will be mostly free of charge as there will be no extra equipment or personnel necessary. Only an adaption of the current ERP system with an algorithm or heuristic is necessary to make this function on a daily basis. Additionally, the solution to this problem will likely produce a better scheduling and routing plan, which will make the research of other possible core problems more representative.

## 1.3. Core problem

One thing that was clarified by the manager is that some customers value their allotted time. If Betsy tells the customer they will be there at 09:15, some customers will start to call at 09:16. This means that the appointments need to be taken care of precisely by appointment time to adhere to their time arrangement. Otherwise, Betsy receives a lot of calls from unhappy customers, which increases personnel costs as someone needs to answer those calls and it also decreases customer satisfaction. This strategy is inefficient as situations can occur where neighbouring patients are not treated

consecutively, as they are not scheduled after each other. This could also result in gaps in the schedule if someone cancels their appointment.

Since customers can choose their own time slots it is not possible for Betsy to artificially create gaps by rescheduling patients. If a patient is rescheduled it must be for a longer time, as most elderly people seek consistency in their own schedules. This reduces the wiggle room for appointments even more. When they call Betsy, they can choose their time slot, or a neighbouring time slot if the one desired is filled already. This system disregards the place of residence, so people in the north of town can plan an appointment after someone that lives in the south, and if it is profitable to fit in the route. Expectedly, this tight schedule results in situations where employees are travelling unnecessarily large distances to clients.

Lastly, there can also be issues with employees having to wait, as there can be large gaps between customer appointment times. Here, the employee can face situations where she has served one customer but has to wait with service for the time agreed with the next one. This waiting time does not count as working time and is therefore not compensated by Betsy. This is then unpleasant for the employees as they experience gaps in their personal schedules that they do not get paid for.

#### 1.4. Norm and Reality

In every research, it is always necessary to look at the present and desired situation. I will be explaining this by using a grid of 3 customers (Figure 2), and a start/endpoint. Let each node be a destination on the map that the employee of Betsy can visit. The starting point, which is also the endpoint, is denoted by 0. Note that the customer numbers are in order of appointment time. So number 1 indicates that the destination is the first time appointed, 2 is the second time etc. The numbers with arrows denote the distance in kilometres between customers the indicated line connects. These numbers are assumed for the sake of simplicity and do not correspond with the node placement.

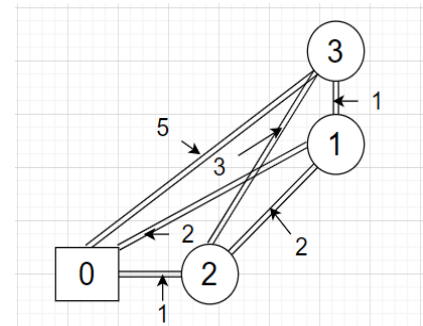


Figure 2: Grid with 3 customers in current operations

One thing that is always present is the uncertainty within the system. This uncertainty manifests itself mainly in the customer base. The amount of customers changes, but with that change a myriad of other factors change as well. Think about the distance between customers, the amount of customers per route or the appointment times at which customers are served. In terms of Figure 2 this can be seen as a node that is removed or added, the numbers on the lines that change, or a switch in numbering of the nodes.

##### 1.4.1. Current situation

As things stand, customers have to be treated in order of appointment time, which with the grid from Figure 2 would get a route depicted in Figure 3. From an optical perspective, this is not the optimal route as the route lines cross each other. Table 1 illustrates the distances from node x to the previous node (e.g. at customer 1: the distance from 0 to 1 equals 2). As is, Betsy also does not incorporate any travel time into their schedule which means that the delay due to travel time increases as the day progresses.

In this case all routes are also driven by bike which has an average speed of 12 kilometres per hour (Centraal Bureau voor de Statistiek, 2002). This done currently as it is a cheap solution since every employee uses their personal bike, which means Betsy does not have to do any investments. Table 1 also shows the travel time the employees would face when driving the route. Since Betsy does not



account for travel time, the total travel time in Table 2 would be amount of hours late the last customer is served.

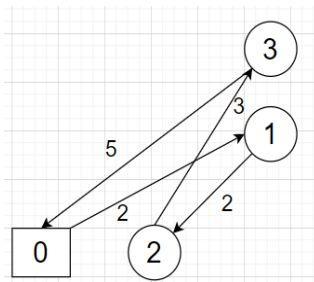


Figure 3: Current routing within the grid

Customer	Distance (km)	Travel time with bike (hours)
0	0	00:00:00
1	2	00:10:00
2	2	00:10:00
3	3	00:15:00
0	5	00:25:00
Total	12	01:00:00

Table 1: Distance with current routing

#### 1.4.2. Improved situation

In the future, the routes should look like the ones depicted in Figure 4. The route is driven in the shortest manner with no restriction in customer sequence. This allows for restructuring the order of customers so that in total a shorter route is driven. In figure 4 no overlapping lines are visible, and a shorter distance driven can be seen in Table 2. In Figure 4 there is a routing heuristic used that determines the next destination by selecting the shortest distance node that has not been served. For example, from node 0, in figure 1, the closest destination is node 1, at node 1 the closest unserved customer is node 2. This is also known as the Nearest Neighbour Heuristic. There are, however, many other ways of determining routes, and the aim is to have an improved one by the end of this research. Furthermore, the time-driven approach should also be included in the schedule so that there is an accurate depiction of the time at which an employee is at a client location. This would then reduce travel costs as well as reduce customer dissatisfaction by better forecasting employee arrivals.

Adding an extra vehicle such as an ebike or car would take down the total travel time and result in a higher chance of arriving on time. An ebike has an average speed of 25 kilometres per hour (Gazelle, 2022). In table 2 there is a clear difference to be seen in total travel time between the normal bike and the ebike. The ebike is much faster and would over larger distances be much more profitable. In the improved situation there would be a number of ebikes and cars acquired that would be most profitable for Betsy.

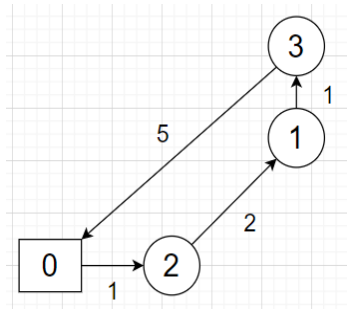


Figure 4: Nearest-neighbour heuristic route

Customer	Distance (km)	Travel time with bike (hours)	Travel time with ebike (hours)
0	0	00:00:00	00:00:00
2	1	00:05:00	00:02:24
1	2	00:10:00	00:04:48
3	1	00:05:00	00:02:24
0	5	00:25:00	00:12:00
Total	9	00:45:00	00:21:36

Table 2: Distance with nearest-neighbour route

### 1.5. Problem-solving approach

In order to find the solution to this problem, it is necessary to identify what steps are necessary to complete. The idea is to construct a simulation for experimentation as to represent the uncertainty in the daily operations of Betsy. This uncertainty is found in the customer base. From one week to the next the customer base changes, and with it the amount of customers, the distance between customers, and the slots they all occupy. In the simulation these should be represented according to statistics gathered from data.

#### 1.5.1. Identify constraints

First of all it is necessary to discuss with Betsy what has to be taken into account when thinking of solutions. Healthcare logistics is different from regular logistics as many human constraints are present, such as maximum waiting times. Moreover certain assumptions need to be made to correctly model the real-life situation. Examples of this are service times and available vehicles.

#### 1.5.2. Map & prepare available data

When the problem, and all its key features have been identified, it is time to seek out data that is necessary for modelling the problem. When the type of data is known, then we can accurately state what problems are feasible to solve, and what problems we have to neglect for now. When the data sets that are to be used are selected, they still need to be cleaned. This includes the removal of empty cells or formatting the data into the right unit.

#### 1.5.3. Construct model

This is the most important step, where the actual model to be used is constructed. For our model, we choose the approach of a Monte-Carlo simulation model which will be constructed in Excel with the use of VBA. This model should effectively represent the real life situation and should give a solution to the indicated problem.

#### 1.5.4. Evaluate and Implement

The last step is the evaluation, and if possible, implementation of the model. This concept is about the testing of the model to the real world to see if it accurately depicts the situation. Likewise, this accuracy is paramount to the conclusions drawn by using the model. If the model seems to be off, then it is necessary to go back to step 1 in Section 1.5.1. Otherwise, one can think of the possibilities of using the results of the model to make a business decision, or even how to use it in the future. When the simulation results are in, they should give Betsy an insight on what decisions to make in order to improve performance in the future. This is based on parameters that are currently relevant,

which means that the study should be done again once parameters have changed to see if conclusions are still relevant.

### 1.6. Deliverables

There will be a set of deliverables presented to the company when this thesis is finished, namely:

1. A Monte-Carlo simulation of the scheduling and vehicle-based operations
2. A set of possible decisions and their trade-offs
3. Recommendations for each decision and how to implement them
4. A report on their problem with corresponding research

### 1.7. Research questions

To resolve, either partially or fully, the core problem from Section 1.3 there needs to be an answer to certain knowledge questions. These knowledge questions are related to the core problem, or other parts of the problem cluster (Figure 1).

*Question 1: What scheduling interval is efficient?*

This first question is related to the current inability to switch the order in which clients are visited. The scheduling interval is the amount of time one allows to be late for the appointed time, so a scheduling interval of 30 minutes means that an employee can arrive at most half an hour after the appointed time. The general principle of such an interval would mean that the larger such an interval is, the more it allows for the restructuring of the customer ordering. Such a restructuring is helpful as it can be done so that the travel time, thus total worktime is decreased as explained in Section 1.4.2.

It would be ideal to remove all restrictions regarding time arrivals, but that would be very impractical for clients. Therefore there needs to be a proper balance between an increase in efficiency and the waiting times for the clients. In order to evaluate the schedule 5 key performance indicators have been selected. Efficiency, Punctuality, Average waiting time, Average Late time, and Average cost per customer will most accurately depict the factors the business thrives on. Note that efficiency refers to the amount of time the employee has spent providing service, versus the total amount of time worked. The answer to this research question would then give a time interval that would fulfil that balance.

*Question 2: What is the best solution method to use for the routing/scheduling?*

In order to create an efficient schedule there is a solution method necessary to compute an improved situation with a certain set of resources and constraints. For Betsy, this would mean that the solution method should pick the patient ordering where the least amount of time is wasted travelling.

There are many different routing solution methods possible, take for example the nearest neighbour heuristic used in Section 1.4.2 or a sweep algorithm. Each solution method is tailored for a specific set of requirements and conditions. It is, therefore, best to search for an algorithm or heuristic that fits the situation at Betsy. If a proper solution method is found, it can then be implemented in combination with the scheduling interval from question 1 to streamline logistics.

*Question 3: Is the purchase of new vehicles such as cars, scooters and electric bikes profitable?*

Due to Betsy only having 3 cars of their own, and employees needing to use their own transportation, the purchase of new vehicles might bring extra benefits. These benefits could include

faster travel for employees and smaller speed deviations. The downside is that there might not be enough benefits to outweigh the substantial up-front costs. It is, therefore, useful to make a cost-benefit analysis for the purchase of those vehicles, and on which occasions they should be used.

## 2. Theory

### 2.1. Definitions

Before we can quantify certain variables, it is necessary to elaborate more on their meaning and the surrounding constructs.

#### 2.1.1. Monte Carlo simulation

A simulation is a model that seeks to simplify and emulate a real-world system using mathematics (Carson & Maria, 1997). This model has a set of input variables that influence the output variables related to the system. Output variables then say something about the performance of a system. Simulation experimentation then involves changing the input variables to monitor the change in output variables, also known as a sensitivity analysis. This is useful for logistics as factors such as drivers or gas prices often influence how certain resources are divided. For Betsy specifically, the effect of available vehicles and interval schedule size are to be tested. Simulation experimentation then allows us to change these variables in a simulation and compare the differences in the result, to evaluate what input variables return the best output variables.

As explained in *Why the Monte Carlo method is so important today* (Kroese, Brereton, Taimre, & Botev, 2014), a Monte Carlo simulation is the production of random procedures or entities with a computer. The randomness of the Monte Carlo simulation is then inserted by its creator and often samples from a certain probability distribution. This would then solve problems that seem to have a single solution by statistics and the Law of Large Numbers. Monte Carlo simulations are then particularly useful when sampling, estimating, and improving. Furthermore, this technique makes it easier to convert real-life problems into a digitalized version by simplifying complex functions and allowing for randomization and experimentation.

This is where Monte-Carlo simulation, for instance, is the appropriate simulation method to use. Most other simulation methods either give a deterministic solution, or provide a set of solutions. These work on the assumption that there is a fixed amount of nodes travelled to that are static or interchangeable. In this situation, nodes are the customers that an employee needs to visit. For our case there is no specific set of fixed nodes, these are added or removed randomly. Furthermore, the characteristics of the nodes might change as well. For instance, customer *x* that is helped at 9 in the morning on one week can be transferred to be serviced at 10 the following week. These uncertainties need to be captured in order to provide solid policy advice for Betsy. Monte-Carlo simulation builds on those uncertainties by multiplying the result of single simulations. So a set of high-distance customers, for example, is possible to generate, but it will occur less frequently as the probability is low yet it will be incorporated into the simulation. In essence, this approach iterates a single simulation instance a large amount of times which makes the overall result more reliable and less susceptible to deviations or outliers.

#### 2.1.2. Scheduling interval

A scheduling interval is the time window in which the client is to be served. This term is derived from the interval scheduling problem, where there are a set of operators and tasks with a certain start and end time (Yim, Hong, Park, & Chung, 2022). In this problem, the idea is that each task has to be

finished after the start time and before the end time for as little cost. The time between the start- and end time is called the scheduling interval and can take on any time unit. In this research, we will tweak and experiment with the scheduling intervals times to see the effect on the cost.

### 2.1.3. Vehicle routing problem

A vehicle routing problem (VRP) can be defined mathematically as the most efficient model with a solution to a function containing the set of customers, the set of positions of those customers, and the corresponding cost of travelling between customers (Eksioglu, Volkan Vural, & Reisman, 2009). This method is appropriate for the problem at hand since the operations of Betsy consist of employees having to visit customers at home to provide service. So when there is a more optimal route that can be driven by an employee, travel costs will go down and the efficiency increases.

Suppose there exists a network:  $G = (V, A)$ , where  $V = \{0, 1 \dots n\}$  is the vector containing the number of customers, and  $A$  is the set of edges. Then  $c_{ij}$  is the cost of travelling from customer  $i$  to customer  $j$ , whose position lies in vector  $A$  - and numbers lie in vector  $V$ . The solution to the VRP is then the lowest sum of costs for which every customer is served. There are many varieties of VRP's that call for different solution approaches. In the research, a literature investigation was done to find the algorithm that solves our specific VRP variant.

The combination of the distances between customers is called the distance matrix (Parlaktuna, Sipahioglu, & Yazici, 2007). This matrix can either be symmetric so the distance  $C_{ij} = C_{ji}$  or asymmetric, where  $C_{ij} \neq C_{ji}$ , where in our case  $C_{ij}$  represents the cost of travelling from customer  $i$  to customer  $j$ . In our specific case the matrix is asymmetric. Furthermore, the travel times that are worked with are stochastic, as traffic obstructions such as red traffic lights or the speed at which the employee bikes are affecting the travel times randomly.

An exact way to solve the VRP variant we are working with is the Branch-and-cut method used for Integer Linear programming (Oyola, Arntzen, & Woodruff, 2017). This method allows for VRP optimization with stochastic travel times. Moreover, the Branch-and-cut makes it possible to solve for either the maximum probability of being on time at each customer, as well as the minimum time to complete the entire route. In order to execute this, an objective function is created that minimizes the expected value (e.g cost or travel time). The Branch-and-cut part arrives from combining the Branch-and-bound algorithm as well as the Cutting-plane method.

Branch-and bound searches for the next possible solution when in a current state. It explores the state space to seek solution options. In our problem that would equate to finding the next possible patients  $y$  for which all customers are served on time when one is currently at patient  $x$ . The Cutting-plane method ensures that there is a realistic, for example integer, solution by removing the fractional component of the variables. This ensures that an optimal solution is found, and the non-integer decision variables can then be rounded up or down into integers to retrieve the nearest possible optimal solution.

The solution that then appears from using the Branch-and-cut method should then be an optimal route without subtours. This result is then accompanied by a percentage, which represents the chance that employees will arrive on time at the customers. This would then mean that for each route Betsy drives an optimal route is given that maximizes the probability of being on time for all customers.

Using this exact method is accurate, but also takes a long time to do with many iterations. Therefore a heuristic was chosen to decrease computation time, but still retrieve accurate results. The entire workings of the model are elaborated in section 3.

## 2.2. Methodological framework

The issue Betsy faces can be evaluated with the use of their gathered data, therefore it will be wise to use the Crisp-DM methodology. According to IBM, Crisp-DM is ideal for data mining and other types of data modelling (IBM, 2021). This is then a basic description of its usage possibilities. But Crisp DM is also used to maximize or improve business efficiency by using predictive statistics gathered from data (Huber, Wiemer, Schneider, & Ihlenfeldt, 2018). The description given here is then exactly what this simulation is about, and should therefore be the methodological method that is used. Appendix D gives an overview of all the different steps present in the Crisp-DM methodology. Its steps are already used in Section 1.5.

### 2.2.1. Crisp-DM steps

First of all, there needs to be an idea of what the business wants. If we do not know what the objective is, there is no vision or goal for the research. Secondly, we need to have data understanding, which is to think about what data is useful and available to us. If there is an overlap between those, then we have data that will be used. Thirdly, the data needs to be prepared for usage. There is no point in using data that has missing columns or that is not converted into the right format. Then, a model needs to be made with the prepared data. A model is a simplified version of reality that can be used for analysis. Many tools can build models, but in this research, I will mainly use Excel. It is then important that the prepared data can be used in the model, or else the data needs to be prepared further. The next step is to evaluate the model for accuracy and usability. If either of these is unacceptable, then there needs to be a re-evaluation of the business understanding, and work from there to the evaluation step again. If the model is deemed acceptable, then it can be employed for use.

## 2.3. Literature review

With the above definitions known, it is possible to understand the literature written in this field. We will examine this literature further to compose a theoretical framework that fits our research.

### 2.3.1. Home healthcare routing and scheduling

Home healthcare routing and scheduling problems (HHCSP) can be described by a set of characteristics and generally have a similar set of constraints and objective function (Di Mascolo, Martinez, & Espinouse, August 2021). All HHCSP consists of several patients on a grid that need a home carer to provide their service. The specifics of an HHCSP may differ per problem. Some home care activities have varying service times, or some home care employees start work from home instead of the office. The difference between the two is that the HHCSP has a variety of non-financial constraints such as minimizing employee idle time or maximizing punctuality for patients. Many home-care providers then use scheduling intervals for client availability in their routing, adding time windows to the VRP. The definition of an HHCSP is very similar to that of a VRP (Cissé, et al., 2017), which makes literature on VRP's often applicable to HHCSP's.

### 2.3.2. Interval scheduling

The addition of time windows transforms the VRP formulation for the HHCSP somewhat (Di Mascolo, Martinez, & Espinouse, August 2021). Assume the mathematical description in Section 2.1.3 still holds. The time window for a customer  $i$  can be calculated by taking his appointment time as the lower bound for the arrival time. The upper bound then is found by adding the chosen scheduling interval to the appointment time. Denote these as  $a_i$  and  $b_i$  respectively, making the

time window for patients  $[a_i, b_i]$ . If, for example, a customer has an appointment at 18:30 with a scheduling interval of 30 minutes, then  $a_i = 18:30$  and  $b_i = 19:00$ . Additionally the time at which service for customer  $i$  starts is stated as  $s_i$ . Each home care provider has a number of employees, which will be denoted as a variable with index  $K$ . Lastly, there is a binary variable that needs to be 1 when employee  $k$  travelled between customer  $i$  and  $j$ , but 0 otherwise. Declare this binary variable as  $x_{ijk}$ .

The optimization function can then be described as  $\min \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ijk}$ , where  $k$  represents the number of employees and  $V$  the set of customers. This function has the basic constraints, such as the obligated service of all customers, and all variables must be nonnegative integers or binary. But this time window variant comes with a new interval schedule constraint:  $a_i \leq s_{ik} \leq b_i$ , which ensures that the employee  $k$  starts its service of customer  $i$  between the schedule interval. Additional constraints may apply depending on the HHCSP variant. Most other constraints such as maximum continuous worktime, or employee qualifications do not apply to our problem. The cause of this is that this type of home healthcare is classified as unskilled labour and is done by people with 0-hour flexible contracts. Results are that employees themselves determine when- and how long they work, with no differentiation in skill between them.

## 2.4. Model horizon

The model will have to simulate a week of operations within Betsy and repeat that multiple times. The interval of a week is chosen as most customers stay in service for at least a week, and generally stay in service for a period of weeks. The adapted interval then allows the simulation to account for customers leaving and joining accurately, while keeping the simulation times reasonable. This model will then consist of three steps. The first step is the generation of the patients grid. The second step is about the generation of the schedule. The last step is about solving the schedule according to a routing algorithm or heuristic. In order to start the simulation some parameters will need to be filled in such as the Interval schedule or the amount of employees.

### 2.4.1. Generation of patients grid

In this first task we needed to draw distributions from the data at hand. An entire overview of all distributions and their purpose is detailed later in Section 6. Betsy has a data set with clients for each city, which will be used to make a distance matrix. The distances of each customer to another customer in that city can be extracted from an API (Application Programming Interface). This API ensures that the distances are expressed in driving distance, instead of Euclidean distance.

After the distance matrix is filled, a histogram can be made. This histogram will have to be fitted along a probability distribution such as a Gaussian distribution (Toril & Chakrabarti, 1993) in order to get the smallest possible deviation. The parameters of this distribution are different for each city. Then, this probability distribution will be converted into a cumulative distribution function, which consecutively have inverted axes. The result is a function that gives a unique value for the distance at each probability insertion. The randomness of this probability can be expressed as a  $U(0,1)$  distribution, or  $Rnd()$  in Excel. This property can then be used to construct a random number generator along a distribution. So for each value of the  $U(0,1)$  distribution a unique number for the distance is computed based on the distributions gathered from data.

The abovementioned function can then be used to generate a patient distance matrix that is randomized along an existing distribution of a city for each week of scheduling. In figure 5 an example is portrayed of how the distance is computed between customers using the Distribution function and the  $Rnd()$  option. During the generation of the entire distance matrix, the starting



customers and ending customers are iterated until each possible combination of existing customers is filled with a generated distance. To clarify, this means that customer x has been the starting customer for all remaining customers (i.e. customers 1 to x-1 and customers x+1 to N), so this would be the distance from customer x to all other customers. Also, customer x needs to be the ending customer for all other customers, which would be the distance from all other customers to customer x.

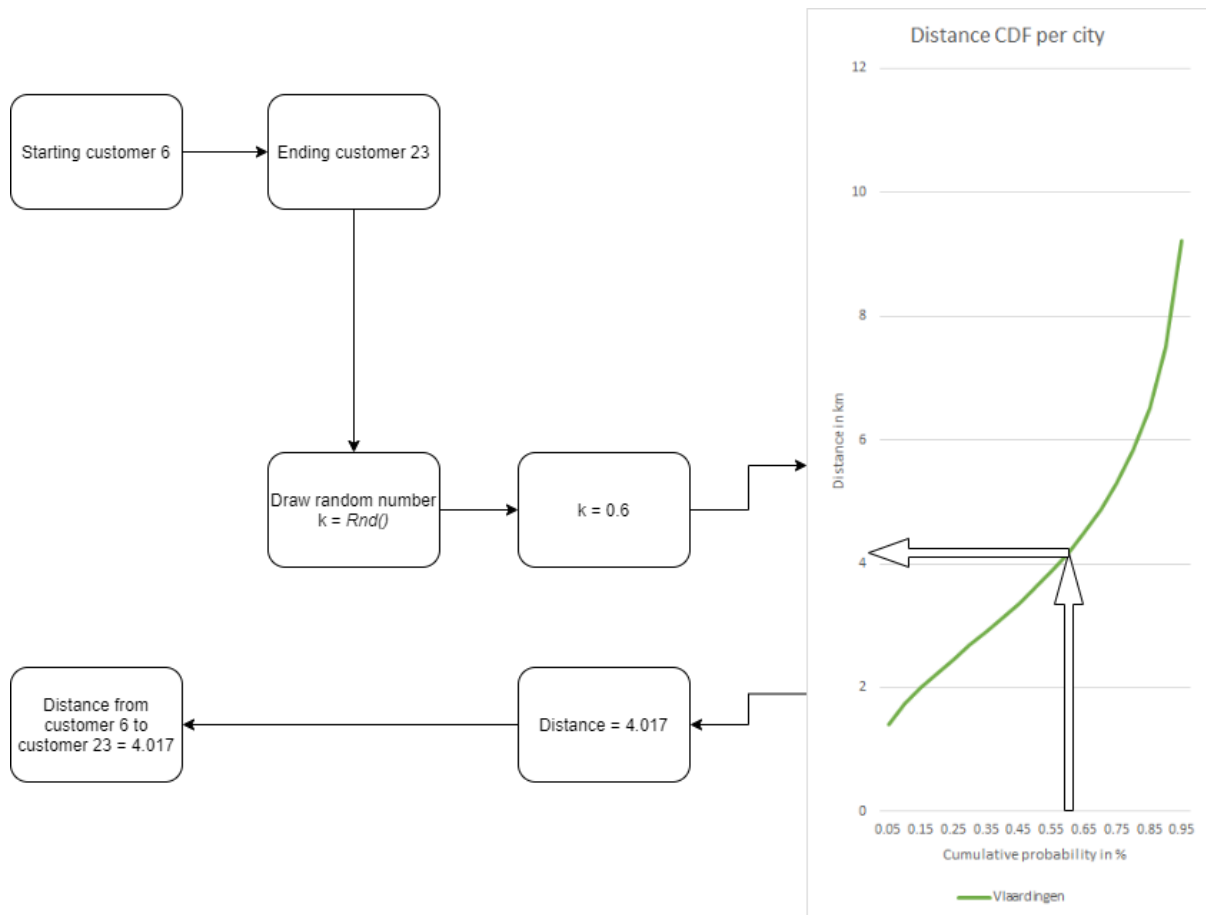


Figure 5: Example of distance matrix generation with the  $Rnd()$  function

#### 2.4.2. Generation of the schedule

After the patient distances are generated it is time to construct a schedule with those distances. Each care activity takes 15 minutes in the morning, and 10 minutes in the evening without any relevant deviation. Morning slots are from 7:00 to 12:00 and evening slots from 17:30 until 22:00. The slot times are the 15 minutes after each other in the morning, and 10 minutes after each other in the evening. This is so because Betsy does not account for travel time yet, and all customers could be scheduled right after each other without breaks.

#### 2.4.3. Scheduling

First of all, an empty schedule is generated with all possible times and slot numbers. This schedule is fixed as slots are not changeable. Think of this as 07:00 being slot number 1, and 07:15 being slot number 2 etc. in which the operational time is the same every day. This means that the slot numbers and times do not change, but only the customers occupying them. We need to iterate all the possible slot numbers and determine if there is a patient in that slot, if so then a random patient number is generated. If all slots are iterated we should obtain a full schedule with patient numbers that needs to be optimized. Solve the schedule



Each routing problem has a solution. Our task is to find the best solution of the generated schedule, and do that for all generations of the Monte-Carlo Simulation. First the patients appointment slot is extended by the scheduling interval chosen at the start. So this means that a scheduling interval of 30 minutes and an appointment at 09:30 would give the employee a window between 09:30 and 10:00 to start service. This means that there is space to swap customers and an employee still arrives within the appointed time window for both of them. Consecutively, there is more room to restructure the patient ordering so employees do not need to drive back and forth across town, but drive in a shorter route.

