Fleet Composition in Operational Time Slot Management

D.J. Kuiper

Fleet Composition in Operational Time Slot Management

D. J. Kuiper

Enschede, The Netherlands, January 2019

University supervisors	Company supervisor
dr. ir. M.R.K. Mes dr. ir. J.M.J. Schutten	dr. ir. P. van 't Hof

UNIVERSITY OF TWENTE.

ORTEC

Management Summary

In this research we consider the ordering process of e-retailers that sell products which require customers to be at home to receive the order. E-retailers in this context may offer customers the option to select a time window in which their order must be delivered, to increase the customer satisfaction. Based on the order details and the customer location an e-retailer determines a set of time windows from which a customer can select one. It may occur that no time windows are available for a certain customer.

The ordering process of an e-retailer can be divided into a booking period, in which customers place their orders, and a service period, in which the orders are delivered. During the booking period customers request available time windows for their orders to be delivered, and they may or may not confirm their order based on the response of an e-retailer. We consider the case of e-retailers that can categorize their products into two categories (A, B) and their delivery vehicles into at most three categories (1, 2, 3). The main characteristics of these vehicle types are presented in Table i.

Table i. *Main vehicle characteristics for each vehicle type*

Characteristic	Vehicle Type 1	Vehicle Type 2	Vehicle Type 3
Dedication	Dedicated to Order Type A	Dedicated to Order Type B	Non-dedicated
Capacity	Small vehicles	Small vehicles	Large vehicles
Costs	Cheap vehicles	Cheap vehicles	Expensive vehicles

We distinguish between four different use cases, defined by combinations of the three vehicle types that are available. The most important use cases are Use Case 2, in which an e-retailer owns vehicles of Vehicle Type 1 and Vehicle Type 2, and Use Case 4, in which an e-retailer owns vehicles of all three types.

The e-retailers that we consider employ a fixed number of drivers, which is typically smaller than the number of vehicles the e-retailers own. Therefore, an e-retailer cannot use all its delivery vehicles to deliver the customer orders. As the vehicle types differ in their characteristics, not every vehicle type can deliver any customer order. This means that if for instance all drivers are assigned to vehicles from Vehicle Type 2, an e-retailer cannot serve customers of Order Type A. We study this impact of the composition of the delivery fleet (determined by the assignment of the available drivers to the vehicle types) on the performance with regard to customer satisfaction and route efficiency. The main challenge for e-retailers of our context, is to find a proper balance between the customer satisfaction and the efficiency in terms of delivery costs. Our aim is to assist ORTEC in finding a good strategy to deal with an unfixed fleet composition in operational time slot management (i.e., the assignment of the drivers to the vehicles may change during the booking period). Therefore, we formulate the following research question:

> *How can ORTEC deal in a proper way with an unfixed fleet composition when implementing a strategy for operational time slot management for its clients?*

Methodology

To work in a structured way to an answer on this research question, we first did a literature review. Subsequently, we proposed a formal problem definition and we defined a solution approach. After that, we created a simulation model and we defined experiments to obtain insights with regard to our research question. Finally, after we carried out our computational experiments we analyzed the results, to respond our main research question.

Solution Strategies

We distinguish between three solution strategies. The first strategy, the *ORTEC Base Strategy* (OBS), is used as a benchmark and reflects the current approach that ORTEC can use in its software solutions. This strategy determines an initial composition of the delivery fleet (i.e., to which vehicles the drivers are assigned) at the start of the booking period. During the booking period drivers cannot be assigned to other vehicles anymore, so the fleet composition remains fixed. Therefore, we call OBS a static strategy.

Our two other strategies, a *Myopic Strategy* (MYS) and a *Balanced Strategy* (BAS) do not have this restriction and are therefore denominated as dynamic strategies. Both may re-assign drivers to other vehicles during the booking period if desired and thus change the fleet composition. MYS tries to do this whenever a customer cannot be offered any time window, whereas BAS only tries to change the fleet composition if several conditions are met. BAS also rejects unattractive customers, aiming to accept more attractive customers instead.

Results and Conclusions

We carried out experiments for two scenarios, based on data from an ORTEC client that is similar to an e-retailer of our context. The *first scenario* considers a central depot that is located in the province Utrecht of the Netherlands. The customer locations are spread around this depot, mainly in the Dutch provinces Utrecht and Noord-Holland. The *second scenario* considers a central depot located in the Dutch province Noord-Brabant. The customer locations are mainly spread over the provinces Noord-Brabant and Limburg of the Netherlands.

For both scenarios we consider a situation where the observed percentage of customers for each order type is on average equal to the expected percentage of customers for that order type according to historical data (a). We also consider a situation where this is not the case, so the observed ratio of customers of each order type does not equal the historical ratio (b). As mentioned, the main challenge in operational time slot management lays in finding a balance between customer satisfaction and route efficiency. We quantify customer satisfaction by calculating the average percentage of customers that can be served by an e-retailer (the larger the better), and the route efficiency by calculating the average delivery costs per customer that is served (the less the better).

Our results strongly indicate that it is worthwhile for ORTEC to implement a dynamic strategy to deal with an unfixed fleet composition in operational time slot management, at least in cases similar to the scenarios we consider. Furthermore, we see that our smart dynamic strategy (BAS) outperforms our myopic dynamic strategy (MYS) for all scenarios that we consider. We performed a benchmark of the performance of our strategies against two prophet strategies. These strategies know upfront everything about each customer that will arrive during the booking period and may therefore select the most attractive customers in order to serve as many customers as possible. Figure i presents the benchmark results in terms of customer satisfaction and Figure ii the benchmark results in terms of route efficiency.