#### 2.4.4. Simulation output

The output of the Monte-Carlo simulation should be a set of graphs and key performance indicators based on which Betsy can make a decision. Most factors experimented on will have a diminishing marginal utility, which means that there should be a cut-off somewhere, but the model output should give a proper visualisation of the costs and benefits.

### 2.5. Systematic literature review

For the systematic literature review I have chosen the knowledge question *What is the best routing solution method to use?* In Appendix B, the whole steps are explained and illustrated. Appendix A then illustrates the thought process on finding the answer to the question. The answer to that knowledge question was to use an altered version of a Greedy Heuristic. This was so because the heuristic is relatively easy to program which is important for possible implementation, as the ERP system Betsy currently uses is quite sluggish. The Greedy heuristic also fits the constraints of the HHCRSP (Home Healthcare Routing and Scheduling Problem), and gives sufficient accurate results for smaller problems such as this (Martinez, Espinouse, & Di Mascolo). Furthermore, precise solutions take up way more simulating time for an assumingly small amount of performance increase.

### 2.6. Research design

#### 2.6.1. Research type

For this research I intend to do an experimental study on optimization and simulation. Firstly, I built the simulation based on the data provided to me. Then I experimented on the simulation with different configurations and evaluations as to give a conclusion on what decisions would be profitable to take.

#### 2.6.2. Subjects of Research

In this research I have looked at employees, as well as patients. Patients will be considered in a passive sense, since their actions do not change. This does not mean customers lose their right to determine their time of service, they lose control of the exact time of service. The indicated arrival time the customer currently indicates is already exceeded often in the current system. The solution here should lead to customers getting more security of service in a time window instead of late service in an exact arrival time system. Employees, however, will be researched in their movement across patients and the travel time thereof.

#### 2.6.3. Key variables

The most important key variables are the scheduling interval and amount of vehicles as explained in Section 2.1.2. The KPI's used in the simulation are defined as follows:

$$Efficiency = \frac{Time\ spent\ on\ providing\ care}{Time\ worked}$$

$$Punctuality = \frac{\text{Amount of ontime appointments}}{\text{Total amount of appointments}}$$

$$\text{Average waiting time per client} = \frac{\sum(\text{Arrival time} - \text{start of appointment interval})}{\text{Amount of clients}}$$

Where  $\text{Arrival time} - \text{start of appointment interval} \geq 0$

$$\text{Average late time per client} = \frac{\sum(\text{Arrival time} - \text{end of appointment interval})}{\text{Amount of clients}}$$

Where  $\text{Arrival time} - \text{end of appointment interval} \geq 0$

$$\text{Average cost per client} = \frac{\sum(\text{travel costs} + \text{salary costs})}{\text{Amount of clients}}$$

Do observe that all summations occur over all patients visited.

#### 2.6.4. Data type

The data used in the research are of a quantitative type, such as distances between patients or hours worked. This means that the output can also be viewed in a quantitative sense. With the use of the Law of Large Numbers the output of a Monte-Carlo simulation can be analysed in percentages and absolute improvements compared to the current situation. The recommendations for the amount of vehicles are then of quantitative nature, buy one car, do not keep more than 2 regular bike routes etc. The scheduling solution is then more of qualitative nature. Here the question is whether to implement this, not how much, yet the basis of that conclusion is derived from the quantitative results.

#### 2.6.5. Analysis type

For the usage of the data a Monte-Carlo simulation was used. The result of each configuration with this simulation has provided a set of numbers for the KPI's, which can be compared with each other to find which configuration is the one to adopt.

#### 2.6.6. Limitations of research method

The current issue that this method faces is the future application. The results of the experimentation can be relevant today, but if tomorrow some key variables change, such as a war in Ukraine driving up fuel prices making car travel extra costly, then the conclusion or feasibility thereof might change. Furthermore, the more data is gathered over time, the more accurate the conclusion will be. This means that after a few years of gathering more data the results might be even more accurate and representative than the current situation. Lastly, the research design takes previous data into account to predict what the best outcome would be for the future. It might mean that the decisions recommended will be best going forward, but are not beneficial to the specific current/future situation. An example of this could be that the company is currently in a statistical outlier situation where they drive 3 times more kilometres than usual. This would make bike travel very inefficient, while in the usual state of business it would do just fine.

#### 2.7. Assessment of validity and reliability

For a research it is important to know whether the research matches with reality, which refers to the validity of research. The consistency matters, which is measured by assessing the reliability of results. These are definitions by Cooper & Schindler(2014) who also provide three types of validity as well as reliability.

### 2.7.1. Validity

The first type of validity is content validity. Content validity is based on the used data and if it compares to the real world. In my research the content validity is proper, because the data is gathered from up-to-date databases which are filled in by the employees themselves. Furthermore I have heard of no external causes that influence the dependent variable further than the topics I have data on.

The second type of validity is Criterion-Related Validity, which is concerned with the extent to which predictions or estimations are used in the research. This might be of concern for this research as there is certain data I need, but is not formally gathered. Examples of this are percentages of evening and morning patients, and customer cancellation rate. If this data is collected in the future the simulation must be adapted and run again for more valid results. Subsequently, I have then to translate the employees description of events into numbers, which will undoubtedly lead to inaccuracies. These estimations then do have an effect on the Criterion-Related validity, but can be corrected by means of re-experimentation in the future, if the company collects enough data on it.

The last type of validity, construct validity, conveys the danger of different definitions to the test subjects, but this is not really applicable to this research as it only uses non-opinionated data.

### 2.7.2. Reliability

The first type of reliability is stability. Stability means that the results do not vary significantly when the experiments are repeated. In my research, this should not be an issue at all as the number of repeated experiments of the Monte-Carlo simulation are adequately large. Essentially this means that if the experiments are repeated enough times, the law of large numbers would make it so that two separated instances of such a large repetition would get more or less the same results.

The second type of reliability, equivalence, concerns with the consistency of the measuring instrument along time. This will not be an issue for my research as there are no instruments involved that can be biased. Each computation will be done the same, apart from the introduction of randomness, which is based on data. Therefore there is little to no equivalence issue.

The last type of reliability is internal consistency. This method would make you split the process in half and compare results to see if there is a correlation. From the separate analyses of cities it can be concluded that for all cities the result lead to a relatively similar conclusion.

## 3. Model Description

In order to make the schedule of Betsy run more efficiently, an algorithm can be constructed to find solutions. This algorithm would then look at the possible combinations of customers and decide what configuration of customers would result in the improved value for the company. In this section the algorithm chosen will be explained, as well as the corresponding objective functions.

### 3.1. Assumptions

In the model and simulation there are a set of assumptions that are done in order to make things work properly.

1. there are an unlimited amount of bikes available at all times. This is so because the way the company runs, all employees have access to at least one bike.
2. Vehicles that are used can be utilized in the morning routes, as well as the evening route. Take for example 1 car, this single car can drive in both morning and evening routes.

3. It is not worthwhile to invest in a vehicle that can only be used on one of the two daily routes. This assumption is there because there are cities where there are more evening routes than morning routes, which would mean that filling all evening slots in those cities with extra vehicles results in an excess of vehicles in the morning.
4. The vehicle speeds are not constant but are subject to randomness. An explanation would be that different factors such as weather, or traffic, can influence the average speeds drastically and this needs to be taken into account.
5. Customers do not stay in the system forever. The schedule of Betsy is subject to change as there are customers present that require short-term care for injuries, as well as long-serving customers. This means that customers should be removed from the system after a period. Each customer will receive an amount of time in which it remains in service (e.g. 5 weeks).
6. There are enough morning/evening routes to serve all customers. Betsy has arrangements with their clients about what time of day customers are served. This means that too little routes will result in a breach of this arrangement which should be avoided. This assumption then ensures that all customers planned in on a certain route are served.

### 3.2. Adapted Greedy heuristic

The algorithm used to optimize the travel plan of Betsy will be an adaptation of the greedy algorithm. The version described in the paper (Martinez, Espinouse, & Di Mascolo) has some constraints that are not relevant for this specific home healthcare type. This type of heuristic is useful as solving the MILP exact could be done, but it would be a time-consuming process within the simulation. A heuristic such as the Greedy heuristic is much faster and easy to program, and since the restrictions for arriving are not particularly strict, sub-optimal heuristics solutions provide enough.

First of all, there is no education requirement for serving patients, as service consists of small and simple tasks.

Furthermore, there are no restrictions on labor time as most employees work with a flex contract. This means that there are no standard working hours or maximum worktime that needs to be factored in.

Additionally, all routes contain 1 caregiver, and the amount of caregivers present at a point in time is an input variable in the simulation. This entails that rest days or employee pools are not relevant here. This is so because for the company it is useful to know how they can best drive their routes, and not necessarily who drives those routes.

For the heuristic to work there need to be start- and end points. Do note that all routes are divided among employees, so the amount of customers that need to be visited, and their according times are fixed. This means that the first customer, and last customer each route are known. The first customer then acts as the starting depot, and the last customer acts as ending depot. So in the Greedy heuristic solution these depots are fixed and cannot be swapped with customers that are in between.

#### Parameters:

Notation	Definition
$V_t$	Set of clients on route t
$V^*$	Set of all clients that are not start – or end points

$V_t^*$	Set of clients on route t with start-and end points excluded
$v_s$	First client of the day (fictive starting depot)
$v_a$	Last client of the day (fictive ending depot)
$N_t$	Set of employees working on day t
$T$	Time Horizon
$[a_j, b_j]$	Hard time window for clients
$d_t$	Duration of service for customer t
$n_j$	Care worker j assigned
$t_{ij}$	Travelling time between client i and j

The only decision variable that is present here is customer  $j$ . If all customers  $i$  have a customer  $j$ , where the first customer  $i$  for every route is the starting customer, then we have a complete route.

**MILP formulation:**

$$x_{ij}^n = \begin{cases} 1, & \text{if worker } n \text{ provides services to client } j \text{ after providing services to client } i \\ 0, & \text{otherwise} \end{cases}$$

$$s_j^n = \text{starting time of the service provided to client } j \text{ by care worker } n$$

$$w^n = \text{time not spent servicing clients of care worker } n$$

Objective function:

$$\min \sum_{n \in N} w^n \quad (1)$$

$$\sum_{i \in V} \sum_{n \in N} x_{ij}^n = 1 \quad \forall j \in V^* \quad (2)$$

$$\sum_{i \in V} x_{ij}^n = \sum_{i \in V} x_{ji}^n \quad \forall j \in V^*, \forall n \in N \quad (3)$$

$$(s_{v_a}^n - s_{v_s}^n) - \sum_{j \in V^*} \sum_{i \in V} x_{ij}^n \times d_j \leq w^n \quad \forall n \in N \quad (4)$$

$$a_j \sum_{j \in V} x_{ij}^n \leq s_j^n \leq (b_j - d_j) \sum_{i \in V} x_{ij}^n \quad \forall n \in N \quad (5)$$

$$s_i^n + (t_{ij} + d_i) \times x_{ij}^n \leq s_j^n + b_i(1 - x_{ij}^n) \quad \forall n \in N, \forall i, j \in V \quad (6)$$

$$x_{ij}^n \in \{0,1\} \forall i, j \in V \forall n \in N \quad (7)$$

$$s_j^n \in \mathbb{R}\{+\} \quad \forall j \in V, \forall n \in N \quad (8)$$

$$w^n \in \mathbb{R}\{+\} \quad \forall n \in N \quad (9)$$

### 3.2.1. Model elaboration

Equation 1 is the objective function which aims to minimize the value of the travel time and waiting time of a care worker. This variable is described in Equation 4, as the waiting time is the total time spent on duty subtracted by the amount of time actually spent serving customers.

Equation 2 should ensure that each customer is visited once by any employee that travels from a previous customer. Which is why equation 3 guarantees balance constraints of the process.

Equation 5 states that each service should be done within the specified time window of a customer. This is necessary because we want to be respectful towards the specified time windows. Where Equation 6 states that the service time added with the arrival time of one customer should not interfere with the latest possible arrival time to the next customer. This is essential because otherwise the service of one client disqualifies the next, and this minimizes the violation of the upper bound of the clients' time window.

Equations 7, 8 and 9 ensure that each variable has its proper format and are nonnegative. This is done to preserve valid results as  $x_{ij}^n$  is Boolean, and should be formatted as such. Furthermore, the allowance of negative numbers would imply the possibility of negative waiting time and service time, which is not possible.

### 3.3. Solution Method

In real life, each route is a division of the city at hand. So if there are 5 routes in that city, the city is divided into 5 regions. These regions are then hard borders which employees do not cross. For the simulation this means that each improvement will occur on a specific employee route (e.g. Morning route of employee 2) instead of a whole section such as morning.

If we consider a specific employee route to be the whole horizon of an optimization problem, then the start point is the customer on the schedule with the earliest start times. and the end point will then be the customer with the latest scheduled time. Everything in between is subject to change.

In order to make the Greedy heuristic function properly, an initial solution is advised. A normally generated schedule is then used as the initial solution. Think of this as a schedule with the patients ordered by appointment time. The generated schedule is then adapted to eventually receive the lowest value for equation 1. The lowest value is obtained by swapping patients in the schedule and check if the value of equation 1 received is lower than the current value. The initial solution is then how the schedule would look under current operations, and should therefore come close to fulfilling all requirements already.

For the minimum value of the objective function it is necessary to loop over every possible permutation, subject to the constraints, of the schedule without repetition. This is done by

beginning at the starting customer and selecting which customers can be visited next without exceeding their maximum arrival time. A swap is then initiated for each possible next customer with the current next customer. The swap with the lowest total route duration, or  $w^n$ , remains in use for the continuation of the loop.

Once the loop is finished, it should return the schedule with the lowest travel- and waiting time. For the speed, the average speed of the bike is taken, as that is the current mode of transportation. This speed is necessary to compute the expected arrival time. This expected arrival time determines at what time the employee is expected to arrive at the customer, and it should not exceed the customers' time window..

The loop in question should of course not violate constraints from the model. Equations 5 and 6 are the only ones endangered by swapping the customers around. So in the loop it is checked whether a swap between customers would mean crossing a time window. If this is the case then the swap is cancelled. It is also possible that equations 5 and 6 cannot be satisfied due to the crowdedness of the schedule and input parameters. In this case very few swaps are done and the schedule may remain the same.

After the loop is done an improved schedule emerges with different ordering of customers. This adapted schedule is then inserted back into the regular schedule sheet so arrival times, delays etc. can be calculated. In Figure 6 an example of how such a swap is done can be seen. In the simulation itself each swap combination is explored, with the help of the Greedy heuristic the most time efficient swap for that particular customer is chosen. If the new order of customers is faster, and does not exceed any time boundaries, then it is considered the best one for that employee in the week.

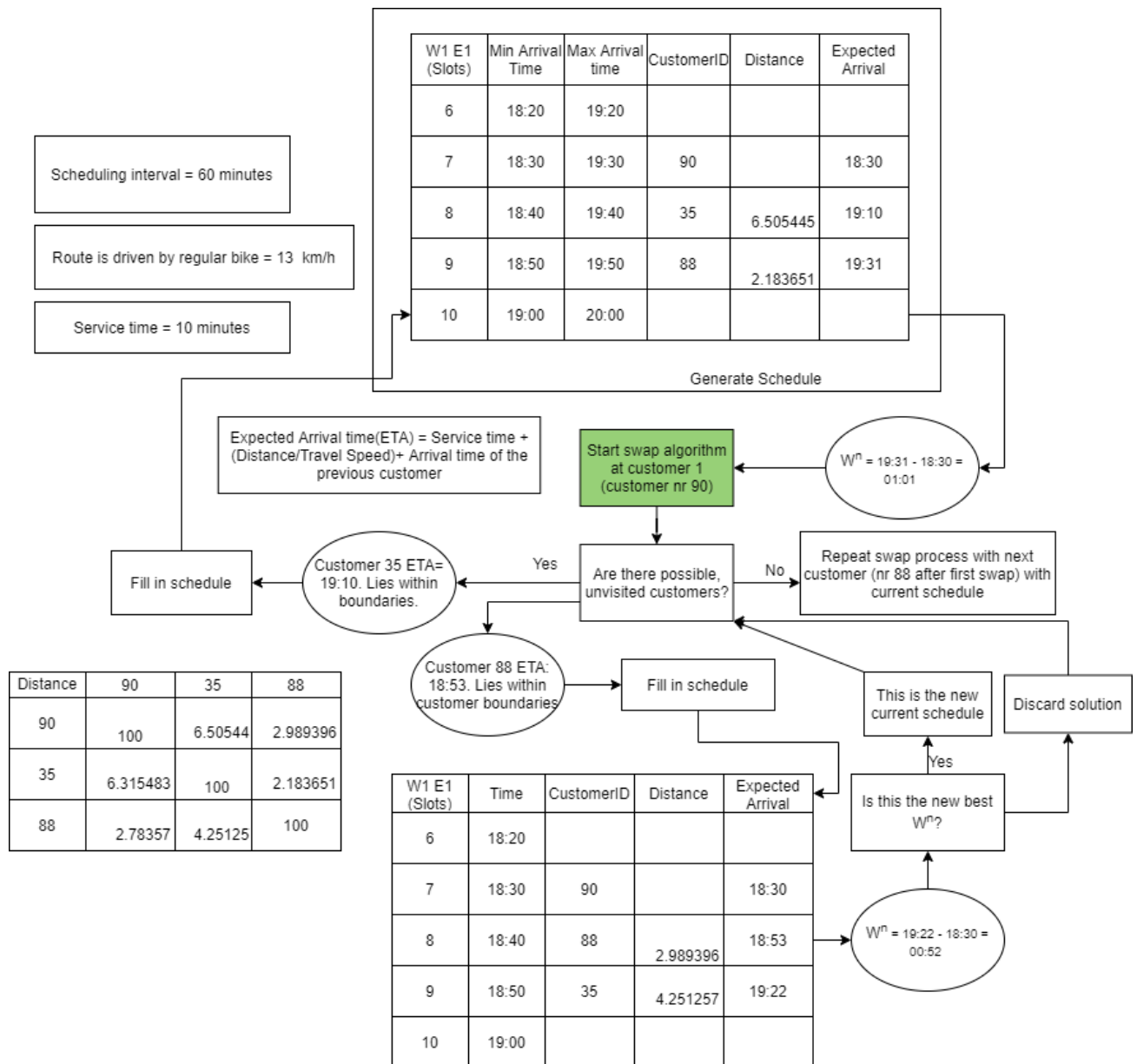


Figure 6: Algorithm swap example

## 4. Simulation description

In this section I will be describing all processes present in the simulation. This is done for transparency reasons, as well as an insight into how the model is constructed.

### 4.1. Input Data

In this sheet all input parameters of the simulation are presented. Below a table is presented with all the variables necessary to make the final simulation run. Each of these fields should contain a value, otherwise the simulation might malfunction.

Variable	Type	Range
Amount of cars	Integer	[0 to ∞)
Amount of bikes	Integer	[0 to ∞)
Amount of e-bikes	Integer	[0 to ∞)



City	Selection	[Cuijk, Doetinchem, Ede, Vlaardingen, Weesp, Zutphen]
Amount of morning routes	Integer	[1 to $\infty$ )
Amount of evening routes	Integer	[1 to $\infty$ )
Enable routing algorithm	Boolean	[No or Yes]
Scheduling Interval	Integer	[0 to 240]
Weeks per simulation (WeeksNr)	Integer	[0 to $\infty$ )
Iterations of the simulation	Integer	[0 to $\infty$ )

Table 3: Input variables table

#### 4.2. City Data

In this sheet all the data that belongs to a certain city is displayed. The table below elaborates on the distributions used and derived in Section 3.

<i>Distribution</i>	<i>Explanation</i>
Client Distance Distribution	This distribution generates the distances one customer has to another. All pairs of customers have a generated distance.
Distance Asymmetry Distribution	This distribution generates the standard deviation in percentages of the difference between A to B and B to A.
Client Duration Distribution	This distribution generates how long a customer will be in service.
Client Demand Distribution	This distribution generates the starting amount of customers in the system of a particular simulation iteration.
Client Change Distribution	This distribution is the change in customers over the weeks. This then builds on the Client Demand Distribution to get the total amount of customers in the system on a given week.
Client Slot Distribution	This distribution describes the slot number of the patients.

Table 4: Distribution table

#### 4.3. Variables

This sheet stores all variables that are essentially fixed real-world constants. These contain vehicle speeds, as well as times that the company is using. These variables can change in the future, but would require substantial changes to technology or the way the business operates. The value of vehicle speeds was gathered from the web (Gazelle, 2022) & (Centraal Bureau voor de Statistiek, 2002). The rest of the values are determined by the company, or derived from suggestions the company gave. Table 5 contains all the variables that are used for simulation. To clarify, morning slot probability is the chance that a customer is only served in the morning, marked as a type 1 customer. Evening slot probability then is the chance that a customer is only served in the evening, which is defined as a type 2 customer. Following that the Morning+evening slot probability is the

chance that a customer is served both in the morning and the evening, also denoted as a type 3 customer.

#### 4.4. Generate Matrix

This sheet generates a table that describes the distances between customers. The way it does so is first by looking at the amount of weeks and generating enough customers over the entire time horizon. All Betsy locations investigated have never operated with more than 80 existing customers in a week, and rarely go over 5 new customers a week. So in order to guarantee enough customers over the entire time horizon, the total amount of customers starts out with 100 and adds 10 extra for every week the simulation runs. With the amount of customers known, a square matrix can be made and filled in with distances according to the Client Distance Distribution by using the *rnd()* function, which is used to gather numbers from all other distributions as well.

Steps:

1. Dimension variables and clear previous matrix on the sheet
2. Determine for how many customers a matrix should be generated. The current formula used is:

$$\text{Total customer amount} = 100 + 10 * \text{weeksNr}.$$

Then label the rows and columns up to the total customer amount value.

3. Fill all cells within the created space with numbers gathered from the distribution of the city. This is done by looping over the columns for every row value. So each iteration of the loop gets assigned a random variable  $k$  between 0 and 1. This value, in combination with the parameters of that specific city, can be used to get a value for the distance by relying on a fitted distribution. All distances follow a lognormal distribution, so to generate a distance for customers we use the lognormal inverse function. So the distance from A to B can be computed as  $D(A, B) = \text{Lognorm.Inv}(k, x_1, x_2)$ , where  $k$  is a random variable between 0 and 1. Here  $x_1$  is the parameter belonging to the mean, and  $x_2$  corresponds with the deviation parameter. Each parameter has its own value per city, which makes for different distributions.
4. If the row and column values are the same, the distance allotted is 100, since no distance generated will ever exceed that value. This is necessary since it should be impossible to route a customer back to itself.
5. If the value of the row exceeds that of the column, the value of the cell with the inverse coordinates is used and provided with an additional change distribution. So, for example, the cell with coordinates (3,1) needs to be filled. In order to do this the value of cell (1,3) is used and added with the value of an inverse normal distribution.

This is done because the distance from A to B does not equal the distance from B to A, but they are correlated. In the schedule this is calculated as follows:

$$D(B, A) = D(A, B) + (1 + N_{inv}(k, 0, x)).$$

Here  $D(A, B)$  is the distance from point A to point B, and  $N_{inv}(k, 0, x)$  the inverse normal distribution with random variable  $k$  as percentage point, an average of 0 (as the deviation should include equal positive and negative values), and the standard deviation  $x$ . Parameter  $x$  is then drawn from the Distance Asymmetry distribution.

#### 4.5. Customer base

The customer base sheet consists of an initialization and a continuation. Both of these phases are subject to several processes. These processes are customer amount generation, customer base generation, route division, slot division, and service time generation.

The initialization phases are done once every iteration in the first week, where an iteration is basically a complete rerun of one simulation where only performance indicator variables are saved. Iterations consist of an amount of weeks, so 2 iterations of 5 weeks, means that the simulation runs for 5 weeks, cleans everything except monitoring variables, and runs 5 more weeks. These iterations are then useful to test the effects of entities generated once per simulation such as the patient distance matrix.

Continuation phases are done every week in the simulation only after the initialization phase has passed. A week in the simulation is then a continuation of a previous week, after week 1 of course. This means that some parameters used, such as amount of customers, in week 5 are dependent on the values in week 4. The use of weeks allows to monitor the behavior of the operation over a longer time horizon.

#### 4.5.1. Customer amount generation

In this process the amount of customers for a specific week is determined. This is directly linked to the size of the customer base.

##### *Initialization:*

In the initialization the starting amount of customers is drawn from a Client Demand Distribution that belongs to a city. A random variable  $k$  is newly generated with a value between 0 and 1. The variable  $k$  essentially stands for a percentile. So a  $k$  value of 0.94 means the distribution will return the 94<sup>th</sup> percentile value, also known the value for which 94% of values are smaller, from the Client Demand Distribution. This again serves as a percentage point and is used to connect a value of  $k$  to a belonging customer amount through the Client Demand Distribution.

##### *Continuation:*

The continuation draws upon the value of the amount of customers from the previous week, and adds- or subtracts a number of customers from that value. The reason to do this stems from the rapid changing customer base of Betsy, and this changing of the amount of clients seeks to emulate that phenomenon. The number added to the customer base is drawn from the Client Change distribution which returns a value that can be both negative or positive. A negative result means that customers are leaving service, while a positive result means customer are added in service. Results between -0.5 and 0.5 are rounded to 0, meaning the amount of customers in that week remains the same. It does this by using random variable  $k$  with a new randomized value. This  $k$  is then connected to a specific distribution value. This value is not an integer, and since fractions of a customer are not possible, the value is rounded to the nearest integer and added onto the amount of customers from the previous week. Every week a change in customer amount is possible, so every time a week progresses this process is repeated.

The motivation behind this is to simulate the volatility of the schedule, and the impact on operations. Moreover it seeks to test the reaction of operations to different sets of customer amounts. This is why in the end, the simulation will be run several iterations for a few weeks (i.e. 100 times with 10 weeks each) , to test both those occurrences.

This essentially turns the amount of customers into a Discrete Time Markov Chain, as the probabilities of change remain the same, all probabilities sum to 1, and all values of the customer amount are applicable to the system. This system is discrete as the variable customer amount can only be expressed in integers, since 0.5 customers do not exist. Figure 7 illustrates how such customer change is handled. This change is then done every week, for as long as the iteration runs.

The values of  $m$  and  $n$  in the figure are dependent on the city which is investigated. Some cities change much, others not, but from analysis done down in Section 6 it is clear that  $m$  does not go over 3, and  $n$  stays below 4.

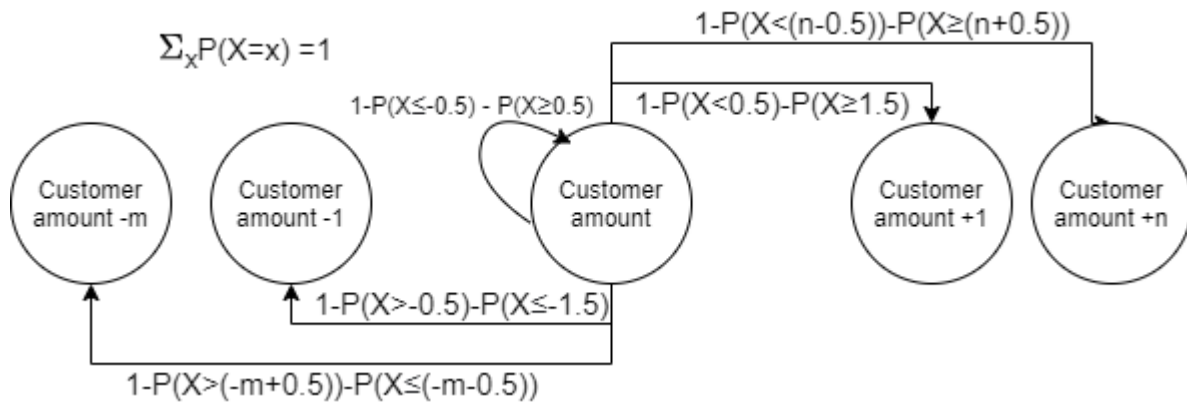


Figure 7: Discrete Time Markov Chain of the customer amount

#### 4.5.2. Customer base generation

This process generates the customerID's and customer types to work with for a week. A customerID is a unique value to indicate a specific customer. Its value ranges from 1 to the total customer amount from the Generate Matrix Sheet. A customer type indicates what type of services that customer requires. Type 1 indicates only morning service, type 2 indicates only evening service, and type 3 indicates morning, as well as evening service.

##### Initialization:

In the initialization phase the first column is filled with unique customerID's until the Customer amount from the previous step is reached. Each customerID is provided with a customer type according percentages given by the company and generated again with random variable  $k$ . If  $k \leq x_1$  then customer type is 1, if  $x_1 \leq k \leq x_2 + x_1$  then customer type is 2, and if  $k \geq x_2 + x_1$  then customer type is 3. Note that  $x_3 + x_2 + x_1 = 1$ .

##### Continuation:

The continuation process is about altering the customer base to the changes in customer amounts. First, it copies the table from the previous week with all customerID's and belonging customer types, as the customer types do not change over the weeks. Accordingly, it is checked whether customers are added or removed in the system. If there is no change in customer amounts, the table will remain the same. If there are customers added, additional customerID's and customer types will have to be generated until the customer base has the amount of customers necessary. If customers are removed, the shortest time left in service will be identified (see later section). This customer is then removed from the system. It repeats this process until the customer base has the amount of customers necessary. Figure 8 is an illustration of the decision tree the simulation goes through every time the customer base needs to be generated for a certain week.

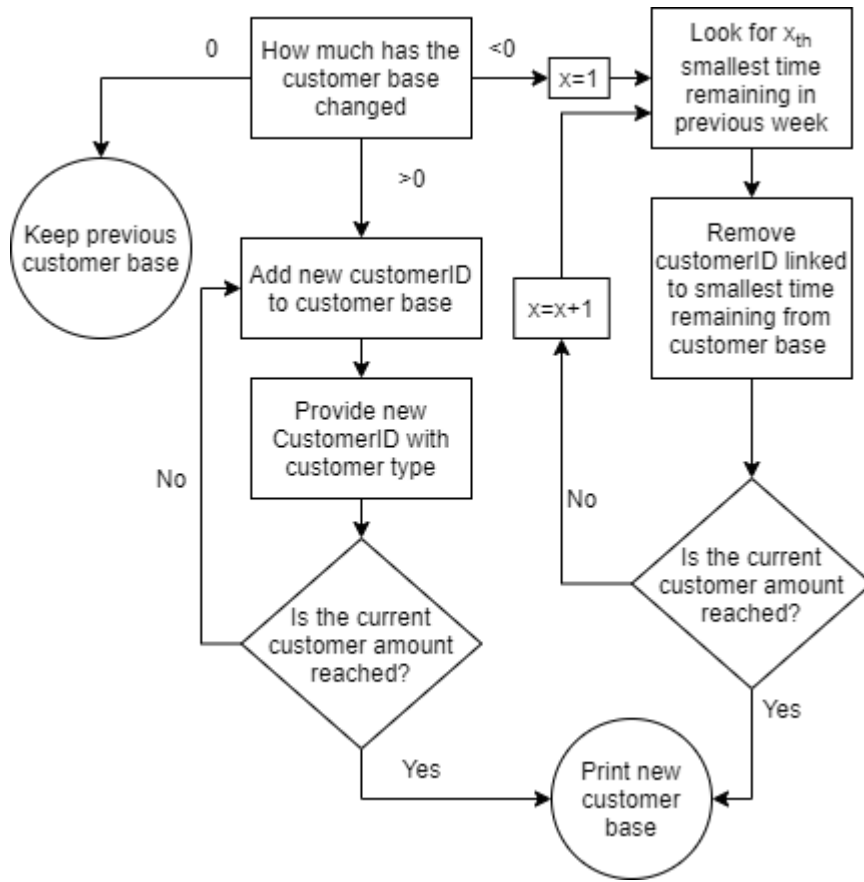


Figure 8: Decision tree of the simulation regarding the customer base

#### 4.5.3. Route Division

For a customer to be served he/she needs to be linked to an employee. It is then important that each employee has an average workload. Each day is divided into a morning route and an evening route. The amount of morning and evening routes are specified in the Input Sheet and determine how many employees are working at a time.

After the previous step was completed all customers are divided into types. Customers with type 1 and 3 are divided into the morning schedule, and customers with type 2 and 3 are inserted into the evening schedule. The customers of both morning and evening routes are ordered based on a Nearest Neighbor algorithm. The starting point is the first customer in the customer base column, and then there is a loop over all customers of the specific morning or evening route to find the patient with the shortest distance to it. The customerID belonging to that patient is ignored further, so only customers that are not ordered yet remain. This loop is repeated over remaining customers until all customers are ordered. The result of this is that one grand tour is created that should give an approximate shortest route to all customers.

Subsequently, this grand tour is divided among employees. Each employee needs to have an approximately the same amount of customers, so the total customer for a route is divided by the amount of employees. Each employee will get an interval appointed. An interval of  $[x,y]$  means that an employee has to serve the  $x^{\text{th}}$  customer in the grand tour up until the  $y^{\text{th}}$  customer, where  $y > x$ .

$$\text{Customers per employee} = \frac{\text{morning/evening total customers}}{\text{morning/evening employees}}$$

This customers per employee ratio can also be a fraction, if the amount of customers are prime for example. This is solved by rounding the amount of customers to the nearest integer. Say customers per employee =  $q$ , where  $q \geq 1$ , and there are  $r$  employees. Then the interval of customers each employee has to serve is:

$$\text{Upper bound serving} = \text{Integer}(r * q)$$

The lower bound of employee  $r$ , is the upper bound of employee  $r-1$ , with one added customer.

$$\text{Lower bound serving} = \text{Integer}((r - 1) * q) + 1$$

After all employees are looped, all customers have one employee they will be served by. The letter M indicates a morning employee, and the letter E indicates an evening employee. So M2, for example, is the second server in the morning. This can be done as all employees start from their own homes. The distance from their house to the first customer is not covered. This negligence also counts for the distance of the last customer to the house of the employee. Figure 9 below clarifies how such a routing system would work in practice.

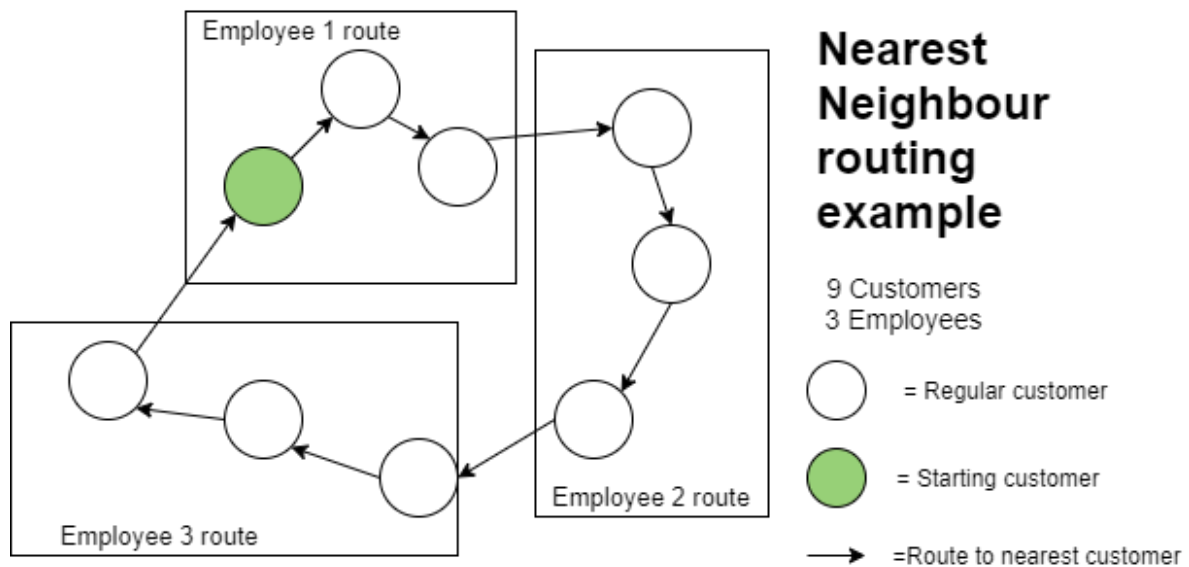


Figure 9: Example of customer grouping within the simulation

#### 4.5.4. Slot Division

For a customer to be served by an employee he/she needs to be allotted a time first. The start time and end time of each employee is the same. Furthermore the service time differs between morning and evening. Also, Betsy does not take travel time into account, so all customers are scheduled right after each other. The amount of slots in a route is then described as:

$$\text{Slots} = (\text{End time} - \text{Start time}) / \text{service time}$$

The time at which a slot takes place is then formulated as:

$$\text{Slot time} = \text{Start time} + (\text{Slot number} * \text{Service time})$$

*Initialization:*

Each employee can only be at one place at a time, so one slot number per employee is available. This slot is then drawn from the Client Slot Distribution. The random variable  $k$  returns with a new randomized value between 0 and 1. This value of  $k$  returns then a corresponding slot number value,

which then has to be rounded to the nearest integer as there are only whole slots available. The concentration of this distribution is often large, as most people wish to be served from 08:00 to 10:00 or from 18:00 to 19:00. It is not feasible to help all clients in these time windows, so customers that want a slot that is taken will be asked to pick a new time until they find a slot that is empty. In the simulation this is represented by generating a new slot number until a unique value is found for a customer with an employee.

*Continuation:*

Customers that are assigned a slot will be served at that slot during their service duration. The only exception to this rule is the transfer of customers between employees. If a customer goes from employee  $r$  to employee  $r+1$ , then his usual slot might already be taken by an existing client. In this case the customer will simply have to pick another slot until an empty slot is generated. Figure 10 provides an example of the procedure of generating slots.

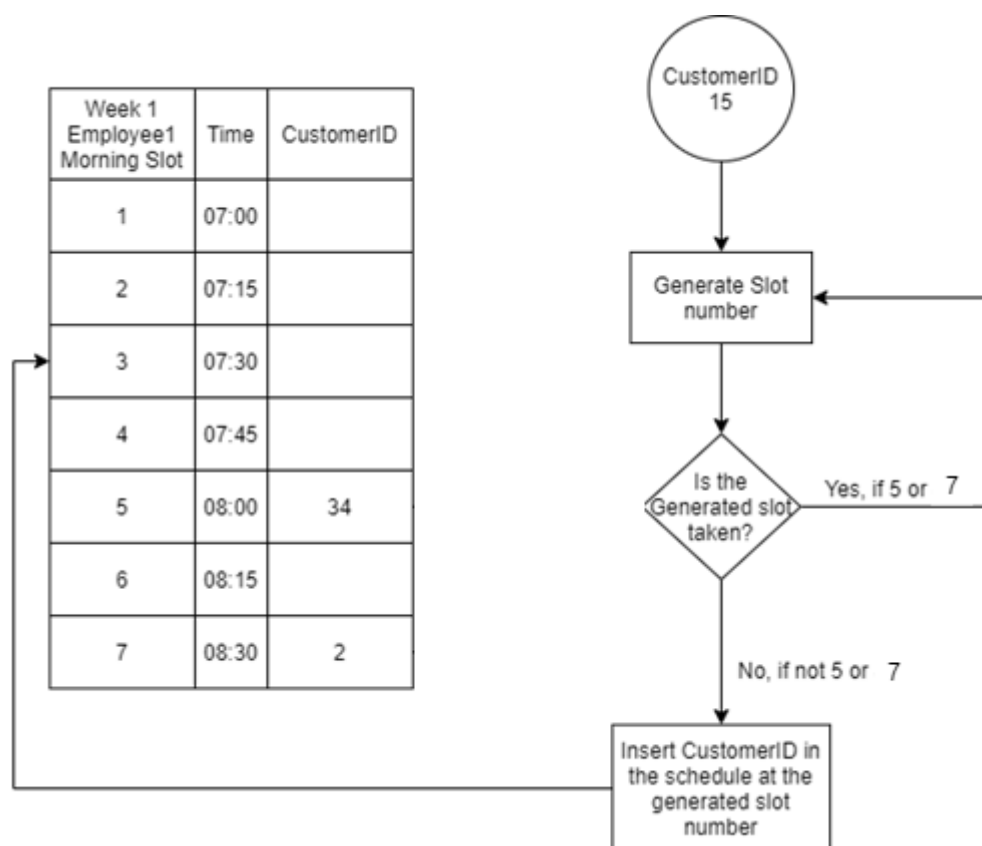


Figure 10: Slot generation example

#### 4.5.5. Service time generation

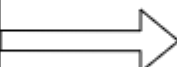
A customer that is in service will not be in service forever. Some people go out of service because they were treated for an injury that is now healed. But others stay in service until they pass away or are transferred to an elderly care facility. Both of these scenarios have a very different service time. The injured patient will likely stick around for a couple weeks only, while the elderly person might be in service for years. For the sake of imagery it is useful to know that at least 1 customer every route has a service time that expires in the near future ( $<10$  weeks). This may not seem like an important part, but one different customer can have a lot of impact on the total driving distance, as well as the routing heuristic from the previous section.

*Initialization:*

The time a customer is in service is drawn from the Client Duration Distribution. Random variable  $k$  is back, and is assigned a new randomized variable between 0 and 1. This value once again is linked to a unique value for the customer duration. Since the Client Duration distribution is set in days, the generated number has to be divided by 7 and rounded down to the nearest integer.

*Continuation:*

When a week progresses, the value of the service time generated is decreased by one. If the customer is new and has no service time yet, one is generated for him/her. When the service time of a customer reaches 0, that customerID is removed in the next week and replaced with a new customerID. This means that if a customer with type 3 (both morning and evening) has a service time remaining of 0, he will be removed from the system as a whole. This is in accordance with the experience of the company that conveys that customers do not drop individual routes' service. In Figure 11 an example is shown from a replaced customer. In the image, the customer that has slot 4 in the first week, with the first employee is customer number 14. The last column shows that there are 0 weeks of service left with the client, which means that it needs replacing. Therefore, the old customer ID and Slot duration are erased, and a new customer ID with a new slot duration is generated. This essentially means that a new customer has taken the place on that slot from the old customer.



Week 1 Employee 1 Evening Slot	Customer ID	Slot Duration
4	14	0

Week 1 Employee 1 Evening Slot	Customer ID	Slot Duration
4	52	64

Figure 11: Example of customer regeneration

## 4.6. Schedule Generation

The generation of the schedule consists of 4 parts. The first part is formatting the schedule. The second part consists of filling in the customerID's. The third step is to insert the distances between customers. Lastly, the arrival time and schedule deviation need to be injected into the schedule.

### 4.6.1. Filling CustomerID's

In the slot duration sheet all customers in a week are given an employee, and slot number. From the codes present in the schedule it is then possible to search for the customers that belong in certain route. In the Slot Duration schedule it is made sure that no slots overlap with an employee, so each employee visits one customer at a time.

In Figure 12 an example is shown of how the customers are put into the schedule. Schedules are numbered by codes. The letter 'W' with a number indicates for what week the schedule is. The letter 'E' with a number conveys the employee number in the evening, where the letter 'M' is for morning employees. So W1 E2, for example, is the schedule in the first week of employee 2. These combinations are also found in the 'Slot Duration' part, which is where the schedule generator takes its information from. First an empty schedule is generated according to the parameters specified in the variables. Following that, the week and employee are searched for a specific schedule (e.g. for W2 M1 the program searches in the second week for the first employee). Then the slot numbers belonging to that employee are filled with the corresponding customerID's.



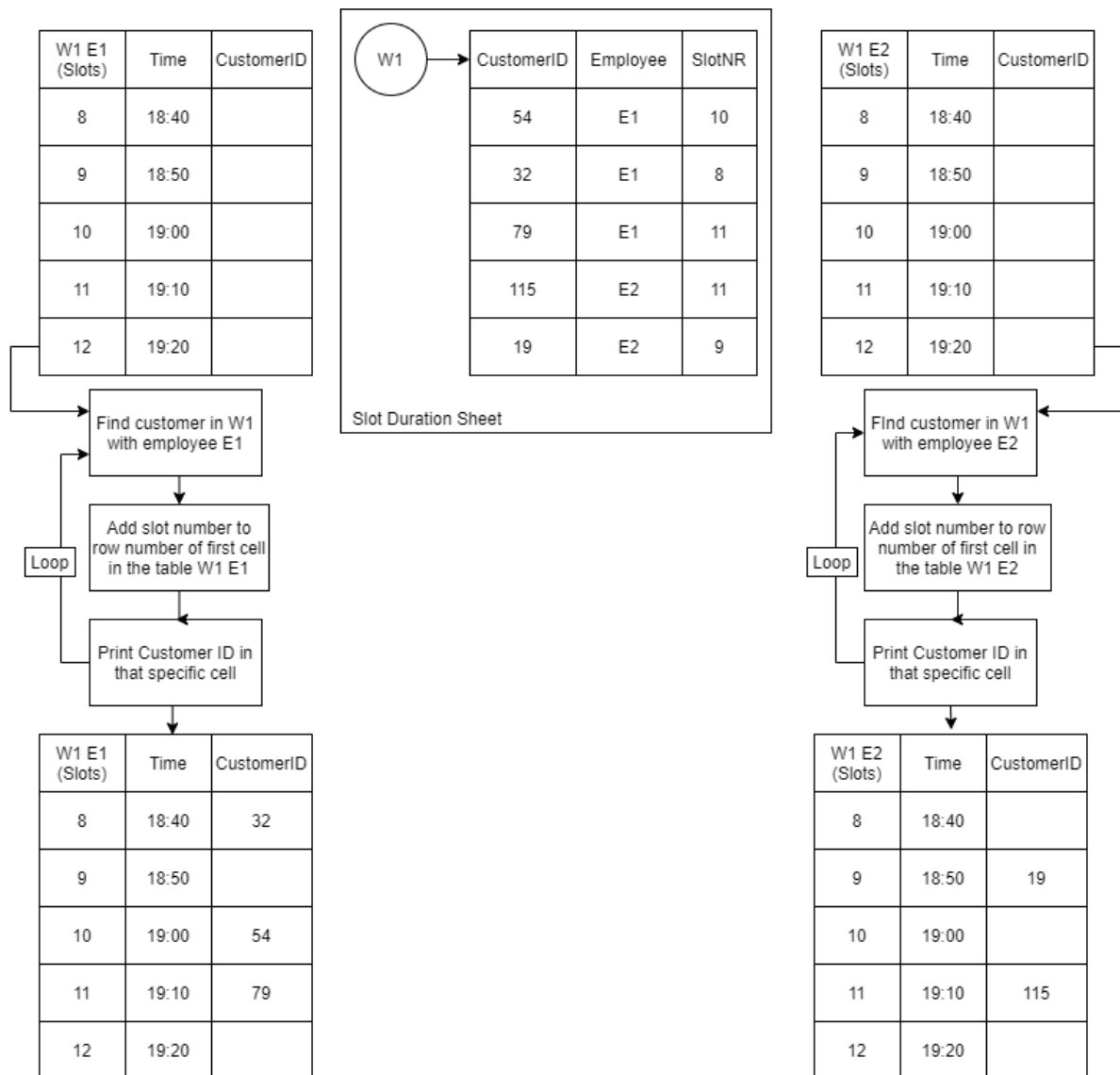


Figure 12: Example of customer data insertion in the schedule

### Insert Distances

In order to determine at what time an employee arrives at the customer it is necessary to derive the distances of the previous customer to the current customer. For computing these distances it is necessary to use the Generate Matrix sheet, where all distances are stored.

This is done by taking the previous customer as the starting point, or the row value in the matrix, and the current customer as the destination column value, or the column value in the matrix. The value of that particular cell is the distance from customer x-1 to customer x. The distance of the first customer in a route is always empty, as the employee starts there.

It can happen that there are gaps in the schedule. These gaps need to be crossed in order to find the right distance. This was done by first signaling what the first customer in the system is. If there are any empty slots before a customer, after the first customer has passed, then the simulation loops back cells until a customerID is found. This then found customerID is the previous customer in relation to the current customer, and will serve as the starting point. In the simulation it is possible to bridge a gap, as is seen in Figure 13, where the distance from one customer to the next is computed over 2 empty slots.

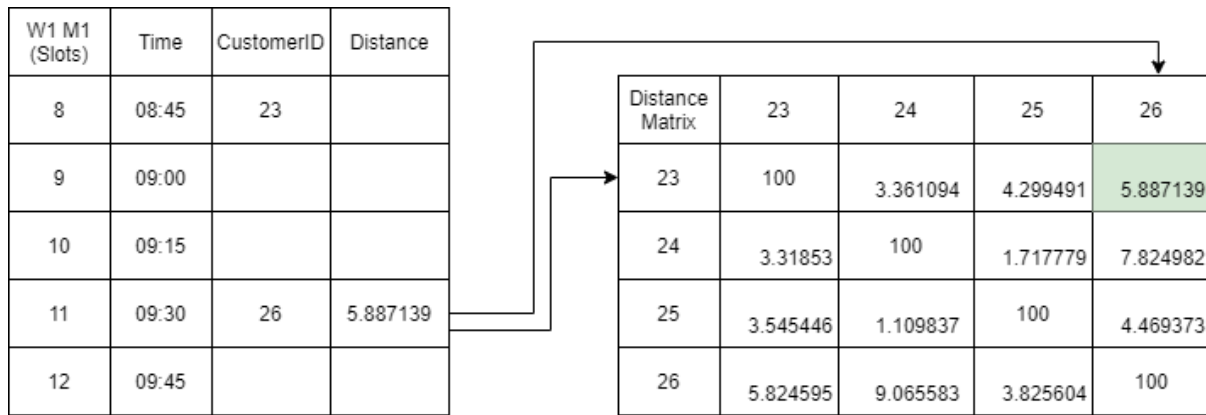


Figure 13: Illustration of distance collection procedure

### Arrival Times

To find out what time the employee arrives, one would need to factor in the time of arrival at the previous customer, service time, the travel distance and the travel speed. The delays are then computed by comparing the arrival time with the appointed time.

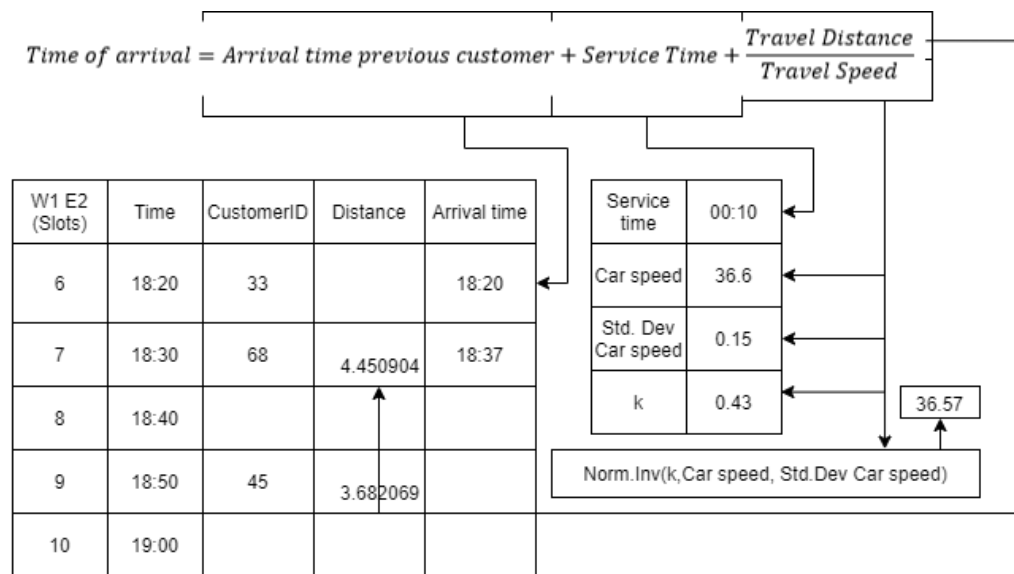
The time of arrival can be described as:

$$\text{Time of arrival} = \text{Arrival time previous customer} + \text{Service Time} + \frac{\text{Travel Distance}}{\text{Travel Speed}}$$

In this equation, *Arrival time previous customer + Service Time* is equivalent to the time at which an employee is done serving the previous customer. The arrival time at the previous customer is a recursive function that starts at the first customer of the route. So the arrival time at the previous customer is then dependent on all other variables. The service time is a constant and differs between morning and evening routes, its value can be found in the Variables sheet.

$\frac{\text{Travel Distance}}{\text{Travel Speed}}$  is the representative of how long it takes to go from the previous customer to the current customer. Travel distance was determined in the previous step. Travel speed is dependent on what vehicle is used. In the input sheet it is possible to identify how many of each vehicle is available. Due to every employee in real life having a bike, it is assumed in the simulation that the amount of bikes available are infinite. The availability of e-bikes and cars can be adjusted in the input user form. Each vehicle identified is available in the morning as well as in the evening, but not simultaneous during morning/evening routes as only one employee can use a vehicle at a time.

Figure 14 gives a complete overview with an example of how arrival time is calculated.



$$\text{Time of arrival} = 18:20 + 00:10 + 4.450904 / (36.57 * 24) = 18:37$$

Figure 14: Derivation of arrival time illustration

The way these vehicles are divided among routes is determined by distance. The routes with the largest distances will be given a car, up and until all cars are allotted. The consecutive largest distances will be given an e-bike, which will be sorted into a route until the specified value, and the remainder will have to travel by bike.

The travel speed follows an inverse normal distribution with the mean and standard deviation specified in the variables sheet. However, overall the car should be the fastest, then the ebike, and then the normal bicycle. Look at Figure 15 to find a decision tree on vehicle division among routes.

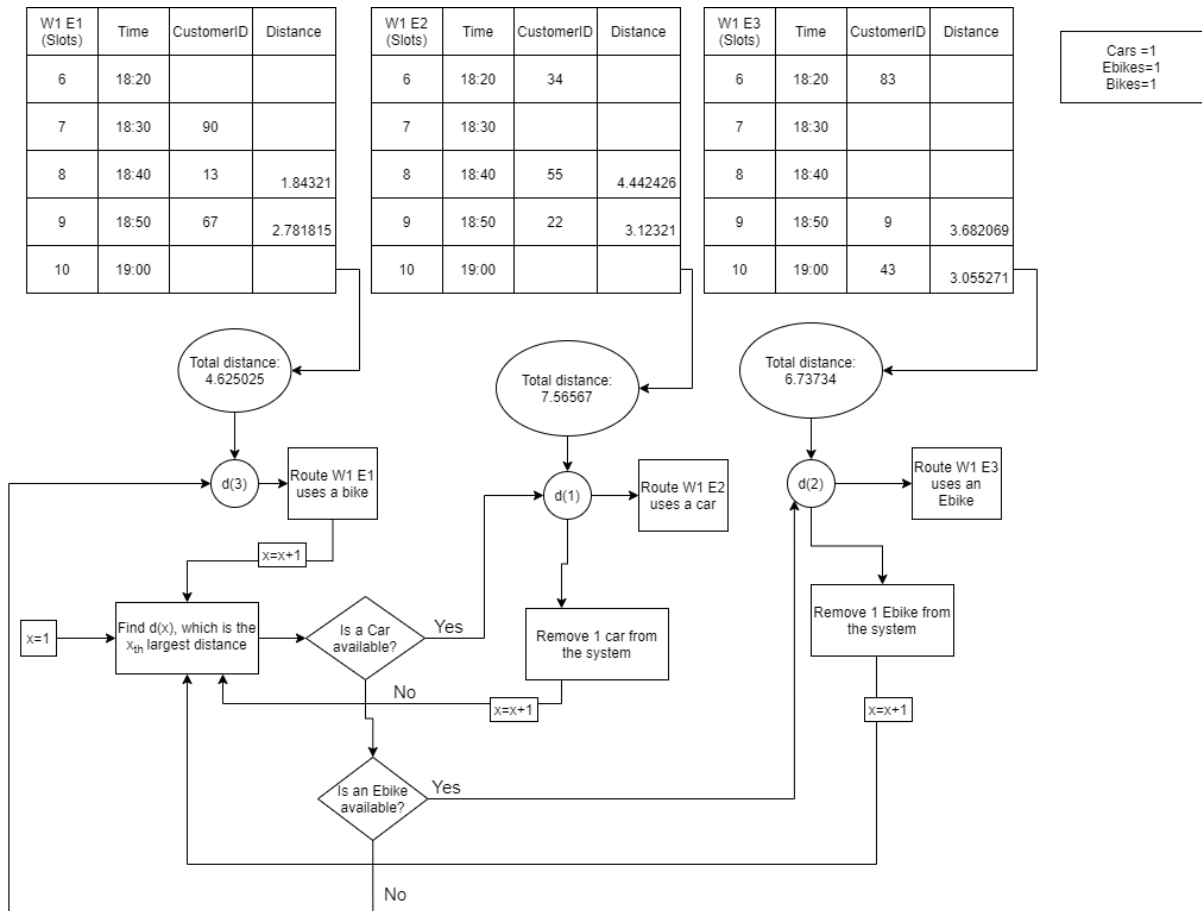


Figure 15: Division of vehicles procedure example

### Schedule deviation

The schedule deviation is the time between the employee arriving at the customer, and the appointed time. This time is also dependent on the scheduling interval inputted. In the schedule this is represented with 3 colours. Blue means that the employee is right on time, or early. The numbers inside the blue cells indicate how many minutes the employee is early. Green means the employee arrives later than the scheduled time, but within the scheduling interval boundary. The numbers inside the green cells indicate how much time the employee had left before they were actually late. The red cells signal when the employee is late, so later than the appointed time with the scheduling interval added. The values inside the red cells then convey how many minutes the employee is late. Figure 16 gives a legend with corresponding boundaries for each arrival time scenario.

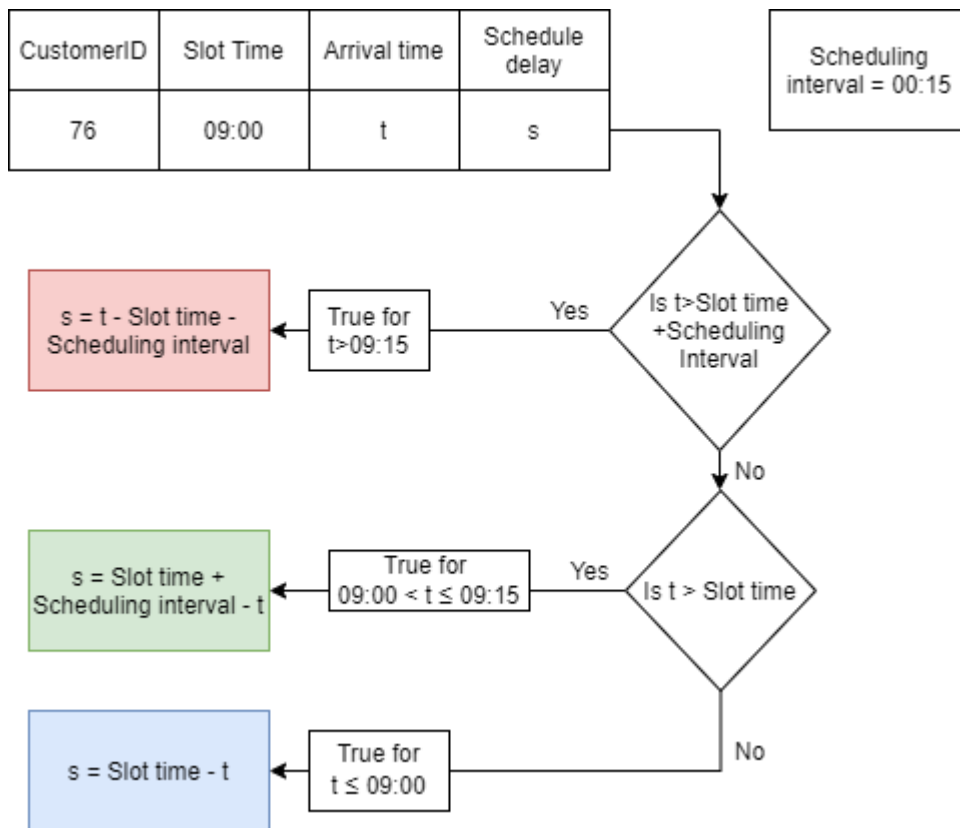


Figure 16: Legend and derivation of schedule delay cases

#### 4.7. Results

In the results tab, a variety of indicators is present, 5 of which are KPI's.

KPI	Meaning	Value
Average waiting time	The average time a customer is waiting to be served. This only incorporates the difference between scheduled time and actual arrival time.	Waiting time is generally not a good thing and should be minimized. Minimizing this factor would signal an improvement in operations.
Average late time	The average time a customer's maximum service time is exceeded, where the maximum service time is the scheduled time with the interval schedule variable added.	Late time is essentially a broken promise to the customer. Ideally this value should be close to 0, and minimizing it would identify improvement in the business.
Punctuality	The percentage of customers that are served before their maximum service time.	This number should be as close to 100% as possible as the company wants to be on time as much as possible. The higher this number is, the better the company performs.
Efficiency	The percentage of time an employee spends on serving customers while on duty.	This percentage should be as high as possible. An increase of this would either mean less time spent travelling/waiting

		or more customers served in a route.
Average cost	The average cost incurred for serving a customer.	The average cost should be as low as possible, as lower costs mean more profit margins for the company. Lower average costs could also indicate that the operation as a whole is being run more efficient.

#### 4.8. Limitations

In order to properly work with this simulation, there are certain input parameters that have restrictions. First of all, the amount of evening/morning routes is dependent on how many customers arrive, and how many slots are operational. This is so because every customer needs to be served by an employee on a given day. Not adhering to this would mean the simulation will endlessly search for a unique slot, making it run infinitely in the process.

Secondly, the amount of evening routes should be larger than or equal to the amount of morning routes. This is so because the simulation assumes that there will be more evening routes than morning routes and loops certain modules for the amount of evening routes. Not adhering to this requirement would make the simulation function, it would just skip some of the procedures used to construct a reliable simulation. This means that cities with flexible route amounts will always use their maximum amount of routes. The result of this is that if a city happens to have a low customer amount in a week, that week will have a more empty schedule instead of reducing the amount of employees. The only further limitation is that the interval schedule should not exceed the time between the start of the morning/evening and the latest customer every morning/evening route. Otherwise the scheduling part of the simulation will function as if two morning/evening are pooled, resulting in one person virtually doing the consecutive morning/evening route after he finishes his own route. Unfortunately this means that scheduling intervals larger than 5 hours are not possible to test in the future, which is not a problem for our case as the maximum tested scheduling interval is 1 hour.

### 5. Simulation setup

For the simulation setup I used a time span of 10 weeks, with 100 iterations for each different configuration. One configuration consisted of testing the performance all possible vehicle combinations for a given city, which will be referred to as the Vehicle adaption simulation. The second configuration consisted of testing different times for the scheduling interval and the effect it would have in combination with the algorithm. This type of simulation will be called the Algorithm-scheduling simulations. This was done so because 10 weeks is enough time to test the flexibility of the schedule as enough customers are added, removed, or replaced. 100 iterations were chosen as this was a number that made simulation times feasible. With this setup the vehicle adaption configurations took between 45 minutes and an hour. The algorithm-scheduling interval simulations took between 2.5- and 4 hours to complete. In table 5 all values for the variables sheet are stated.

#### Constants

Morning slot time	00:15
Evening slot time	00:10
Start time morning	07:00
End time morning	12:00
Start time evening	17:30
End time evening	22:00
Average speed of a car	36.60
Standard deviation of car speed	0.15
Average speed of an e-bike	25.00
Standard deviation of e-bike speed	0.10
Average bike speed	13.00
Standard deviation of bike speed	0.05
Morning slot probability	0.2
Evening slot probability	0.4
Morning + evening slot probability	0.4
Wages per hour	11.65
Car costs <10km per day	0.15
car costs >10km per day	0.27
Bike costs	1.16

Table 5: Constant Variables table

### 5.1. Vehicle adaption simulations

In the vehicle adaptation simulation, I used the parameter values currently used in the company, but I changed the amount of vehicles available. The set of vehicles that can be bought consists of ebikes and cars. This set of simulations is done to determine the effect of investing in new vehicles on the existing KPI's.

Note that all parameters except the vehicles are constant, there can be no excess vehicles, and the number of morning routes is always equal to or less than the number of evening routes. In essence this means that the amount of different parameter configurations for cities is dependent on the amount of morning routes. So the vehicle slots of e-bikes and cars can take values from 0 to the amount of morning routes. Each combination of ebikes and cars is then tested per city to determine the effectiveness of both. The amount of combinations depends on how many routes a city has. Doetinchem with 1 morning route had 3 combinations, Weesp with 2 morning routes had 6 combinations. Zutphen and Cuijk both have 3 morning routes, which made for 10 combinations each city. The total combinations across all cities then amounts to 29.

### 5.2. Algorithm-scheduling simulations

In algorithm-scheduling simulations, the effectiveness of the algorithm discussed in Section 3 is tested. The scheduling intervals tested in this simulation start at 15 minutes, with 15 minute increments up to an hour. These values are chosen on the basis of 15 minutes being the current scheduling interval, and the increments being equal to the maximum service time per client. The upper bound is then chosen as it is ethical nor desired to let clients wait longer than an hour, no matter the financial/operational gain. For each city scheduling intervals were tested at 15 minutes, 30 minutes, 45 minutes, and 60 minutes. This amounts to 4 simulations per city, with 16 simulations in total.

### 5.3. Parameter extraction

In order to find the proper parameters for the Monte-Carlo simulation it is necessary to derive distributions from the existing data. This data is evaluated per city in a separate Excel file.

#### 5.3.1. Distance distribution

The distance distribution is used in the simulation to generate distances between patients. This distribution has different parameters for every city, as one city has customers closer to each other than the other. So when generating customer distances these different distribution parameters result in more accurate distance generation.

The first step is to load in all customer addresses from a city of the past year. These addresses are converted into coordinates. These coordinates can be used to gather distances between customers with the use of the BingMaps API. This API converts starting coordinates and ending coordinates into a distance expressed in kilometres. This method can then be used to generate a distance matrix which contains every distance from one customer to the other.

Subsequently, these distances were inserted into bins as to formulate a histogram. This is done to get a general sense of the frequency of each distance interval which is necessary when one wants to create and fit a distribution. The shape of the histogram illustrates the type of distribution that the data follows. This distribution is then put in its cumulative form and fit with the proper parameters using the Excel solver function. From the results it was clear that the distance distribution follows a lognormal distribution, where each city has its own mean and standard deviation.

This allows us to make a random number generator by inverting the distribution type, so that a random number between 0 and 1 is representative of the percentage point of the cumulative distribution, and returns an appropriate value for that number. A visualization of each distribution can be seen in Appendix D.2.

#### 5.3.2. Distance Asymmetry distribution

From analysing the distance matrix it was clear that it is not a symmetrical matrix (i.e.  $D(B, A) \neq D(A, B)$ ). The issue is here that these two are related in a way. If these distances were computed the usual way  $D(A, B)$  might be 5 times larger than  $D(B, A)$ , which is not realistic. In order to solve this, the distances from A to B are compared to the distances from B to A. For every city the difference between  $D(A, B)$  and  $D(B, A)$ , for every A and B, was compared in terms of percentages. This returned an average deviation, which is denoted in this section as x. This x is different for all cities. Since the deviation in distance is not dependent on the order or value of A or B, the average deviation should be 0.

Incorporation of this will be as follows:

Let x be the average deviation in percentages, and let k be a randomly generated number between 0 and 1. If  $B > A$  then  $D(B, A) = D(A, B) \cdot (1 + N\_inv(k, 0, x))$ .

This formulation allows the function to only be used once for every interchange of A and B if the combinations are iterated over A first. Take for example customers 1 and 7. When generating customers over A first,  $D(7, 1)$  will be generated without adaptation first. Which means that  $D(1, 7)$  will be adapted with a deviation as  $7 > 1$ .

The adaptation function itself transforms  $D(A, B)$  by multiplying it with a factor. This factor consists of the base value 1, as  $D(A, B) \cdot 1 = D(A, B)$ , added with the generated deviation. The generated deviation is computed with the use of randomly drawn variable k, where  $0 < k < 1$ , as well as the



computed value of  $x$  for that city. These values of  $k$  and  $x$  are then inserted into the Inverse Normal distribution function to retrieve a value for the deviation in percentages.

Let us assume that  $B=7$ ,  $A=1$ ,  $k=0.75$ , and  $x=0.01$ . Then the adaptation function would result:

$$\text{Then } D(1,7) = D(7,1) \cdot (1 + N_{inv}(0.75, 0, 0.01)) = D(7,1) \cdot (1 + 0.0674) = 1.0674 \cdot D(7,1)$$

### 5.3.3. Duration Distribution

The customer base needs to be constructed and it is of course not feasible to assume that customers remain in service forever. This duration is not dependent on eventual business growth, or scale of the company, so these data points are gathered from the beginning of the business in that city. These data points come in start dates of the service, and end dates.

These intervals are converted into days which are then sorted into bins. This allows for constructing a histogram again, allowing us to derive the distribution type from its shape. Make the distribution cumulative and fit the appropriate distribution parameters by using Excel Solver.

This distribution will be used in the Monte-Carlo Simulation to simulate customers leaving and having to enter a new one. The Duration distribution follows the exponential distribution with a different lambda for each city. A graph depicting the distributions for each city is found in Appendix D.1.

### 5.3.4. Demand Distribution & Change Distribution

The amount of customers the company has also differs per week. This means that there needs to be a function that describes the amount of customers that are in service in a given week. The data of the entire business horizon is imported. But since the company experiences large growth at the beginning of its existence it is necessary to find a point where the gradient of the trendline is close to 0. From that point to the most current data point is the time window in which to analyse customer amounts. From Figure 17 one can deduce that there is a large growth and spike at the starting weeks. The amount of customers simultaneous in service stabilizes in the latter half of the graph. Figure 18 is then the interval of which the customer amounts are analysed as it provides a more stable, and realistic version.

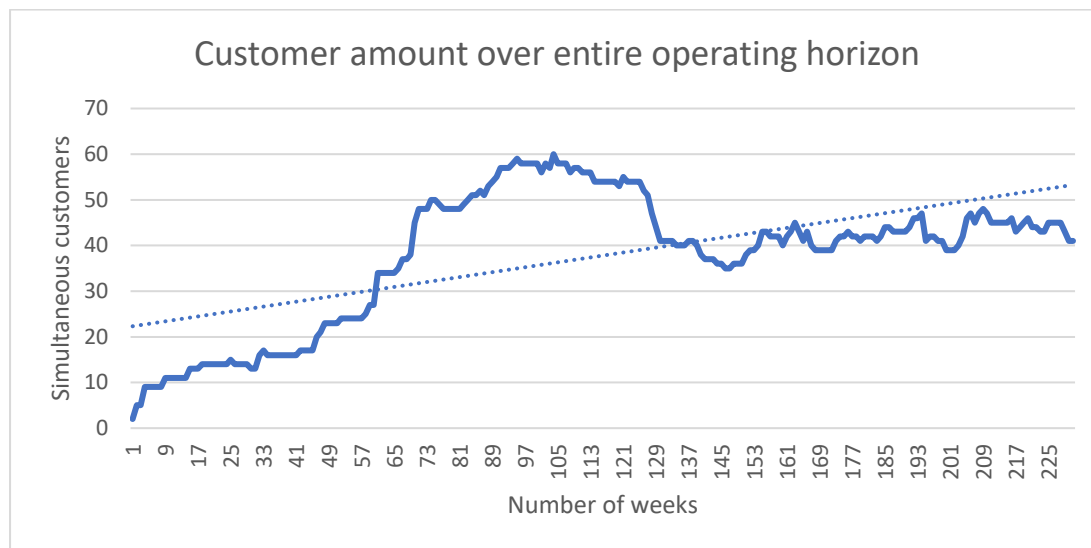


Figure 17: Customer amount visualization over the entire operating time example

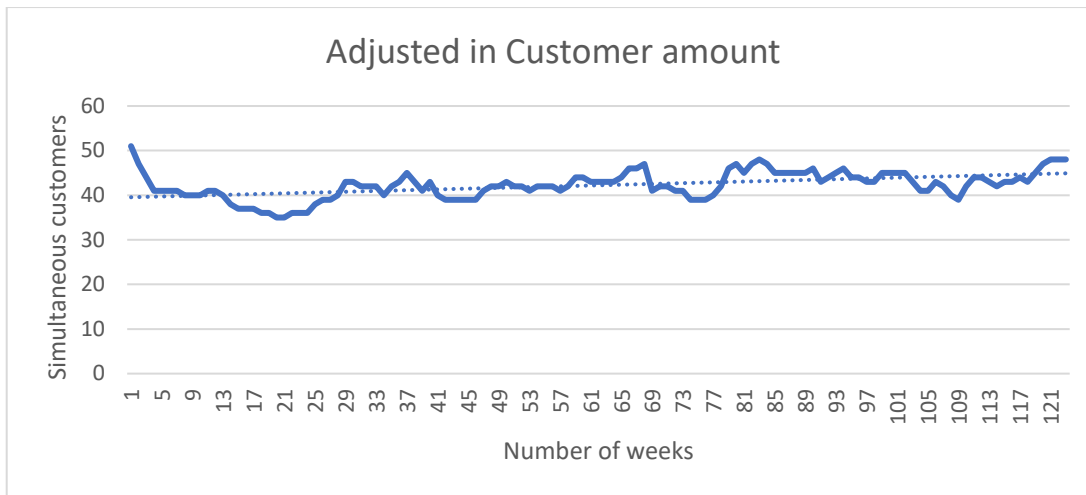


Figure 18: Customer amount visualization over adjusted period example

These customer amount numbers are then also put into bins, but the histogram is so inconsistent that it is hard to select a distribution type. Any common distribution fitted would have such an unreliable compatibility with the raw data that it would give a very warped comparison with reality. Instead, the cumulative percentages will be taken for each integer from the minimum amount of customers, to the maximum amount of customers. This results in an empirical distribution that does not follow a specific distribution type. In Figure 19 an example of this can be found. From this Figure one can see that there is no real consistent line, and an absence of probabilities for certain customer amounts. Unfortunately the empirical distribution means that there is no room to go out of the current bounds, but it represents reality most accurately. This will then be used to determine the amount of starting customers in the simulation. Again, an image of how this distribution looks per city can be viewed in Appendix D.4.

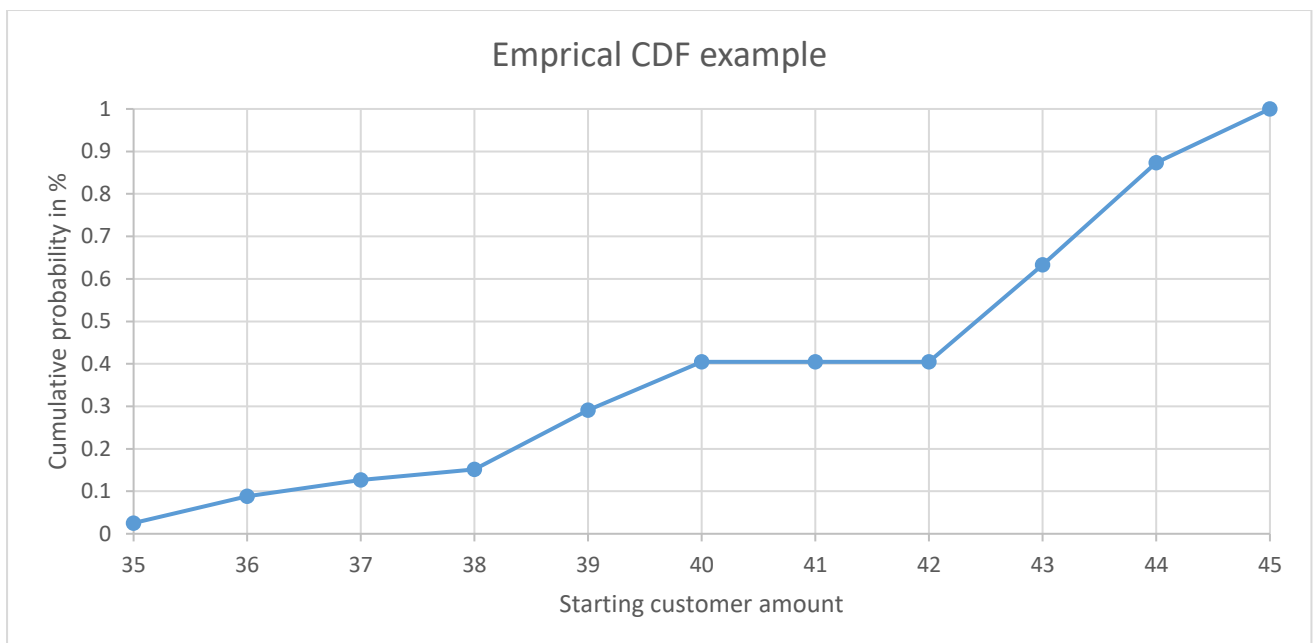


Figure 19: Empirical CDF example

Another procedure involves describing the change of customers each week, as the amount of customers changes from week to week. This is then done by taking the same data used for the Demand Distribution and look at the change in customers, instead of amount of customers. By

looking at the histogram and fitting a distribution with the *Excel Solver* it was possible to derive distinct distribution parameters for every city as is elaborated in Section 4.5.1. Figure 20 shows an example of such a distribution. The change in customers is normally distributed for each city, and by picking a random variable value between 0 and 1 it is then possible to generate random values of change according to the distribution. A generated change value rounded to the nearest integer in combination with the amount of customers in the previous week result in the amount of customers that need to be served this week. Appendix D.3 is where a graph of the change distributions per city is inserted.

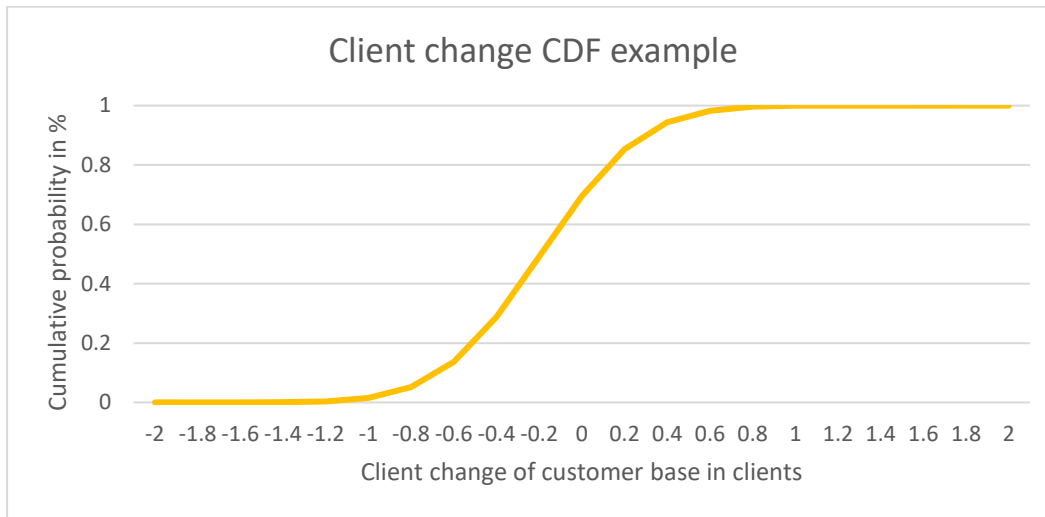


Figure 20: Client change distribution example

### 5.3.5. Slot Distribution

This information was gathered from the company manager that stated that most customers would want to be served in the middle of the route time, so 08:00 to 10:00 in the morning routes, and 18:00 to 20:00 in the evening routes. So for the distribution I made sure that 95% of the resulted slots lie within that window by computing the two tailed probability of a normal distribution with a mean and standard deviation for both the morning and the evening. Then the standard deviation was adjusted so that the two tailed probability of slots between 08:00 to 10:00 and 18:00 to 20:00 is exactly 95% . This means that the morning route has a different mean and standard deviation than the evening route. It is assumed that the slot distribution follows a normal distribution as the function should be symmetric and data near the mean should have a higher probability of occurring.

If we take the morning restriction as an example, then the mean should be 9, as that is the middle point between 8 o'clock and 10 o'clock. Let the lower bound be  $a$ , and the upper bound  $b$ . The probability of going lower than the lower bound combined with the probability of exceeding the upper bound should be 5%, as  $100-95 = 5$ . Then the standard deviation should be so that  $P(X < a) = P(X > b) = 0.025$ , since  $a$  and  $b$  have the same distance to the mean. Figure 21 shows an illustration of the left tailed probability in this scenario. Note that the standard deviation in this image is in hours. If the standard deviation in hours is converted into slots (e.g. standard deviation of 0.5, with 4 slots an hour =  $0.5 \times 4 = 2$  slots deviation) and we take the slot number that is at 9 o'clock as a mean then it is possible to construct a slot distribution using the Normal distribution as a framework.

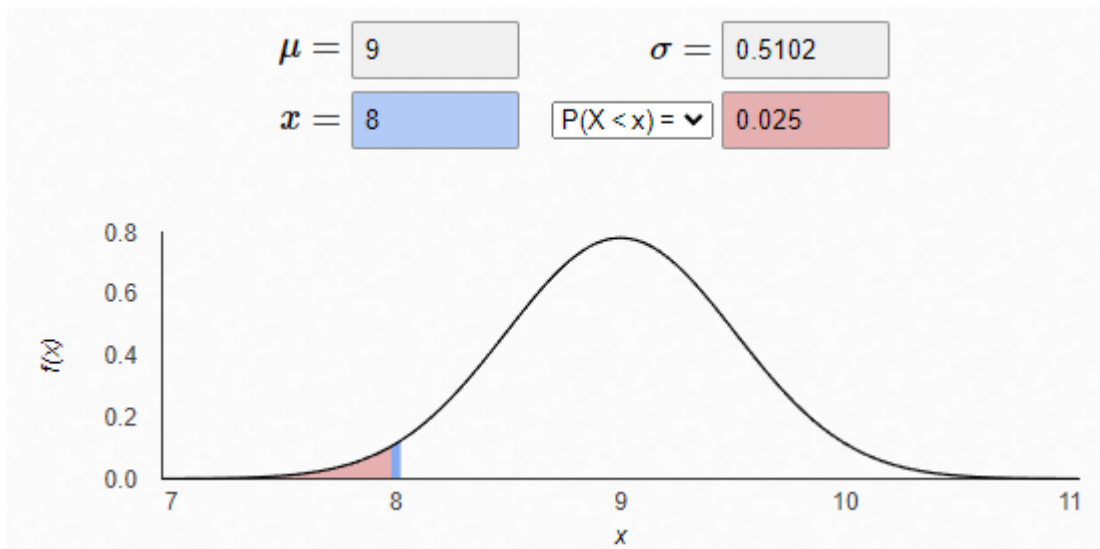


Figure 21: Normal distribution illustration

### 5.3.6. Examined cities

In this analysis 6 cities were taken under investigation, there are many more but these were particularly interesting to Betsy. These cities are Zutphen, Doetinchem, Cuijk, Ede, Weesp and Vlaardingen. Of these cities, 4 were actually subject to simulation. The city Doetinchem was elected for simulation as it was an outlier with its minimal customer base and large travel distances. Zutphen then also was simulated due to its large customer pool and busy schedules. Cuijk is also an interesting city with short customer service durations and small travel distances and was therefore also chosen for simulation. The last candidate picked was Weesp, as this was a middle ground on almost all factors between Doetinchem and Cuijk. Vlaardingen fell off the list because the city was too similar to Weesp, and leaned more towards Cuijk in factors such as customer amount. Ede also was not simulated as its values of various factors were too close to Cuijk and Zutphen to be worthwhile. The elaborate graphs of factors can be found in Appendix D.

Another argument is the amount of routes used per city. It would be best to diversify the route amounts for simulation to see the different effects of them. In the table below one can see the different routes per city. The lowest amount of routes are found in Doetinchem, whilst the highest amount lies in Zutphen. Weesp then has a route amount that lies somewhere in between with 2 routes each. Cuijk and Ede do provide the same amount of routes, but Cuijk was chosen on the basis of arguments in the previous paragraphs.

Cities and their routes	Morning routes	Evening routes
Zutphen	3	4
Doetinchem	1	1
Cuijk	3	3
Ede	3	3
Vlaardingen	3	2
Weesp	2	2

## 6. Results

From the simulation results, it is apparent that there are many differences between cities. Weesp, for example, has a starting average cost of €7,53, while Cuijk starts with an average cost of below €6.

This means that some results are subject to high variances. Another externality is that Weesp is the only city tested that has 2 morning routes, resulting in the fact that the “½ fractions” of vehicles is only populated by that city, which is an outlier compared to other cities. Appendix G provides the full unaltered results of all simulations.

### 6.1. Vehicle adaptation results

When looking at the average cost it is apparent that the composition of vehicles is definitely important. From the graph in Figure 22 it can be deduced that the benefits of one extra car does not necessarily outweigh those of one extra ebike. The average cost difference between the amount of ebikes and cars is a maximum increase of 0.3%, and could be considered negligible. This is a remarkable outcome because ebike costs are calculated on a daily basis, while car costs are based on travelled kilometres.

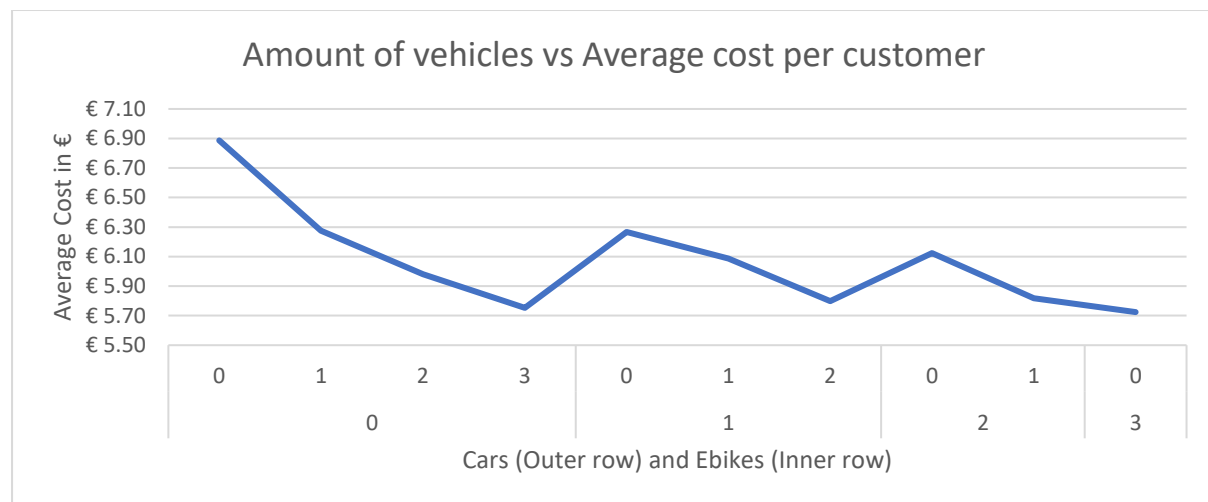


Figure 22: Vehicle amount vs Average cost graph

Punctuality wise, it is clear that without any extra vehicles there is staggering amount of employees arriving late. Figure 23 illustrates the effect of extra vehicles well. With only bike travel the punctuality lies at an average of 46%, an extra ebike would bump that up to 59%, whilst an extra car would result in 67%. What is striking is that in the case of both ebikes and cars is that the average punctuality increase from 1 to 2 vehicles is smaller than the increase when going from 2 to 3 vehicles. My explanation for this would be that the city of Weesp is not accounted for when considering 3 vehicles. Weesp has a higher average cost per customer than the rest of the cities that use 3 vehicles, which means its exclusion would result in a lower average cost per customer.

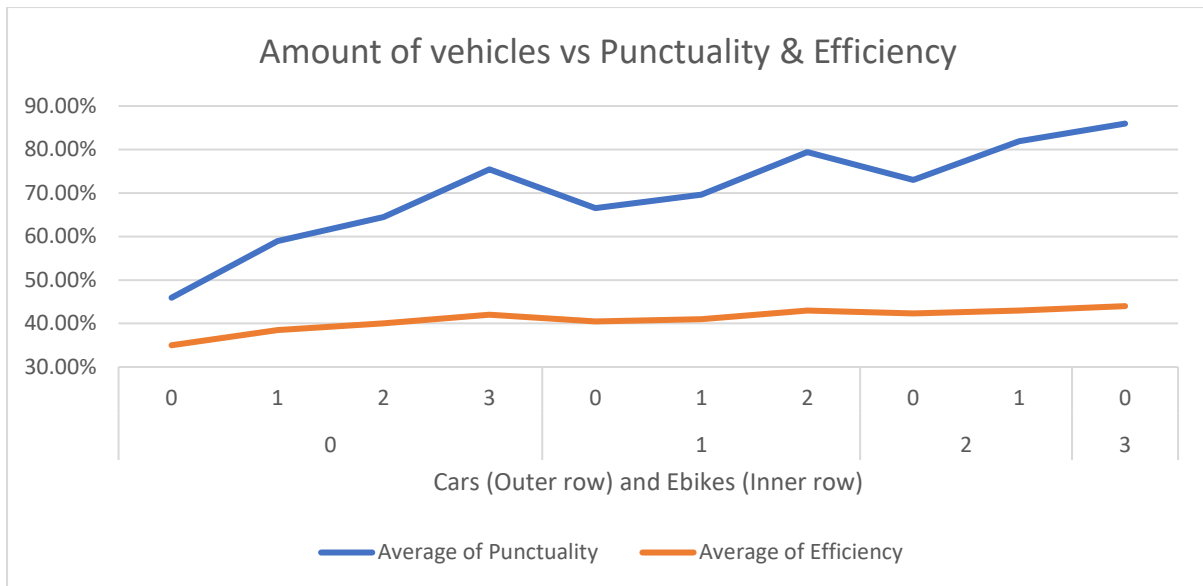


Figure 23: Vehicle amount vs punctuality & efficiency graph

When looking at the average late time and average waiting time there is a clear correlation between the two. The gradients of both variables regarding the amount of vehicles, as well as the fraction of vehicles is almost identical, as can be seen in Figure 24. Also here the first vehicle provides a significant change to both waiting time and late time. The first added ebike saves an average of 13 minutes per customer in waiting time, and 12 minutes in late time. The first car then saves 18 minutes in waiting time and 15 minutes in late time. Gradually the marginal effect of adding vehicles decreases to the point where an extra vehicle saves just about 5 minutes in waiting time, and 4 minutes late time.

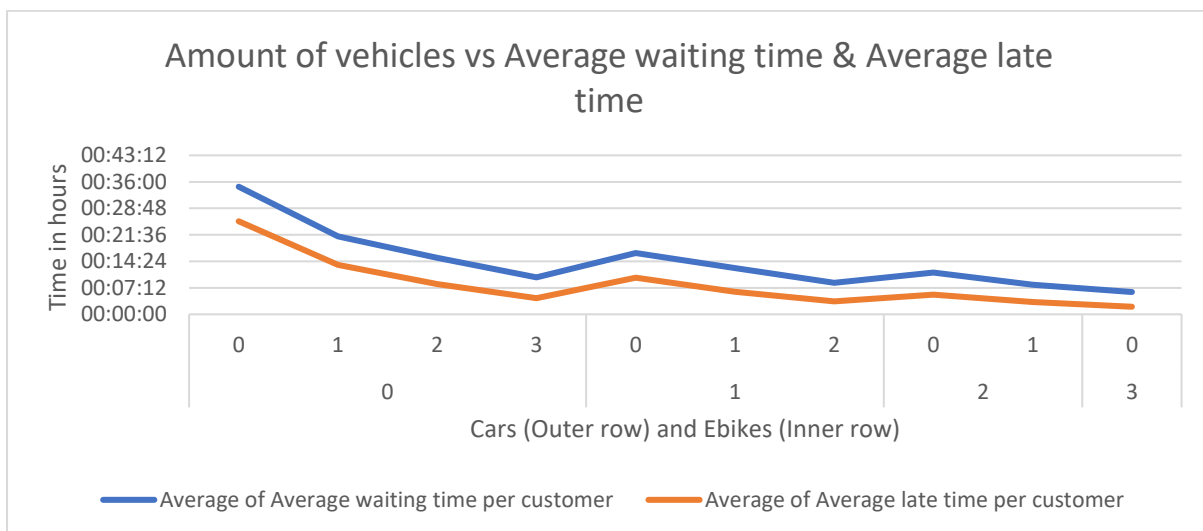


Figure 24: Vehicle amount vs Average waiting time & Average late time graph

The non-service time is special in the sense that it remains lower on average when using ebikes instead of cars. From figure 25 it can be deduced that the amount of driven kilometres remains somewhat constant, but the non-service time is subject to change. When looking at the amount of vehicles it is apparent that when interchanging the amount of cars and ebikes, the composition with the most ebikes has a lower percentage of non-service time compared to the baseline. The vehicles

per city graph in appendix E.4 also shows that there is a wide disparity between cities concerning both service time and driven kilometres.

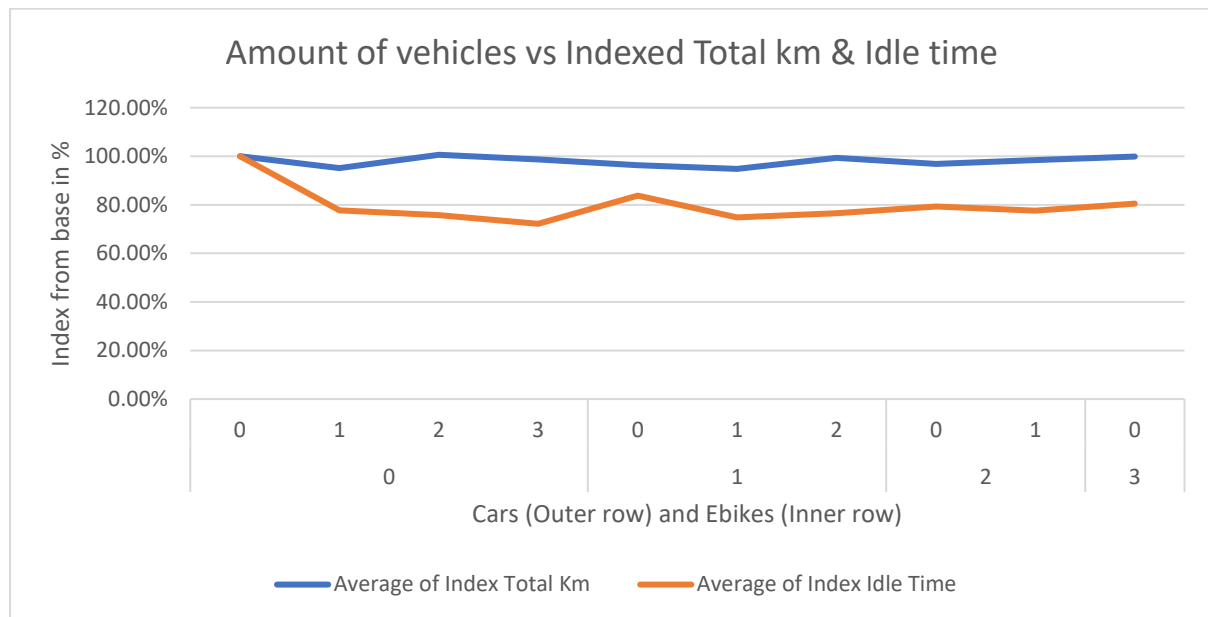


Figure 25: Vehicle amount vs Indexed Total km & Indexed Idle time graph

The average customer cost dissection reveals that labour waste is the prime factor for the variation of average cost. Labour waste is here defined as billed working time that is not spent by servicing clients, but filled in by travel, waiting etc. Figure 26, illustrates this as the first contains large spikes only when looking at the Labour waste. The average cost of labour then remains relatively the same among all compositions, and the average travel cost is then dependent on the amount of routes and cars. It is also apparent that cars contribute extra to the average cost when looking at indexed costs. Here each car adds to the share of average costs. Ebikes, however, do not contribute extra to the travel cost share of the total average cost. When 1 or more cars are present, additional ebikes even decrease the travel cost share.

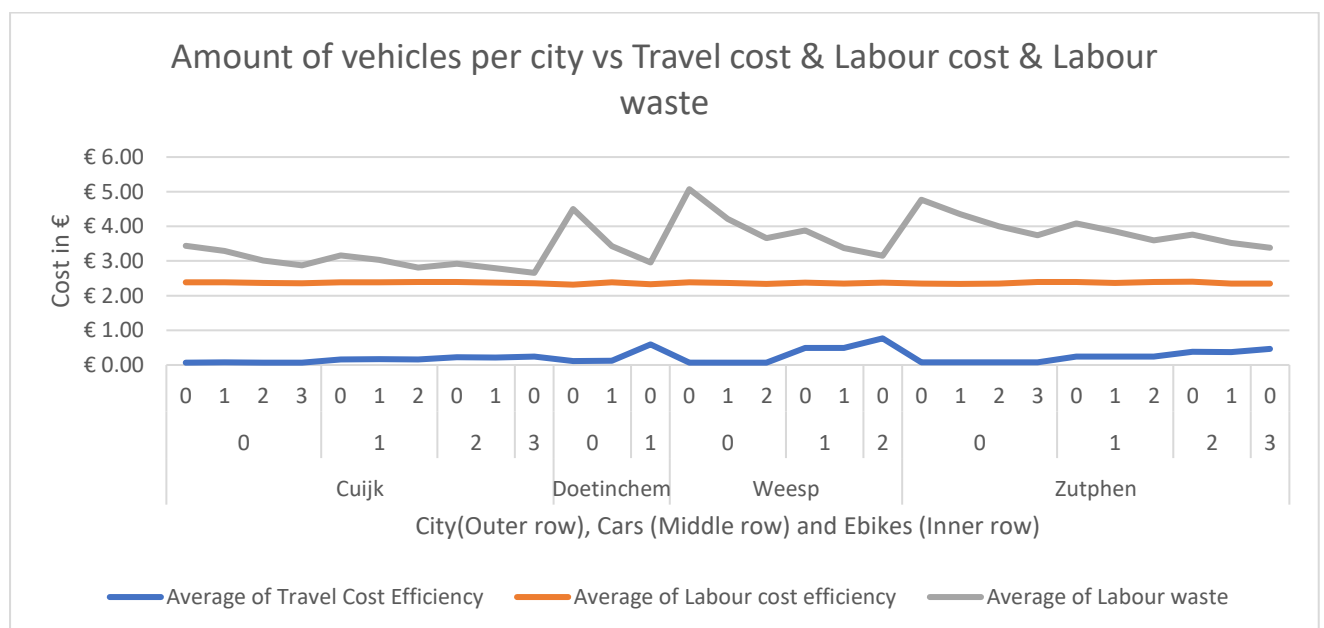


Figure 26: Vehicle amount vs cost share graph

In the table below all standard error intervals for the performance indicators are listed. These values are drawn from the “amount of vehicles” analysis. The numbers listed in table 6 tell us that these standard errors are negligible, as the customer won’t even notice an employee arriving 9 seconds late, for example,

Performance Indicators vs Standard Error	Minimum Standard Error	Maximum Standard Error
Average cost per customer	€0.02	€0.05
Average Punctuality	0.72%-point	1.30%-point
Average Efficiency	0.16%-point	0.30%-point
Average waiting time per customer	00:00:29 (29 seconds)	00:00:30 (30 seconds)
Average late time per customer	00:00:09 (9 seconds)	00:00:41 (41 seconds)
Average of Index Idle time	0.13%-point	0.45%-point
Average of Index total km	0.11%-point	0.25%-point

Table 6: Standard error table for the Vehicle-Adaption simulations

## 6.2. Algorithm-scheduling results

From the simulations it became clear that punctuality is always increasing when the scheduling interval increases. This is likely due to the constraint in the algorithm that states no time window may be exceeded in combination with the 15 minutes extra the employees get to arrive on time. Peculiar, however, is the development of the efficiency as the scheduling interval progresses. There seems to be a diminishing decrease in efficiency at first, but when the scheduling interval changes from 45 minutes to an hour there is a decrease in efficiency. The most plausible explanation is that the extra time space allows for more punctual decisions, which means employees are driving longer distances overall to be on time more.

Strange phenomena also occur when looking at the average waiting time and late time, which are detailed in Figure 27. For the late time one would expect that an extra increase of 15 minutes of arrival window would result in at least 15 minutes of reduced average late time. Instead, each 15 minutes extra result in lower returns in late time that get progressively worse. However, these little decreases add up and can transform the average late time per customer from 25 minutes to a little over 5 minutes. Average waiting time then follows the opposite trajectory. The changes to it are small compared to late time, but its decrease is growing when the scheduling interval is decreased. Also here the marginal gains are small, but adding all that up results in a 10 minute decrease in average waiting time when the scheduling interval goes from 15 minutes to 60. It also follows from this that the average late time is more volatile than the average waiting time.



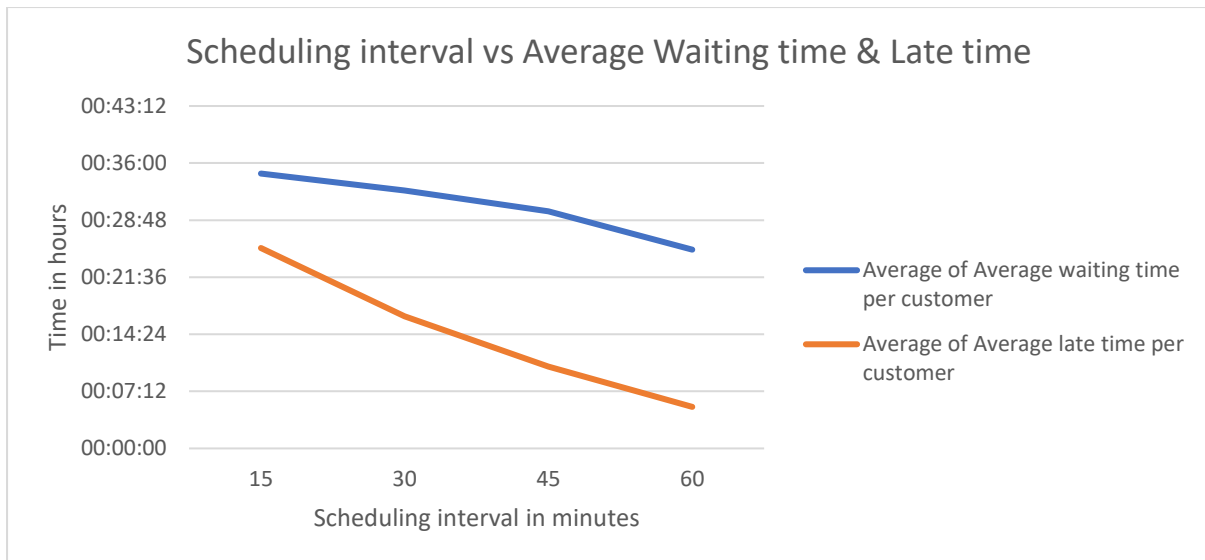


Figure 27: Scheduling interval vs Average waiting time & Average late time graph

When examining the average cost per customer the change follows a similar pattern. The first 15 minute increment has the largest decrease in average cost, which slowly subsides as the scheduling interval gets larger. The scale of the change in cost can be substantial. The first increment on average skims of 89 cents per service, which might not seem like a lot but that is almost a 13% decrease. At 60 minutes, a decrease of 1,5 euros is on the cards on average per customer.

The distance is different from the other variables in the sense that the gradient of decrease grows as the scheduling interval increases. This, naturally, has to do with the used algorithm and the consequences thereof. When the interval becomes larger, it is more likely that the used route lies closer to the optimal route, due to there being less constraints.

The non-service time follows a much more linear decrease, as can be seen in Figure 28 and Figure 29. This means that the shortage in non-service time between a scheduling interval from 30 to 45 minutes is almost equal to the difference between 45 to 60 minutes. It is here then clearly apparent that the current scheduling interval allows for much improvement.

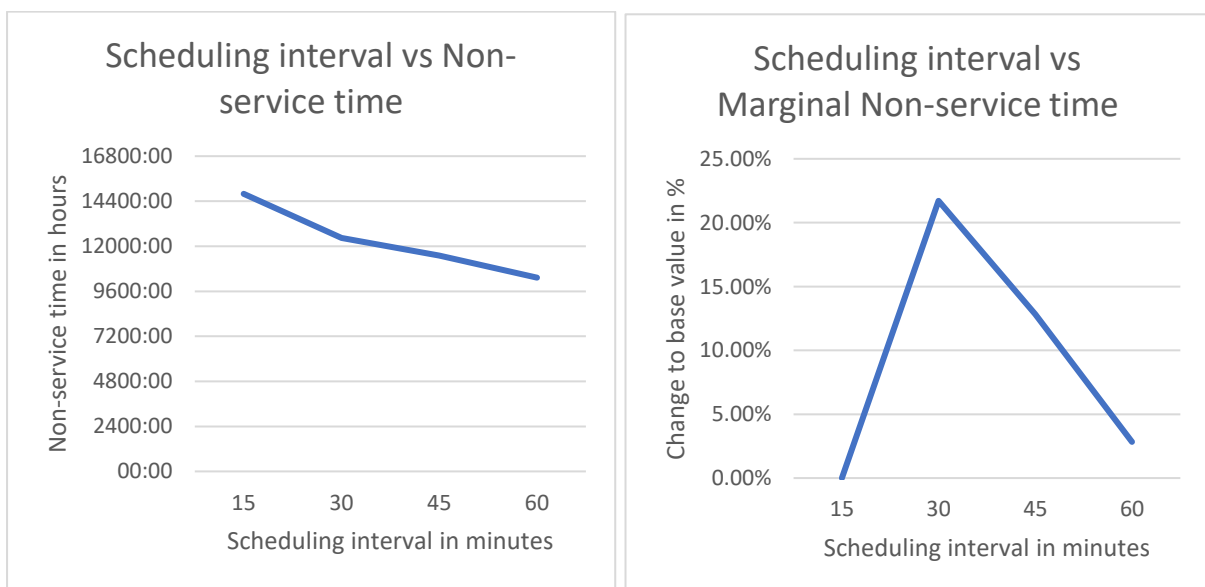


Figure 28 & 29: Scheduling interval vs Non-service time graph & Scheduling interval vs Marginal non service time graph

Table 7 tells us the same story that table 6 does. These standard errors found from analysis are not large enough to factor into the conclusion. Consequently, the number of iterations used in the Monte-Carlo simulation is sufficient to provide reliable estimates.

Performance Indicators vs Standard Error	Minimum Standard Error	Maximum Standard Error
Average cost per customer	€0.03	€0.07
Average Punctuality	0.56%-point	0.76%-point
Average Efficiency	0.18%-point	0.84%-point
Average waiting time per customer	00:00:40 (40 seconds)	00:00:46 (46 seconds)
Average late time per customer	00:00:015 (15 seconds)	00:00:41 (41 seconds)
Average of Marginal Idle time	0.53%-point	0.93%-point
Average of Marginal total km	0.17%-point	0.25%-point

*Table 7: Standard error table for Scheduling-algorithm simulations*

## 7. Discussion and conclusion

### 7.1. Discussion

From both the vehicle adaptation simulations and the algorithm-scheduling simulations it is clear that there is a strong presence of diminishing marginal gains for almost every variable. This means that the first addition, it being a vehicle or an added scheduling interval, has the largest positive effect for the company. From this it is then certain that the current situation is not the optimal state, and the company should be looking to implement at least one change.

The cities tested had a range of one possible vehicle to three, but the amount differs per specific city. This is why in the recommendations I will distinct cities with one vehicle, two vehicles, and three vehicles.

For one vehicle cities it would be wise to adopt a car. This can be concluded from the analysis of both the general vehicle tendencies of all simulations, as well as the results of Doetinchem, which is the only city tested with one vehicle. General tendencies show that cars do provide the largest benefit initially compared to ebikes. This is especially visible in punctuality and average waiting time. From Doetinchem specifically the cars provide a noticeably larger Punctuality and Efficiency than the ebike. As well as only 65% of the base idle time, compared to the 71% of the ebike.

For two vehicle cities a strategy of one car and one ebike would be ideal. This remark is based on the general behaviour of vehicle change, as well as the case of Weesp, which is the only city that operates with two vehicles. This would be so because the effect of the first car is major, as is with the one vehicle cities. The main difference here is that the added benefits of a second car would not offset the investment required. An ebike also gives extra benefits but at a much lower cost initially as in the long run. On other indicators, the option with two cars performs slightly better, yet not enough to justify extra spending.

Three vehicle cities face the same situation as the two vehicle cities. One car brings significant benefits, but the extra ones do not justify the extra cost. This why also here a configuration with one car and the remainder ebikes works best. The second ebike is also worth purchasing as operating by bike brings the overall average performance down significantly.

Generally, it is most profitable for a branch to adopt a single car, and then fill the remaining route vehicles with ebikes.

For the scheduling interval a more uniform recommendation can be given. From the results it is apparent that the routing algorithm in combination with the increased scheduling interval results in significant performance increases. On the contrary, a scheduling interval of 60 minutes has too little benefits to justify the extended waiting time of a customer. This is supported by the lack of change in efficiency, as well as average cost per customer. Therefore a decision has to be made on whether to use the 30 minute scheduling interval or the 45 minute scheduling interval.

The difference between 30 minutes and 45 minutes scheduling interval is that the 30 minute version has larger marginal increases on important fronts such as average cost and non-service time. Even in areas where 30 minutes does not provide the maximum marginal increase, it does not follow far from its progressed counter parts. Personally I would recommend to use a scheduling interval of 45 minutes. This is so because the 45 minute option encompasses the 30 minutes scheduling interval benefits and adds substantial performance increase still. When looking at the average waiting time it is clear that the average waiting time decreases when the scheduling interval is larger.

This raises the question of whether it is ethical to let a small set of patients wait 45 minutes to decrease other people's waiting time and increase company performance. I would argue so as the added benefits are so large that there is additional room for customers in the schedule still, increasing the total utility of the community.

There is a stark contrast when looking at the current state of affairs and the most important factors of punctuality, efficiency, average waiting time and average cost per customer. Adopting the recommendations regarding the vehicles, 1 car and the rest Ebikes, would result in an average of 74.86% of punctuality, or an average increase in punctuality of 62.99%. An average of 42.75% of efficiency can then be found, which means on average an 22.14% increase in efficiency. An average waiting time of 9 minutes and 48 seconds is then found across all cities, which would mean an average decrease of 76.12%. Average cost wise the mean lies at €5.93, which is an average decrease of 14%.

Adhering to the scheduling interval and algorithm recommendations also provides benefits compared to the current operations. A scheduling interval of 45 minutes combined with the swap algorithm would result in a punctuality of 74.77% on average. This is an increase of 64.05% of times the employees arrive to the customers on time. The efficiency then averages at 49.25%, or an increase of 40.71% compared to the base value. The average waiting time with this approach gives an average value of 29 minutes and 54 seconds. This is a decrease of 13.79% when put next to the usual average waiting time. The average cost is €5.38 on average, or compared to the base value, a 22.92% decrease.

## 7.2. Conclusion

In this thesis I have built a simulation that is supposed to reflect the daily operations of Betsy. This simulation was constructed in excel with the use of Visual Basic code language. In order to complete this task, the Crisp-DM methodology cycle was used to make sure that the results and inputs are properly implemented for usage. In the end, a simulation document was produced which can simulate the daily operations of cities, for now and in the future.

From the simulation it became apparent that a scheduling interval of 45 minutes is the most efficient when looking at the possible waiting time of customers versus the overall improvement in performance.

Implementation of time in the schedule was heavily used in the routing algorithm. In here the travel time in the schedule was calculated as the distance travelled divided by the average speed of the

vehicle. The difference with a usual calculation would be here that the distance would be taken over the road, not as the distance between two points.

For the routing algorithm I used an adapted version of a greedy algorithm. This algorithm allows for time windows and different simultaneous routes. In this case it was necessary to remove some constraints used. These constraints such as maximum worktime per employee or employee qualification can be added again in the future if necessary.

The purchasing of vehicles has been proven effective for operational improvement. It seems optimal to buy one car, and fill the empty spots with ebikes. Adopting this approach will drastically improve, efficiency, punctuality, as well as a decrease in non-service time and average waiting time. This then leaves space to allow new customers to enter the system, as well as lower the average waiting time for the rest and staying within the appointment window.

### 7.3. Limitations and future work

The simulation constructed is somewhat a representation of reality, but there are some components missing. There are things such as cancellations of services, or customer division across morning and evening, which are present in real life and do have an effect on daily operations, but have no data on them yet. These areas then are unexplored and could have an effect on the scope of the effectiveness of certain measures, or could alter the conclusion altogether. When going forward it might then be wise to collect data on those topics that are relevant, in order to adapt the simulation and get more reliable results.

Furthermore there are few cities tested. This might mean that the conclusions drawn are not perfectly reliable for all cities which Betsy operates in. In the future, the simulation could be extended to incorporate more, or even all cities to determine a uniform strategy.

Lastly, the results of the Vehicle-Adaptation simulations and the Algorithm-Scheduling simulations might interfere. It would be possible that the increase in cars and ebikes are already effective enough that there is no need anymore to use an algorithm and increase the scheduling interval, or vice versa. In order to progress research it would be an idea to combine these two simulation types, and check whether there is a correlation between the two.

## 8. Appendix

### Appendix A: Systematic literature review

#### 1. definition of the knowledge problem/research question

The knowledge question I will be evaluating in the systematic literature review is:

*What is the best routing algorithm to use?*

#### 2. defining the inclusion and exclusion criteria (and why)

In order for an article to be relevant here, it has to be set in a healthcare surrounding. This is so because many business articles regarding logistics are only out on profit. On the contrary, healthcare logistics often involve human considerations and the maximization of profit is secondary to the wellbeing of patients and employees alike.

#### 3. defining the databases used (and why)

The databases used are Google Scholar and Scopus. Google Scholar I will mainly used to find well known article. Since I already searched a lot on Google about this specific assignment, their

search algorithm spits out more appropriate articles. Scopus I use for in depth, structured search on a subject. The Scopus search parameters are much more refined and help me find articles that are new, but relevant (as opposed to Google Scholar).

#### 4. describing the search terms and the used strategy

For searching articles there are some factors that need to be included such as: HHCRSP, HHC, VRP and Time-window. HHCRSP is useful because it is a scientific used abbreviation that describes the problem at hand. VRP is also effective as most HHCRSP can be, or are, described as VRP's. The phrase Time-window is necessary as it is an important constraint to our problem. The last search term is Algorithm, as I am interested in how to use the most efficient routing, otherwise I will just get a description of the problem, which is not necessarily helpful. Combining these search terms with additional constraints, or each other, will give plenty academic papers surrounding healthcare logistics. In these papers there are often methods mentioned on how to divide patients, or schedule them. It is then necessary to read the abstract first to determine whether the research is narrow enough for my assignment, and then proceed to assess the routing algorithm described.

#### 5. listing the number of search results found, the number of duplicates, to the final set of articles

Input factors SCOPUS	Document result
HHCRSP	19
HHCRSP + Algorithm	12
HHCRSP + Time Window	9
HHCRSP + VRP	3 – 2 of which are duplicates of above query
HHCRSP + VRP + Algorithm+ Time Window	1 - Duplicate

The input of solely HHCRSP is too broad for this research. HHCRSP + Algorithm does not fully reflect all constraints of this research. HHCRSP + Time Window results in some proper articles related to the research, although some insert extra constraints that are unnecessary. HHCRSP+VRP also returns some valid articles, which are centered around the field of Operations Research. Combining all factors results in one article, which might as well be the most important one.

Of the HHCRSP + Time Window input, 1 article includes perishable goods, which is not applicable to this research. Another article talks about synchronized visits and lunch breaks, which are heavy constraints and do not occur in my research. One other article includes multiple availability periods for patients, which is here also not the case. Another report is focusing on large scale operations, which considering we only have 35 patients or so, is not applicable here. This then leaves 5 articles from the particular input query that are usable.

The one unique article from HHCRSP + VRP is a mere description of how such a problem should be formulated, not any information on how to solve that problem.

So the final set of articles = 5 articles

#### 6. a conceptual matrix; listing the articles with their core topics or findings, organized along concepts

	Solution method type
Gossip Algorithm	A Gossip Algorithm for Home Healthcare Scheduling and Routing Problems (Riazi,

	Chehrazi, Wigstrom, Bengtsson, & Lennartson, 2014)
Greedy algorithm	Continuity of care in home services: a client-centered heuristic for the home health care routing and scheduling problem (Martinez, Espinouse, & Di Mascolo)
Gossip CG algorithm	Decomposition and distributed algorithms for home healthcare routing and scheduling problem (Riazi, Wigstrom, Bengtsson, & Lennartson, 2017)
Pure Linear programming	Multiple Service Home Health Care Routing and Scheduling Problem: A Mathematical Model (Dengiz, Atalay, & Altiparmak, Multiple Service Home Health Care Routing and Scheduling Problem: A Mathematical Model, 2019)
Tabu Search	Novel Routing-Scheduling Problem for Home Health Care Network (Dengiz, Atalay, & Altiparmak, Novel Routing-Scheduling Problem for Home Health Care Network, 2021)

**7. integration of the theory, also organized around concepts (not just a summary of the articles). We are also looking for what you find as an answer to your knowledge problem.**

All articles agree that the solution should involve some sort of Mixed Integer Linear Programming. The Gossip Algorithm does work to a reasonable extent according to the article, but only works optimally if there is already a feasible solution, which is not necessarily the case in this research. This exact issue occurs at the Gossip CG Algorithm, where this method is optimized for large problems, and could require other algorithms for subproblems. The pure linear programming approach is only viable if there are many constraints taken into account, which then brings with it a longer computation time. This is not the algorithm I plan on using as the amount of constraints is relatively little to what is used in the study, and there are quicker ways of solving a problem. The Tabu Search algorithm looks to be a good fit, as it circumvents local extremes, and iterates even worse solution to eventually get to the optimal solution. The only problem here is that there needs to be a lot of programming done with extensions that are not available in Excel/VBA. The Greedy algorithm looks the proper algorithm to use as its computations are relatively simple, the constraint programming not too complex. The result of the Greedy algorithm will not be as optimized as, for example, a Tabu Search, but a small margin of error is acceptable.

Therefore the answer to my knowledge question “*What is the best routing algorithm to use?*” is the Greedy Algorithm

## Appendix B: Check on good research

Characteristics of Research	What a Manager Should Look For in Research Done by Others or Include in Self-Directed Research	Chapter
Purpose clearly defined	<ul style="list-style-type: none"> <li>Researcher distinguishes between symptom of organization's problem, the manager's perception of the problem, and the research problem.</li> </ul>	4, 5
Research process detailed	<ul style="list-style-type: none"> <li>Researcher provides complete research proposal.</li> </ul>	4, Appendix A
Research design thoroughly planned	<ul style="list-style-type: none"> <li>Exploratory procedures are outlined with constructs defined.</li> <li>Sample unit is clearly described along with sampling methodology.</li> <li>Data collection procedures are selected and designed.</li> </ul>	3, 4, 5, 6–14
High ethical standards applied	<ul style="list-style-type: none"> <li>Safeguards are in place to protect study participants, organizations, clients, and researchers.</li> <li>Recommendations do not exceed the scope of the study.</li> <li>The study's methodology and limitations sections reflect researcher's restraint and concern for accuracy.</li> </ul>	2, 19, 20
Limitations frankly revealed	<ul style="list-style-type: none"> <li>Desired procedure is compared with actual procedure in report.</li> <li>Desired sample is compared with actual sample in the report.</li> <li>Impact on findings and conclusions is detailed.</li> </ul>	6, 14, 15, 19, 20
Adequate analysis for decision maker's needs	<ul style="list-style-type: none"> <li>Sufficiently detailed findings are tied to collection instruments.</li> </ul>	15–20
Findings presented unambiguously	<ul style="list-style-type: none"> <li>Findings are clearly presented in words, tables, and graphs.</li> <li>Findings are logically organized to facilitate reaching a decision about the manager's problem.</li> <li>Executive summary of conclusions is outlined.</li> <li>Detailed table of contents is tied to the conclusions and findings presentation.</li> </ul>	15–20
Conclusions justified	<ul style="list-style-type: none"> <li>Decision-based conclusions are matched with detailed findings.</li> </ul>	15–20
Researcher's experience reflected	<ul style="list-style-type: none"> <li>Researcher provides experience/credentials with report.</li> </ul>	19, 20

Figure 29: Check on good research table (Cooper & Schindler, 2014)

## Appendix C: CrispDM cycle

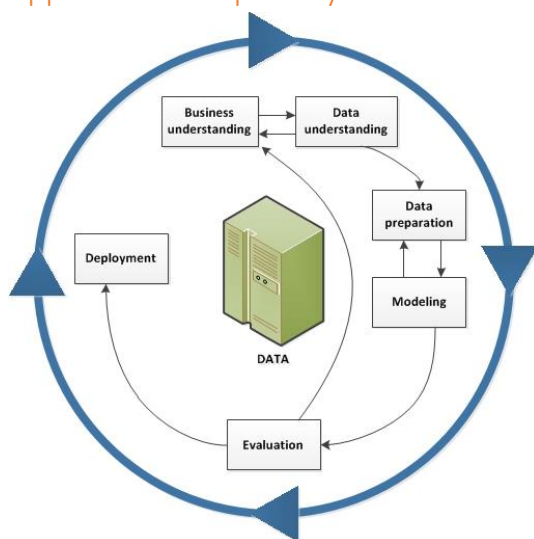
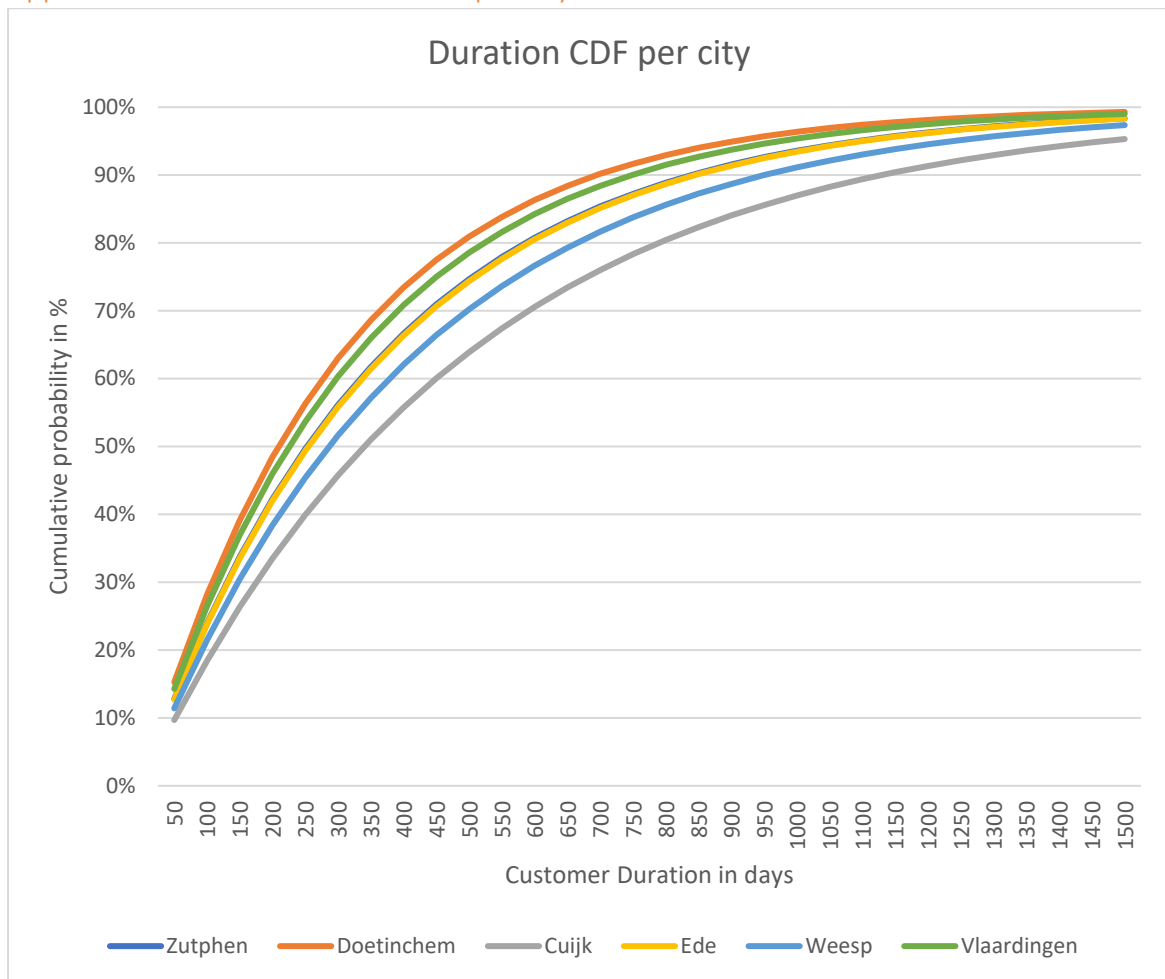


Figure 30: : Crisp-DM cycle (IBM, 2021)

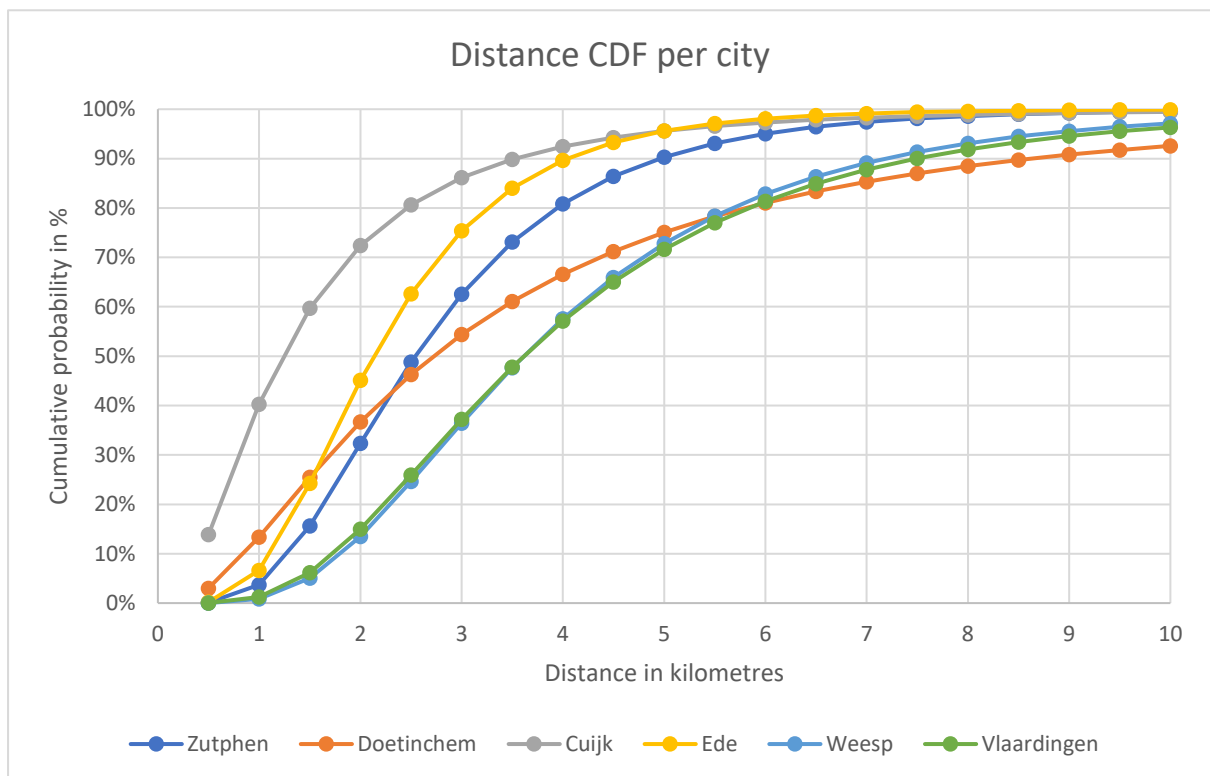
## Appendix D: City distribution comparison

### Appendix D.1: Duration Distribution per city

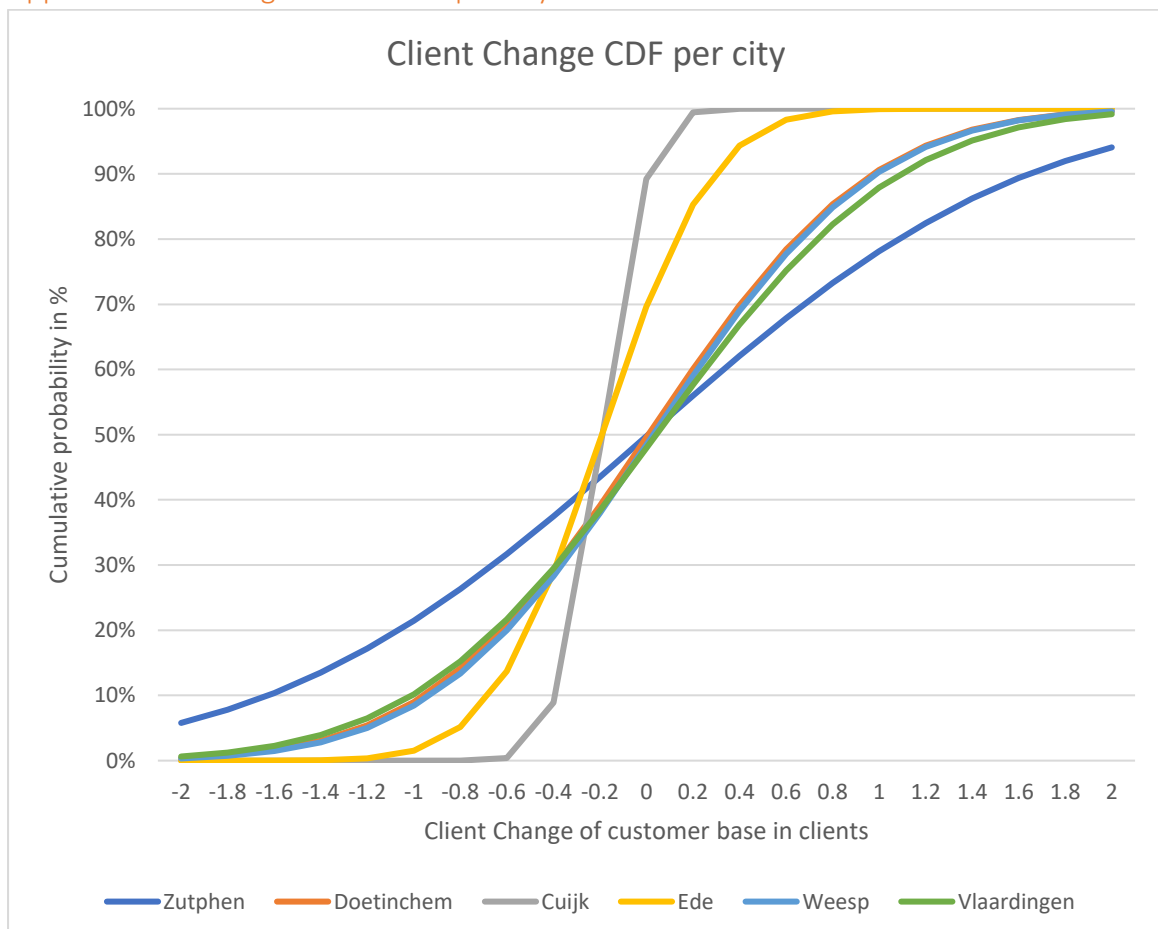




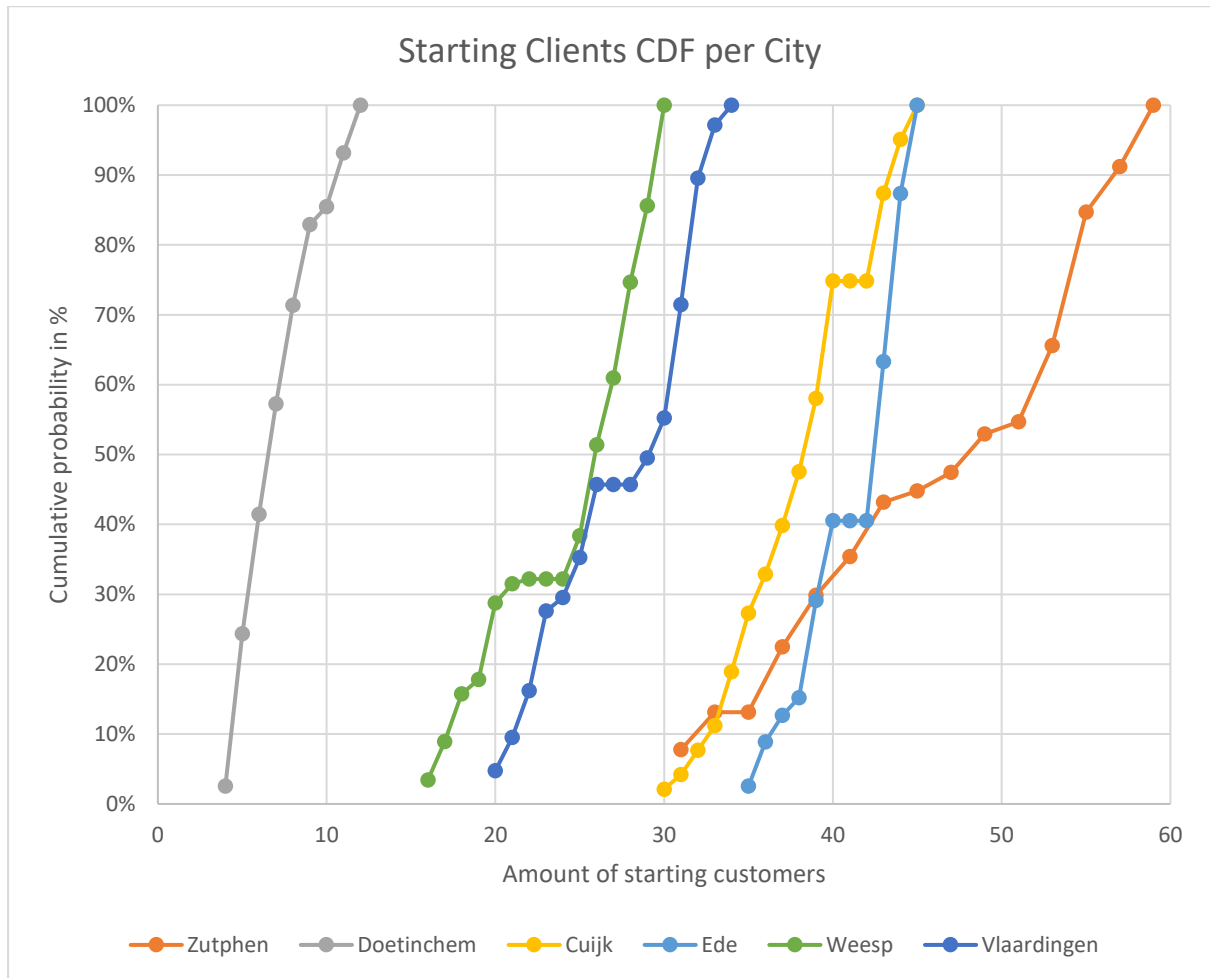
## Appendix D.2: Distance Distribution per city



## Appendix D.3: Change distribution per city

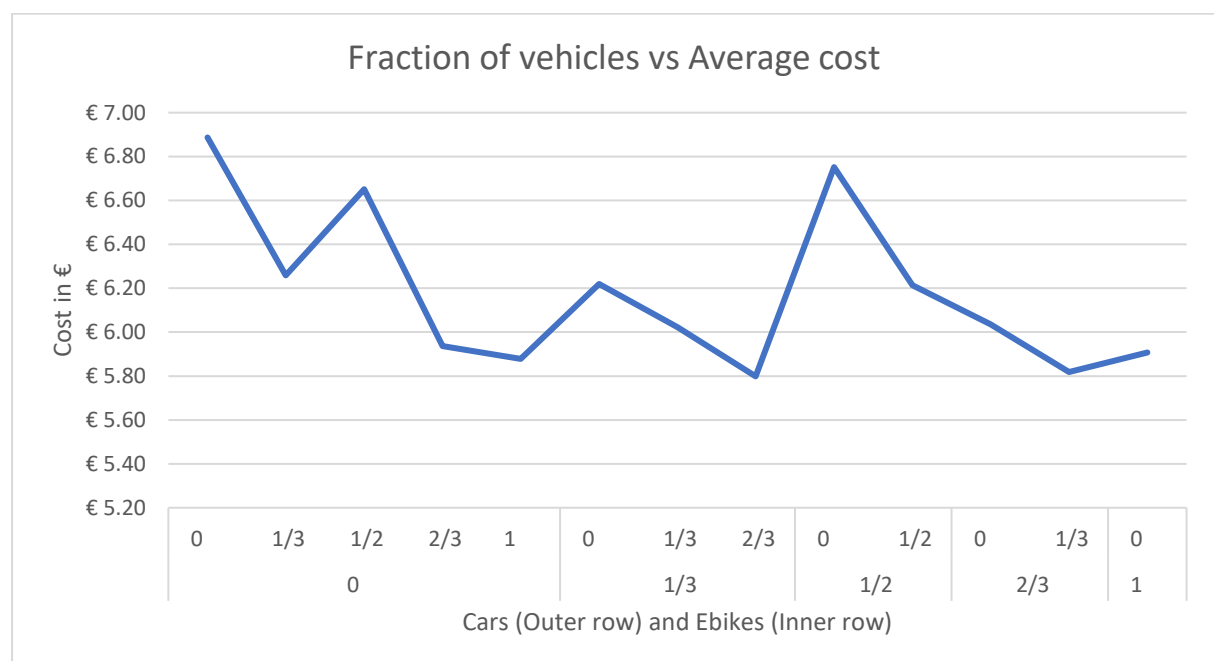
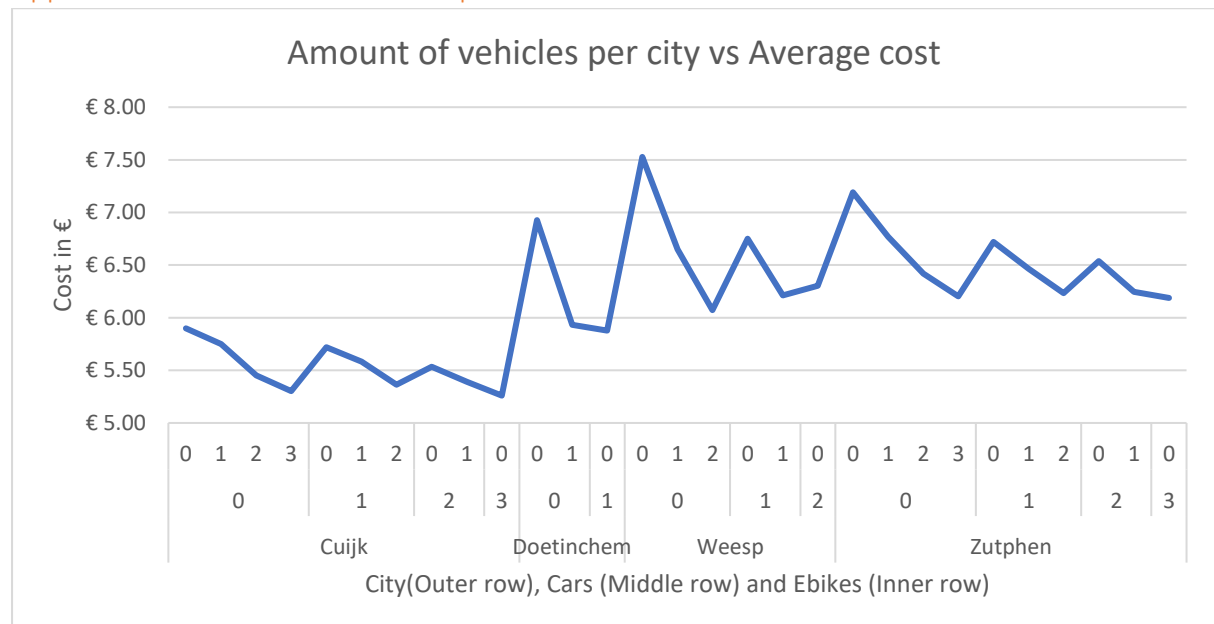


#### Appendix D.4: Nr. of Clients distribution per city

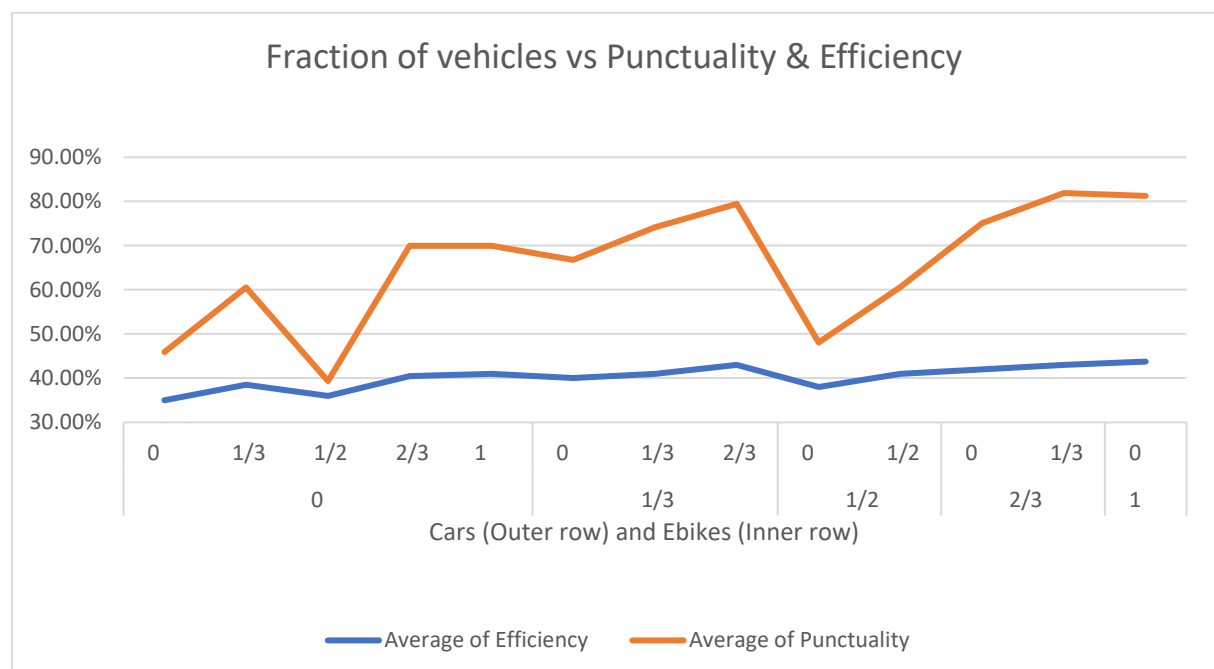
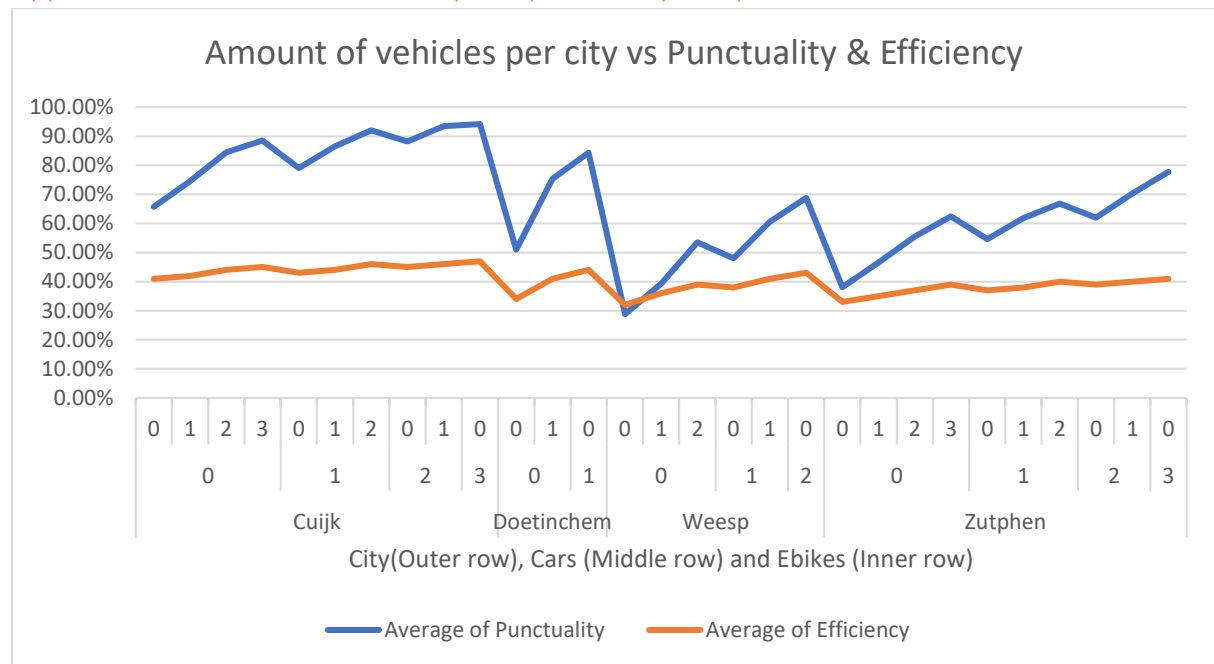


## Appendix E: Vehicle adaption simulation graphs

### Appendix E.1: Vehicles vs cost comparison

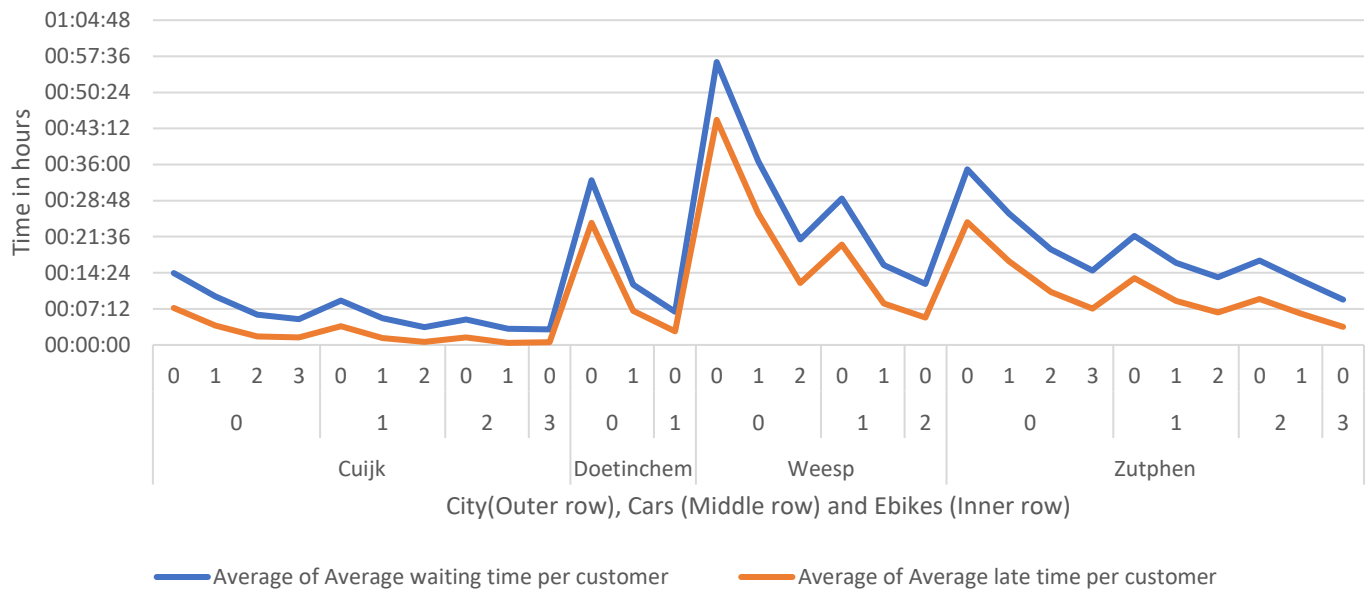


## Appendix E.2: Vehicles vs efficiency and punctuality comparison

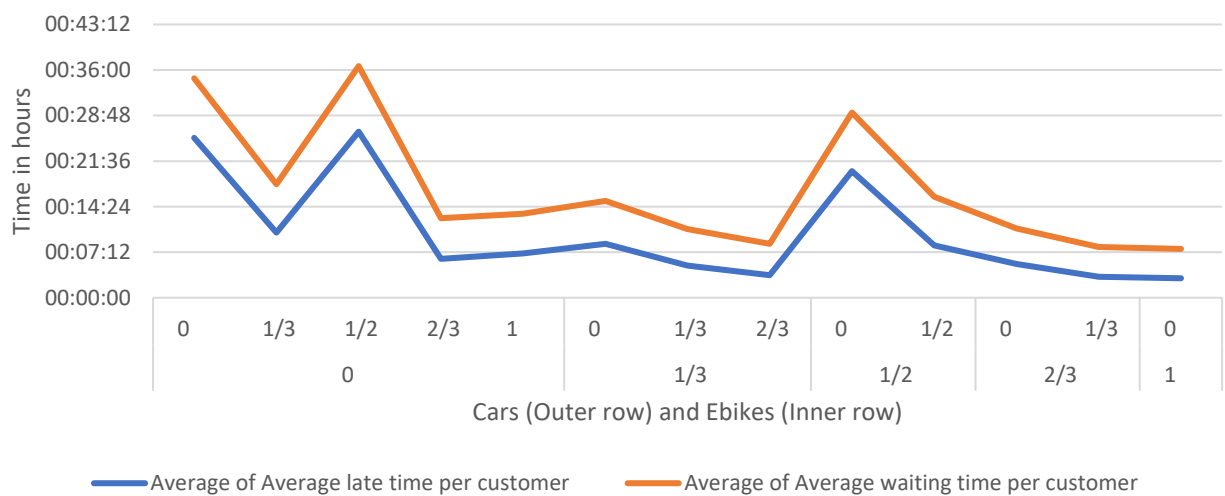


### Appendix E.3: Vehicles vs waiting time and late time comparison

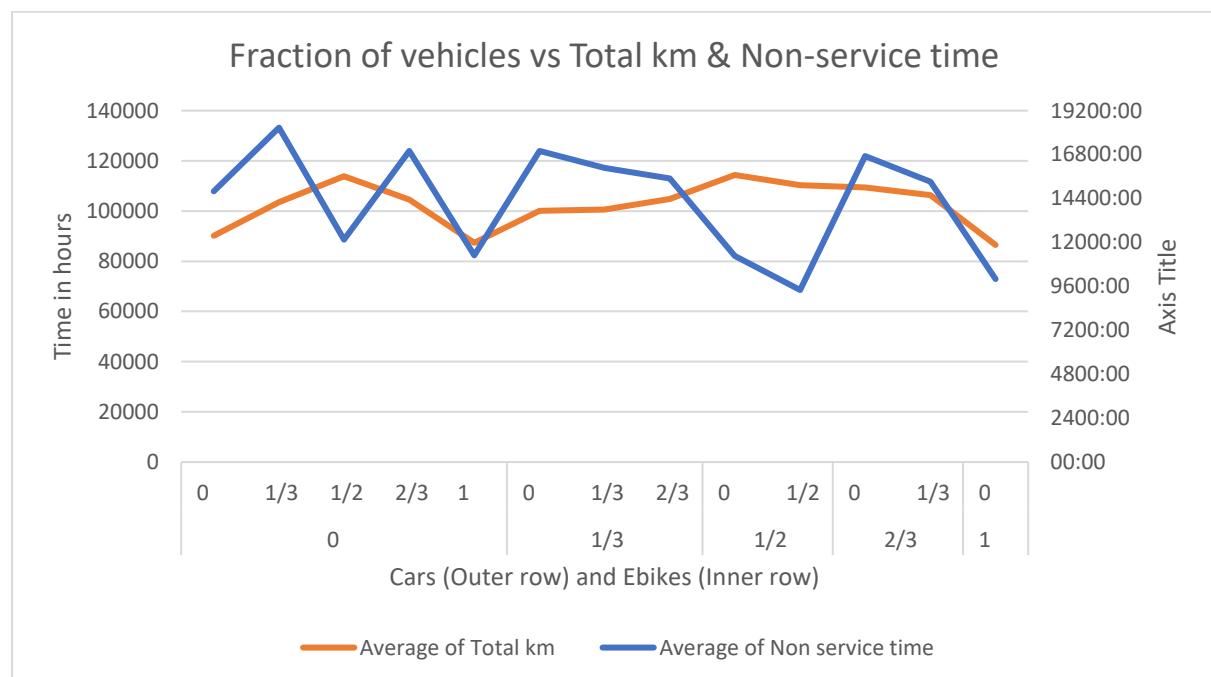
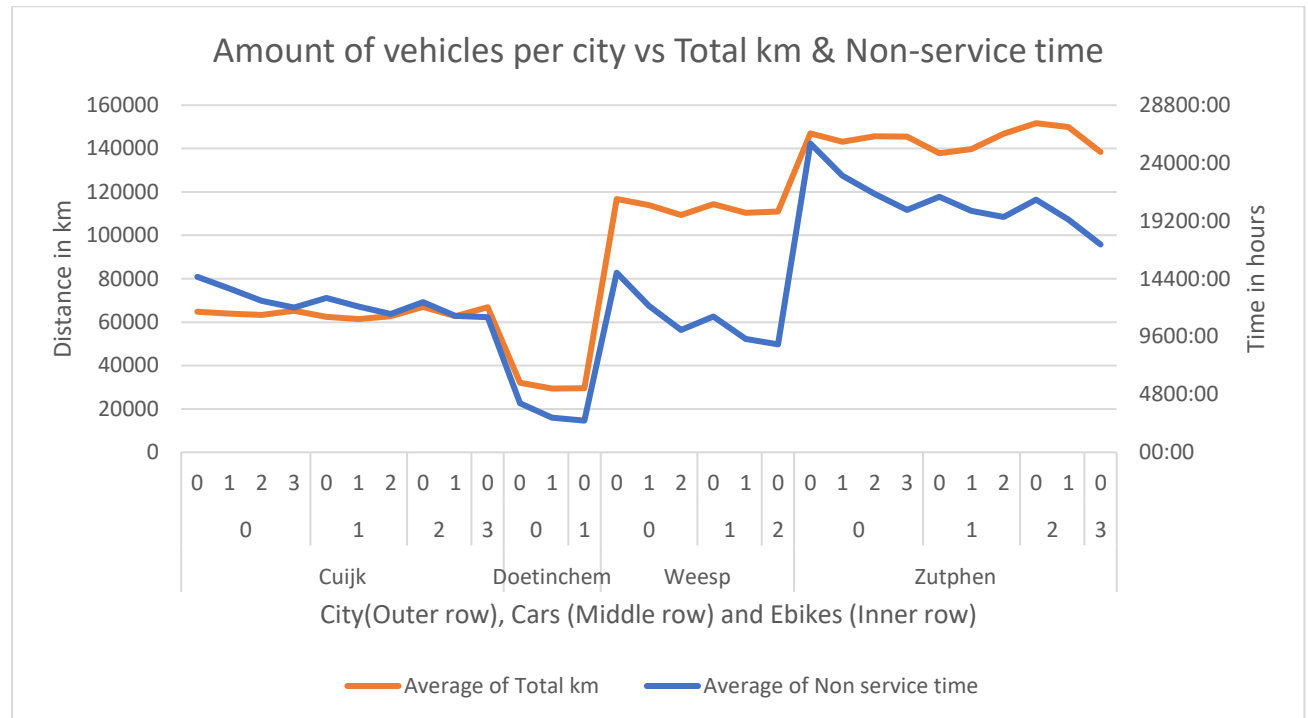
Amount of vehicles per city vs Average waiting time & Average late time

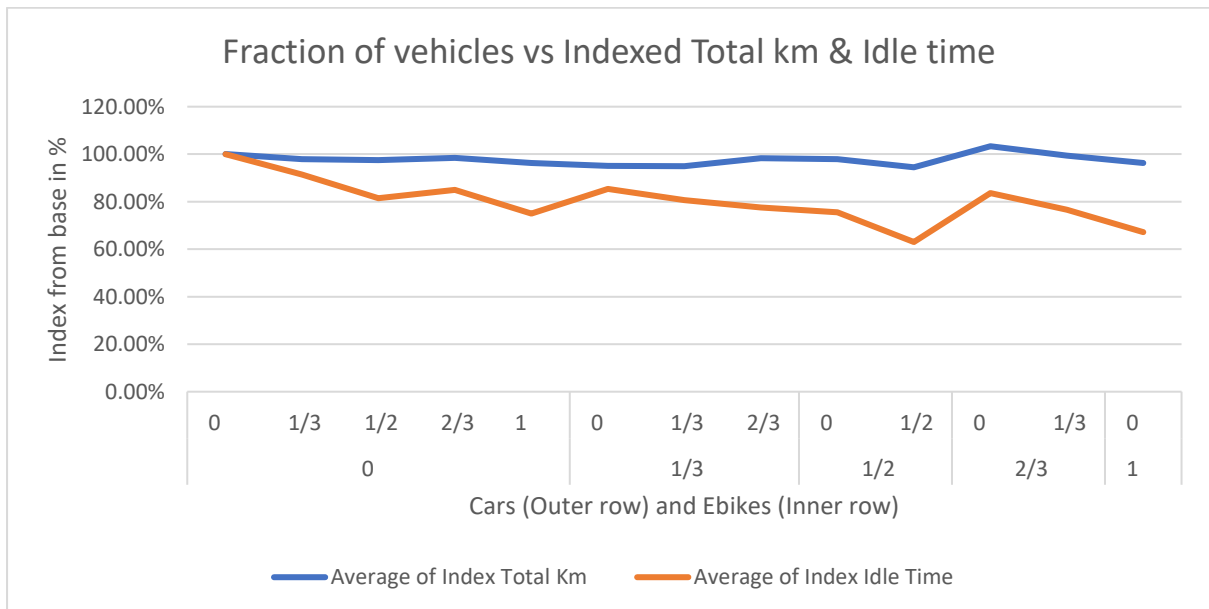
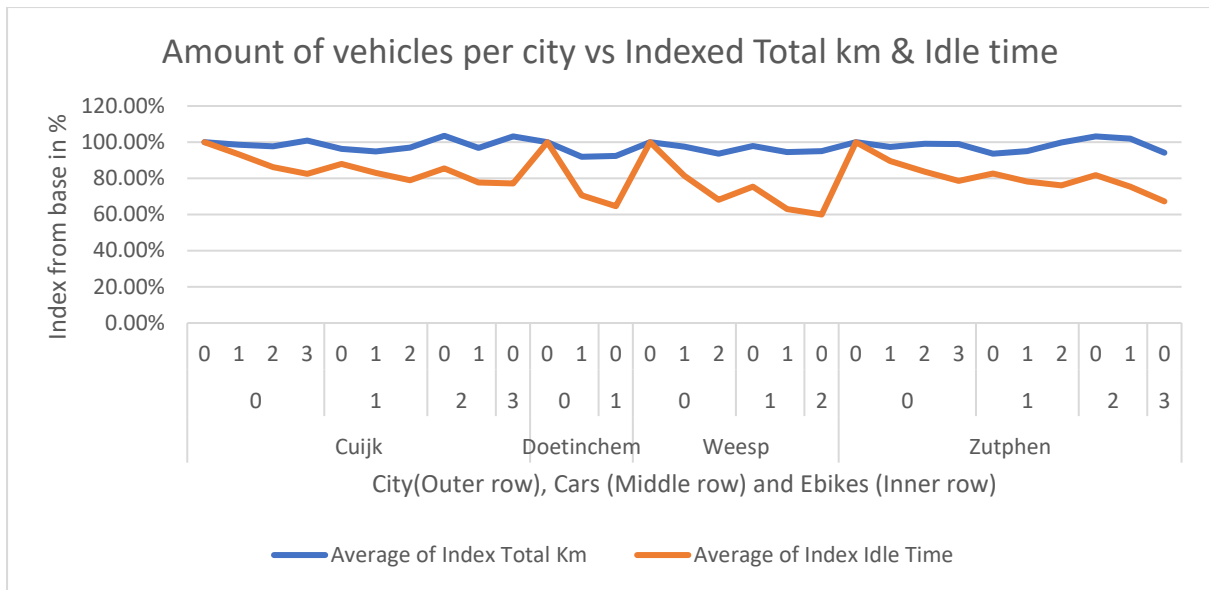


Fraction of vehicles vs Average waiting time & Average late time

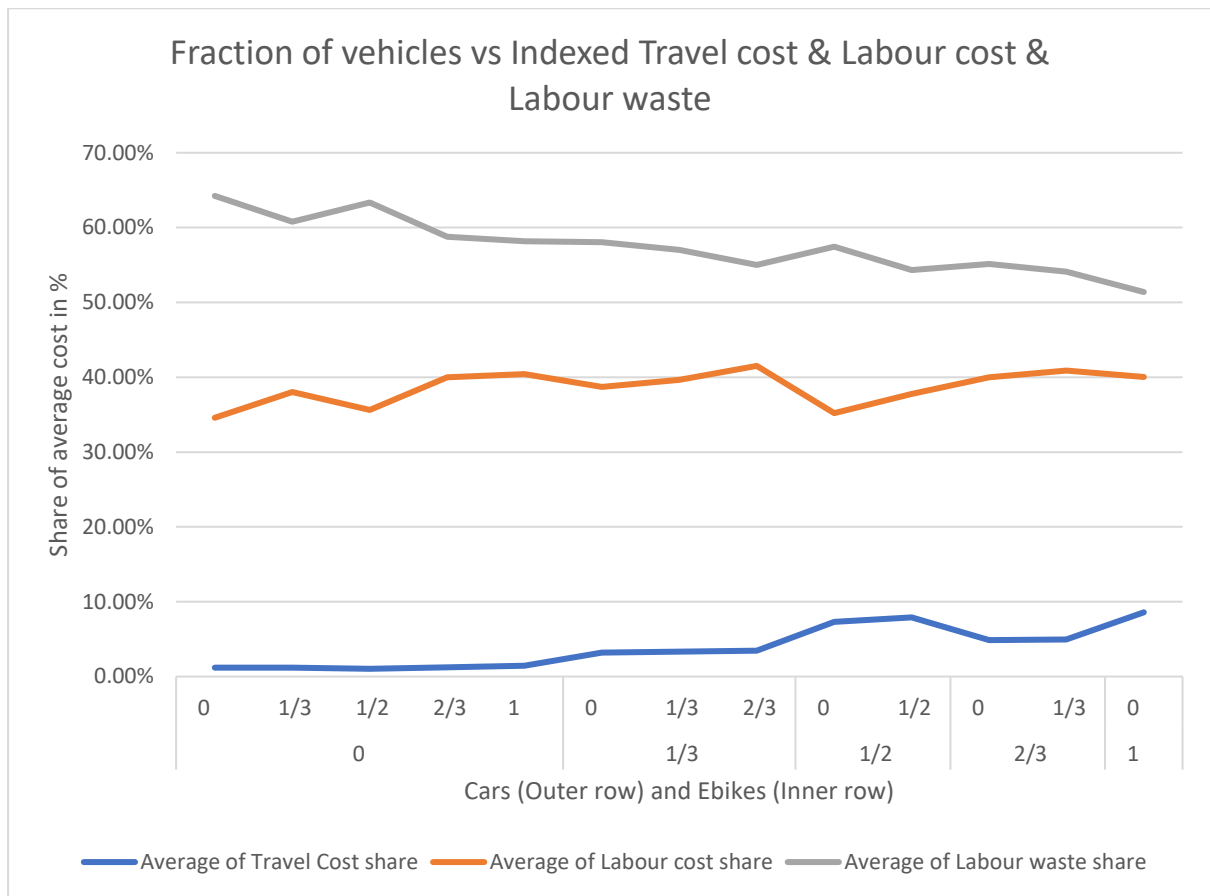


## Appendix E.4: Vehicles vs driving distance and non-service time comparison

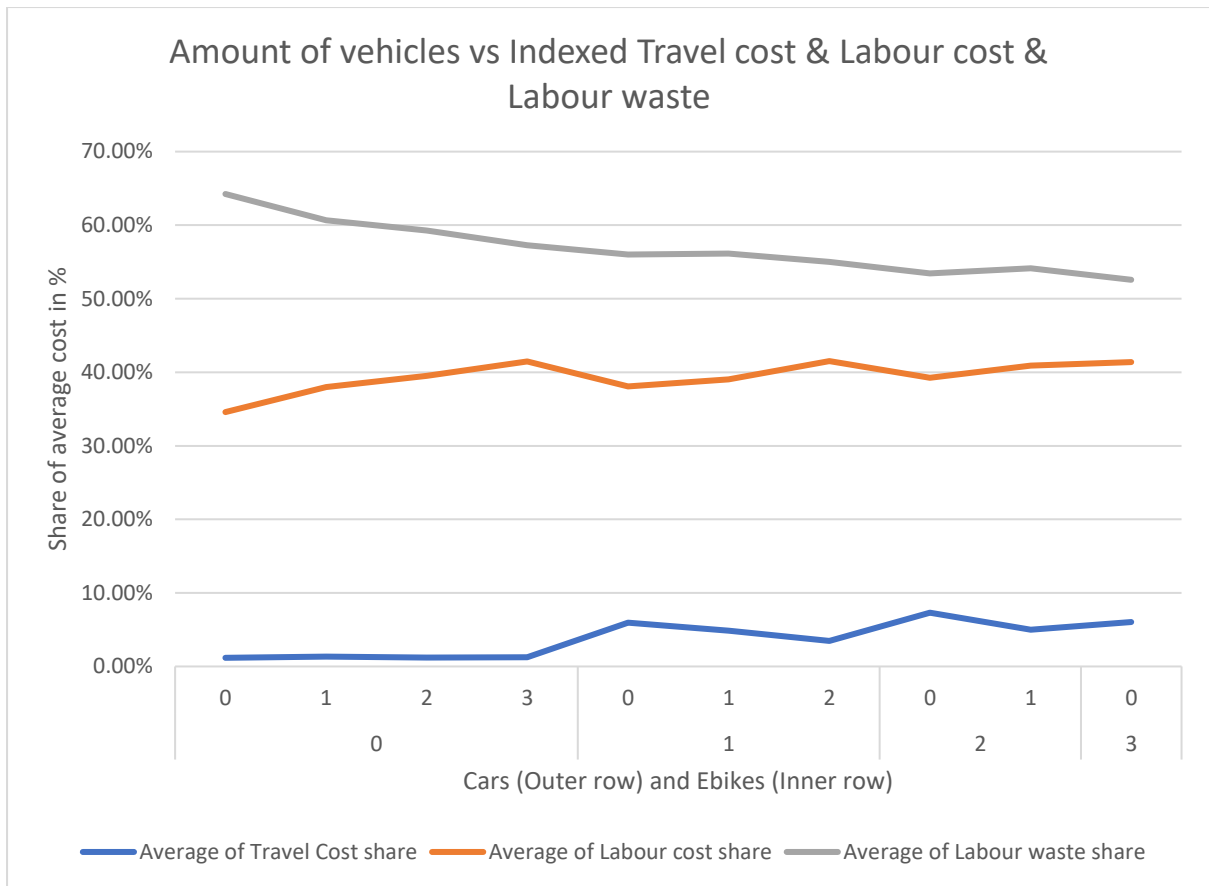




## Appendix E.5: Vehicles vs Average cost share comparison

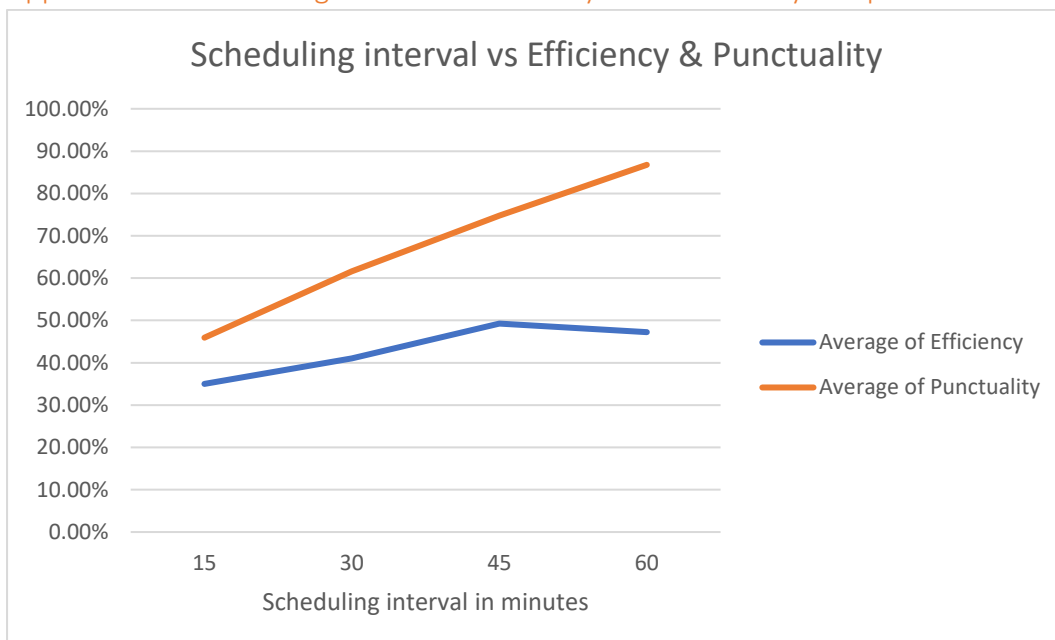


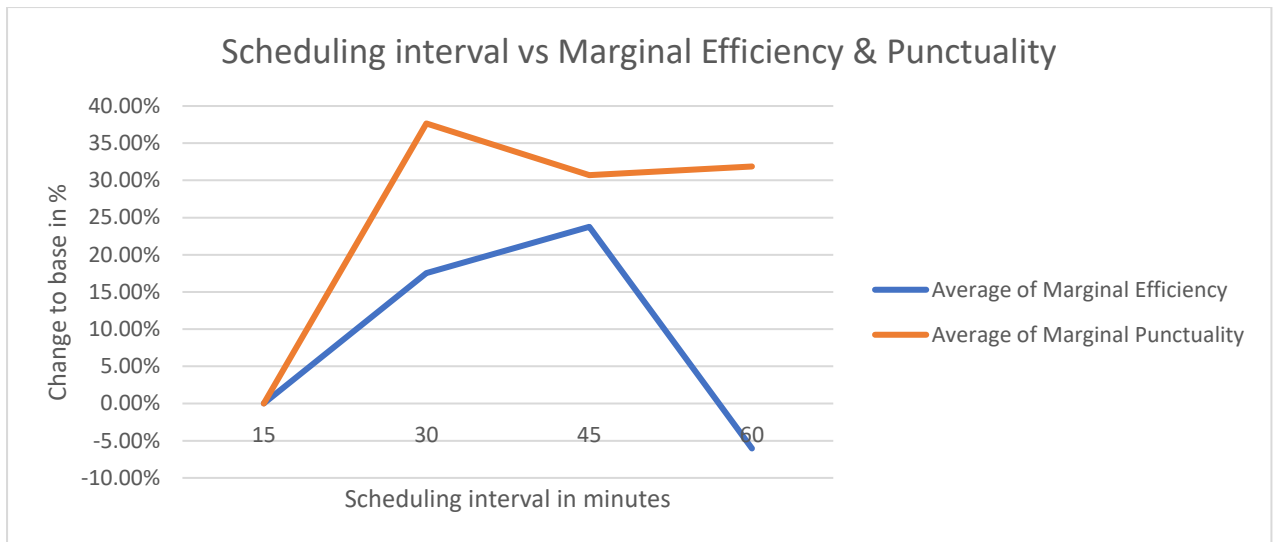




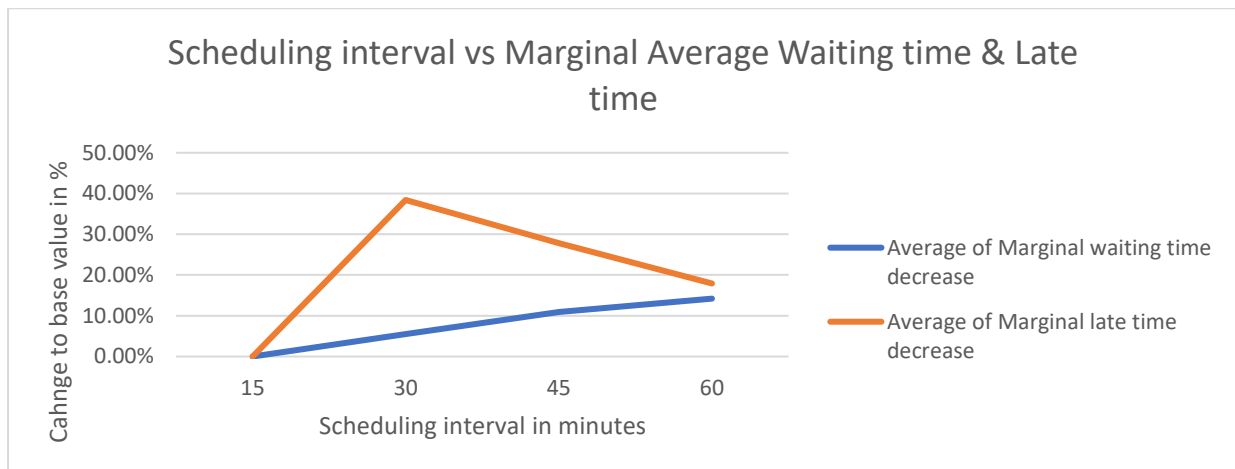
## Appendix F: Algorithm-scheduling simulation graphs

### Appendix F.1: Scheduling interval vs Efficiency and Punctuality comparison

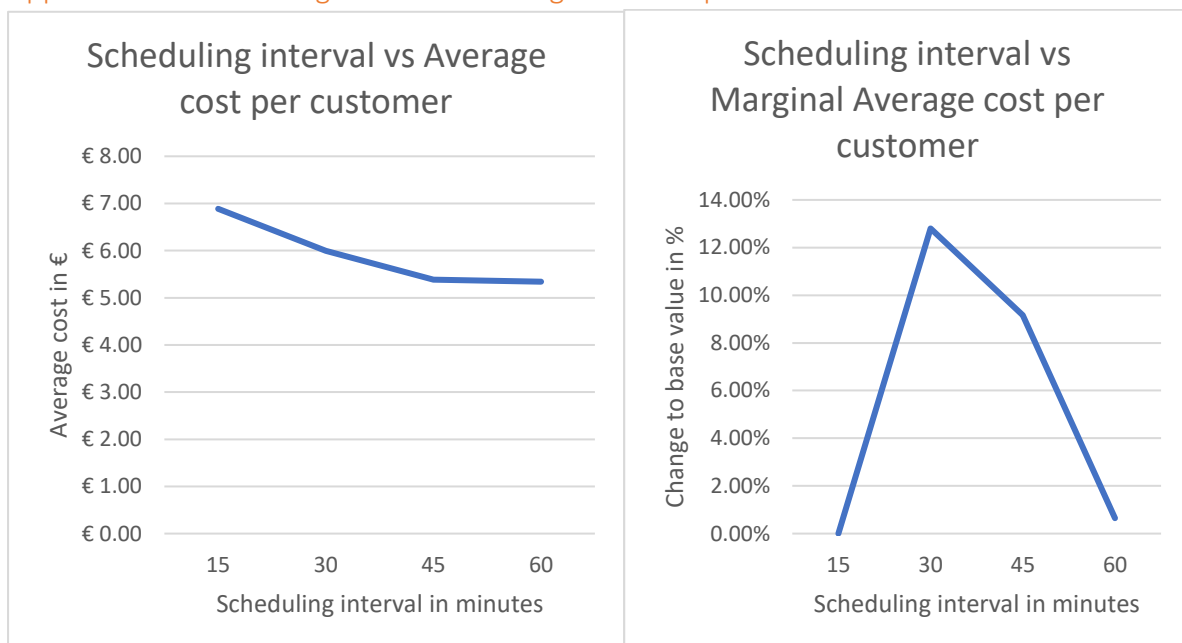




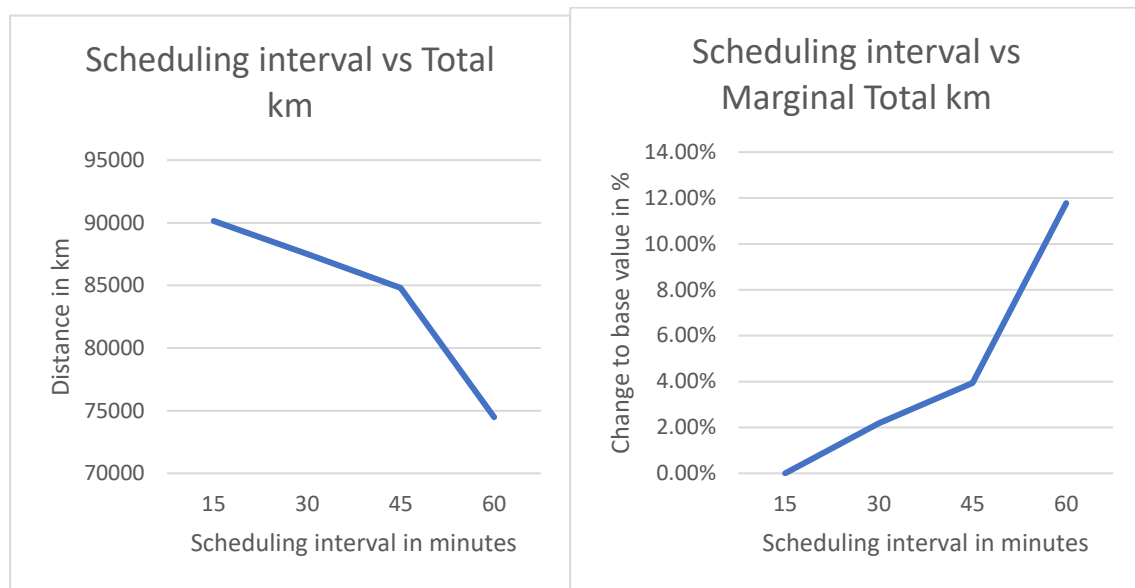
## Appendix F.2: Scheduling interval vs Average waiting time and Late time comparison



## Appendix F.3: Scheduling interval vs Average cost comparison



## Appendix F.4: Scheduling interval vs Total distance comparison



## Appendix G: Full result tables

### Appendix G.1: Vehicle Adaptation simulation tables

Cars	Bikes	Ebikes	Scheduling interval	City	Morning routes	Evening routes	Amount of weeks	Amount of s	Algorithm used	Total customers
0	1	0	15	Doetinchem	1	1	10	100	FALSE	10608
0	1	1	15	Doetinchem	1	1	10	100	FALSE	9706
1	1	0	15	Doetinchem	1	1	10	100	FALSE	10428
0	1	0	15	Weesp	2	2	10	100	FALSE	34156
0	1	1	15	Weesp	2	2	10	100	FALSE	33606
0	1	2	15	Weesp	2	2	10	100	FALSE	32570
1	1	0	15	Weesp	2	2	10	100	FALSE	33748
2	1	0	15	Weesp	2	2	10	100	FALSE	32923
1	1	1	15	Weesp	2	2	10	100	FALSE	32644
0	1	0	15	Cuijk	3	3	10	100	FALSE	49026
0	1	1	15	Cuijk	3	3	10	100	FALSE	47925
0	1	2	15	Cuijk	3	3	10	100	FALSE	48625
0	1	3	15	Cuijk	3	3	10	100	FALSE	48821
1	1	0	15	Cuijk	3	3	10	100	FALSE	46796
2	1	0	15	Cuijk	3	3	10	100	FALSE	49335
3	1	0	15	Cuijk	3	3	10	100	FALSE	49478
1	1	1	15	Cuijk	3	3	10	100	FALSE	46284
1	1	2	15	Cuijk	3	3	10	100	FALSE	47290
2	1	1	15	Cuijk	3	3	10	100	FALSE	46942
0	1	0	15	Zutphen	3	4	10	100	FALSE	62917
0	1	1	15	Zutphen	3	4	10	100	FALSE	61895
0	1	2	15	Zutphen	3	4	10	100	FALSE	62889
0	1	3	15	Zutphen	3	4	10	100	FALSE	62395
1	1	0	15	Zutphen	3	4	10	100	FALSE	60100
2	1	0	15	Zutphen	3	4	10	100	FALSE	64496
3	1	0	15	Zutphen	3	4	10	100	FALSE	59860
1	1	1	15	Zutphen	3	4	10	100	FALSE	60722
1	1	2	15	Zutphen	3	4	10	100	FALSE	62901
2	1	1	15	Zutphen	3	4	10	100	FALSE	64229

Average wait	Average late time per custom	Punctuality	Total km	Car Km	Ebike Km	Bike Km	Total worktime	Non service time	Efficiency	Labour cost	Car costs	Bike costs	Average cost per customer
00:32:53	00:24:22	51.00%	31972.57279	0	0	31972.573	2138:45	4069:59	34.00%	€ 72,331.77	0	€ 1,160.00	€ 6.93
00:12	00:06	75.37%	29388.02594	0	29388.0259	0	1966:25	2876:30	41%	€ 56,420.10	-	€ 1,160.00	€ 5.93
00:06	00:02	84.38%	29549.65396	29549.65	0	0	2103:05	2628:15	44%	€ 55,120.18	€ 6,166.39	-	€ 5.88
00:56	00:44	28.81%	116758.9453	0	0	116758.95	6954:15	14916:39	32%	€ 254,796.14	-	€ 2,320.00	€ 7.53
00:36	00:26	39.35%	113913.3157	0	65999.2627	47914.053	6833:45	12153:56	36%	€ 221,206.63	-	€ 2,320.00	€ 6.65
00:21	00:12	53.52%	109376.674	0	109376.674	0	6623:15	10153:31	39%	€ 195,449.45	-	€ 2,320.00	€ 6.07
00:29	00:20	48.05%	114374.1439	66278.89	0	48095.258	6867:50	11258:24	38%	€ 211,170.70	€ 15,508.33	€ 1,160.00	€ 6.75
00:12	00:05	68.84%	111053.0705	111053.1	0	0	6690:20	8947:04	43%	€ 182,175.86	€ 25,302.32	-	€ 6.30
00:15	00:08	60.60%	110322.4821	64063.07	46259.4095	0	6628:35	9401:17	41%	€ 186,748.07	€ 14,917.53	€ 1,160.00	€ 6.21
00:14	00:07	65.77%	64779.77034	0	0	64779.77	9969:15	14560:17	41%	€ 285,769.19	-	€ 3,480.00	€ 5.90
00:09	00:03	74.59%	63871.16876	0	28588.6896	35282.479	9766:00	13586:29	42%	€ 272,056.45	-	€ 3,480.00	€ 5.75
00:06	00:01	84.43%	63308.29448	0	49065.2095	14243.085	9901:00	12554:01	44%	€ 261,601.10	-	€ 3,480.00	€ 5.45
00:05	00:01	88.49%	65304.78047	0	65304.7805	0	9921:35	12004:04	45%	€ 255,433.94	-	€ 3,480.00	€ 5.30
00:08	00:03	79.01%	62396.38496	28306.43	0	34089.956	9515:50	12796:51	43%	€ 259,942.92	€ 5,506.90	€ 2,134.40	€ 5.72
00:05	00:01	88.19%	67866.20984	51839.66	0	15226.549	10042:15	12458:04	45%	€ 262,128.76	€ 9,735.30	€ 1,102.00	€ 5.55
00:03	00:00	94.19%	66870.04395	66870.04	0	0	10077:30	11225:30	47%	€ 246,180.14	€ 12,073.93	-	€ 5.26
00:05	00:01	86.54%	61428.97753	28131.74	19764.8924	13532.343	9425:00	12097:48	44%	€ 250,740.67	€ 5,490.99	€ 2,111.20	€ 5.58
00:03	00:00	92.00%	62794.4389	28530.6	34263.8389	0	9624:50	11483:23	46%	€ 245,910.75	€ 5,528.40	€ 2,180.80	€ 5.36
00:03	00:00	93.51%	62744.0419	48424.31	14319.7336	0	9531:00	11324:43	46%	€ 242,969.24	€ 9,018.70	€ 1,078.80	€ 5.39
00:35	00:24	38.12%	147057.7963	0	0	147057.8	12821:00	25620:01	33%	€ 447,837.93	-	€ 4,640.00	€ 7.19
00:26	00:16	46.54%	143118.6863	0	51384.0507	91734.636	12608:55	22955:54	35%	€ 414,330.25	-	€ 4,640.00	€ 6.77
00:19	00:10	55.53%	145682.0318	0	95200.6045	50481.427	12814:55	21445:04	37%	€ 399,128.94	-	€ 4,640.00	€ 6.42
00:14	00:07	62.45%	145540.8436	0	130314.902	15225.942	12708:15	20120:10	39%	€ 382,451.06	-	€ 4,640.00	€ 6.20
00:21	00:13	54.59%	137806.4757	49523.73	0	88282.75	12232:30	21194:25	37%	€ 389,423.77	€ 10,974.09	€ 3,480.00	€ 6.72
00:16	00:09	61.96%	151693.4441	99106.15	52587.296	0	13152:45	20960:41	39%	€ 397,421.66	€ 21,997.13	€ 2,320.00	€ 6.54
00:09	00:03	77.73%	138496.147	124117.6	0	14378.582	12192:55	17234:58	41%	€ 342,834.89	€ 26,448.41	€ 1,160.00	€ 6.19
00:16	00:08	61.83%	139785.1728	50258.02	41075.8179	48451.339	12393:15	20035:39	38%	€ 377,796.82	€ 11,182.11	€ 3,480.00	€ 6.46
00:13	00:06	66.87%	146794.5626	52833.63	78931.2526	15029.675	12823:05	19517:07	40%	€ 376,763.35	€ 11,868.51	€ 3,480.00	€ 6.23
00:12	00:06	70.26%	149891.8606	97705.61	36619.7666	15566.483	13083:30	19901:52	40%	€ 377,289.56	€ 21,604.68	€ 2,320.00	€ 6.25

## Appendix G.2: Algorithm-Scheduling simulation tables

Cars	Bikes	Ebikes	Scheduling Inter City	Morning routes	Evening routes	Amount of weeks	Amount of simu Algorithm used	Total customers
0	1	0	15 Doetinchem	1	1	10	100 FALSE	10608
0	1	0	30 Doetinchem	1	1	10	100 TRUE	10658
0	1	0	45 Doetinchem	1	1	10	100 TRUE	10471
0	1	0	60 Doetinchem	1	1	10	100 TRUE	10147
0	1	0	15 Weesp	2	2	10	100 FALSE	34156
0	1	0	30 Weesp	2	2	10	100 TRUE	32484
0	1	0	45 Weesp	2	2	10	100 TRUE	34104
0	1	0	60 Weesp	2	2	10	100 TRUE	33382
0	1	0	15 Cuijk	3	3	10	100 FALSE	49026
0	1	0	30 Cuijk	3	3	10	100 TRUE	47181
0	1	0	45 Cuijk	3	3	10	100 TRUE	47908
0	1	0	60 Cuijk	3	3	10	100 TRUE	49116
0	1	0	15 Zutphen	3	4	10	100 FALSE	62917
0	1	0	30 Zutphen	3	4	10	100 TRUE	62933
0	1	0	45 Zutphen	3	4	10	100 TRUE	64486
0	1	0	60 Zutphen	3	4	10	100 TRUE	59468

Average waiting	Average late time per customer	Punctuality	Total km	Car Km	Ebike Km	Bike Km	Total worktime	Non service time	Efficiency	Labour cost	Car costs	Bike costs	Average cost per customer
00:32:53	00:24:22	51.00%	31972.57279	0	0	31972.573	2138:45	4069:59	34.00%	€ 72,331.77	0	€ 1,160.00	€ 6.93
00:32	00:17	62.69%	33271.59196	0	0	33271.592	2157:10	2360:11	48%	€ 52,627.24	-	€ 1,160.00	5.05
00:27	00:09	77.99%	31561.89812	0	0	31561.898	2125:05	686:19	76%	€ 32,752.88	-	€ 1,160.00	3.24
00:23	00:06	88.67%	25782.62149	0	0	25782.621	2059:45	1221:34	63%	€ 38,227.44	-	€ 1,160.00	3.88
00:56	00:44	28.81%	116758.9453	0	0	116758.95	6954:15	14916:39	32%	€ 254,796.14	-	€ 2,320.00	7.53
00:50	00:30	45.03%	110067.902	0	0	110067.9	6612:15	11899:04	36%	€ 215,656.96	-	€ 2,320.00	6.71
00:49	00:21	54.66%	111241.9044	0	0	111241.9	6971:25	13120:55	35%	€ 234,075.68	-	€ 2,320.00	6.93
00:44	00:12	69.43%	101005.5269	0	0	101005.53	6803:55	12353:48	36%	€ 223,187.48	-	€ 2,320.00	6.76
00:14	00:07	65.77%	64779.77034	0	0	64779.77	9969:15	14560:17	41%	€ 285,769.19	-	€ 3,480.00	5.90
00:13	00:03	83.59%	60598.577	0	0	60598.577	9593:50	12406:11	44%	€ 256,300.30	-	€ 3,480.00	5.51
00:10	00:00	95.84%	55497.22869	0	0	55497.229	9741:15	10513:20	48%	€ 235,966.08	-	€ 3,480.00	5.00
00:08	00:00	99.27%	51930.33512	0	0	51930.335	9994:05	9919:09	50%	€ 231,989.25	-	€ 3,480.00	4.79
00:35	00:24	38.12%	147057.7963	0	0	147057.8	12821:00	25620:01	33%	€ 447,837.93	-	€ 4,640.00	7.19
00:33	00:15	55.01%	146143.2581	0	0	146143.26	12811:55	23108:03	36%	€ 418,467.79	-	€ 4,640.00	6.72
00:31	00:08	70.57%	140920.5	0	0	140920.5	13147:35	21714:58	38%	€ 406,148.84	-	€ 4,640.00	6.37
00:23	00:02	89.68%	119215.4359	0	0	119215.44	12088:20	17804:12	40%	€ 348,248.20	-	€ 4,640.00	5.93

## 9. References

- Carson, Y., & Maria, A. (1997). Simulation optimization: methods and applications. *WSC '97: Proceedings of the 29th conference on Winter simulation*, (pp. 118–126).
- Centraal Bureau voor de Statistiek. (2002). *Fietsend achterop*.
- Cissé, M., Yalçındağ, S., Kergosien, Y., Şahin, E., Lenté, C., & Matta, A. (2017). OR problems related to Home Health Care: A review of relevant routing and scheduling problems. *Operations Research for Health Care*, 1-22.
- Cooper, D., & Schindler, P. (2014). *Business Research Methods*. McGraw-Hill.
- Dengiz, A. O., Atalay, K. D., & Altiparmak, F. (2019). Multiple Service Home Health Care Routing and Scheduling Problem: A Mathematical Model. *Advances in Intelligent Systems and Computing*.
- Dengiz, A. O., Atalay, K. D., & Altiparmak, F. (2021). Novel Routing-Scheduling Problem for Home Health Care Network. *Advances in Intelligent Systems and Computing*.
- Di Mascolo, M., Martinez, C., & Espinouse, M.-L. ( August 2021). Routing and scheduling in Home Health Care: A literature survey and bibliometric analysis. *Computers & Industrial Engineering*.
- Eksioglu, B., Volkan Vural, A., & Reisman, A. (2009). The vehicle routing problem: A taxonomic review. *Computers & Industrial Engineering*, 1472-1483.
- Gazelle. (sd). *Hoe hard gaat een elektrische fiets?* Opgehaald van <https://www.gazelle.nl/advies/elektrische-fietsen/snelheid-elektrische-fiets#:~:text=De%20gemiddelde%20snelheid%20van%20een,voor%20alle%20e%2Dbike%20motorsystemen>.
- Huber, S., Wiemer, H., Schneider, D., & Ihlenfeldt, S. (2018). DMME: Data mining methodology for engineering applications – a holistic extension to the CRISP-DM model. *ScienceDirect*, 403-408.
- IBM. (2021, August 17). *CRISP-DM Help Overview*. Opgehaald van IBM.com: <https://www.ibm.com/docs/en/spss-modeler/SaaS?topic=dm-crisp-help-overview>
- Kroese, D., Brereton, T., Taimre, T., & Botev, Z. (2014). Why the Monte Carlo method is so important today. *WIREs computational statistics*, 386-392.
- Martinez, C., Espinouse, M.-L., & Di Mascolo, M. (sd). Continuity of care in home services: a client-centered heuristic for the home health care routing and scheduling problem. *2018 5th International Conference on Control, Decision and Information Technologies (CoDIT)*.
- Oyola, J., Arntzen, H., & Woodruff, D. (2017). The stochastic vehicle routing problem, a literature review, Part II: solution methods. *EURO Journal on Transportation and Logistics*, 349-388.
- Parlaktuna, O., Sipahioglu, A., & Yazici, A. (2007). A VRP-Based Route Planning for a Mobile Robot. *Turk J Elec Engin*.

- Riazi, S., Chehraz, P., Wigstrom, O., Bengtsson, K., & Lennartson, B. (2014). A Gossip Algorithm for Home Healthcare Scheduling and Routing Problems. *IFAC Proceedings Volumes*, 10754-10759.
- Riazi, S., Wigstrom, O., Bengtsson, K., & Lennartson, B. (2017). Decomposition and distributed algorithms for home healthcare routing and scheduling problem. *22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*.
- Toral, R., & Chakrabarti, A. (1993). Generation of Gaussian distributed random numbers by using a numerical inversion method. *Computer Physics Communication*, 327—334.
- Yim, S., Hong, S.-P., Park, M.-J., & Chung, Y. (2022). Inverse interval scheduling via reduction on a single machine. *European Journal of Operational Research*.