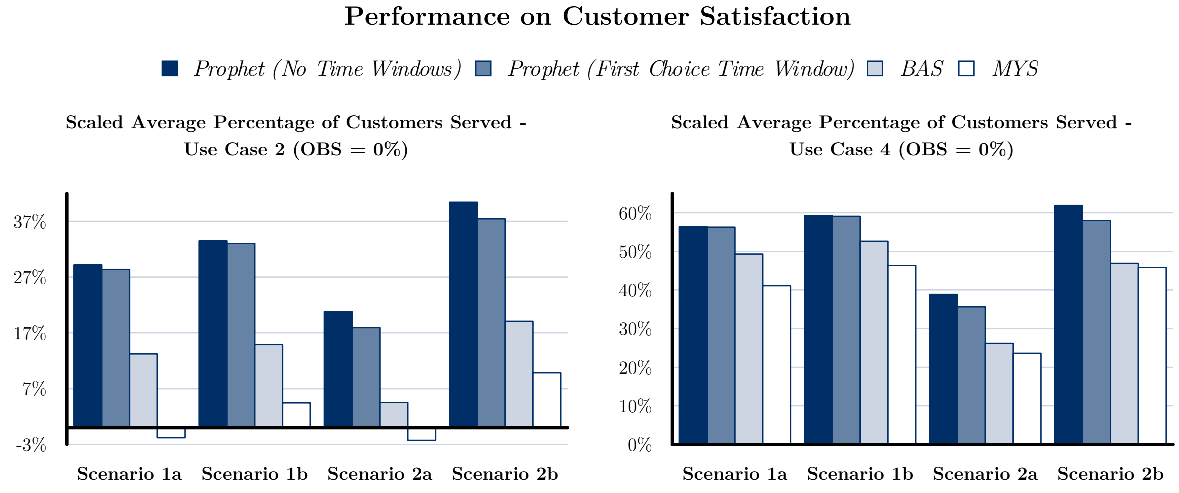


Figure i. *Performance benchmark of our strategies against prophet strategies for customer satisfaction*

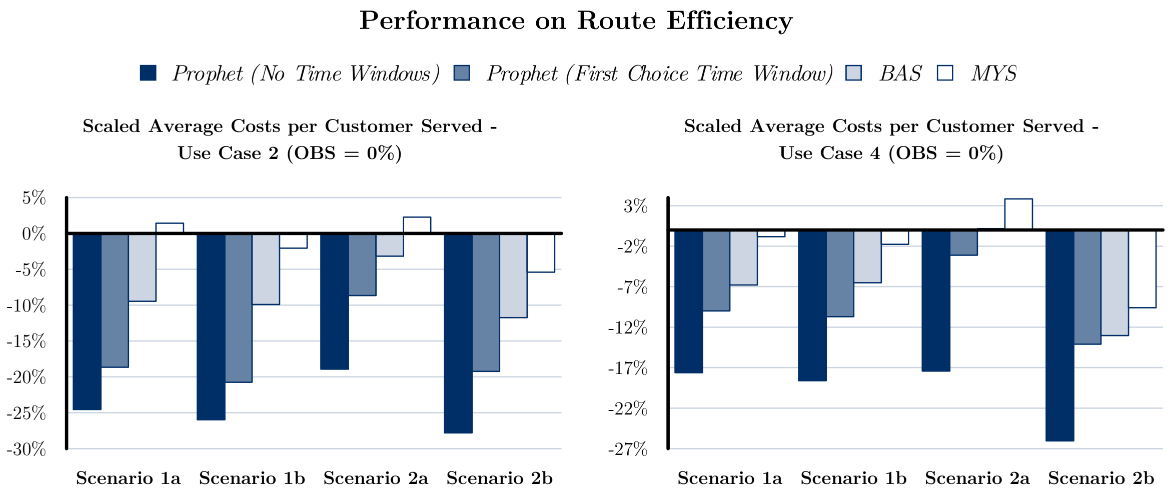


Figure ii. *Performance benchmark of our strategies against prophet strategies for route efficiency*

We see that our strategy BAS shows a decent performance, somewhere half the way between the lower bound set by the performance of OBS (0%) and the upper bound set by the performance of the prophet strategies. At the same time the results show that there is enough space for future improvements of our strategy BAS. We must keep in mind that in reality we can probably never attain the performance of the prophet strategies, because they use information that will never be available to an e-retailer. However, getting closer in performance should be within the bounds of what is possible to achieve. Summarizing, BAS seems to be a good starting point for ORTEC in the search for a proper way to deal with an unfixed fleet composition in operational time slot management. However, more attention should be paid to tuning its parameters and to improving the way in which the strategy handles forecast errors.

Main Recommendations

- We propose to increase the dimensions of the scenarios for which we test the performance of our strategies. It would be very interesting to know if BAS still shows a good performance when for instance more customers arrive during the booking period, more drivers are available, the delivery fleet contains more vehicles and other factors are considered at a larger scale.
- We recommend to investigate how BAS performs in a generalized version of the e-retailer case. What happens for instance when we consider more than two order types and more than three vehicle types? What is the impact of the driver assignment to the vehicles when drivers have different capabilities and cannot drive any vehicle type anymore? It would be interesting for the contribution to literature to find an answer to such questions.
- Another important recommendation is that we propose to spend time to develop a good method to tune the parameters of BAS and improve the strategy. Each practical application of the e-retailer case may require its own parameter tuning. It would be beneficial for ORTEC if a standard method could be developed to find the best parameter tuning for each case in a structured way.
- We propose to study the impact of customer choice behavior in operational time slot management in a deeper way. The customer has a vital impact on whether his or her order can be delivered in the end. An e-retailer can offer many time windows to a customer, but if in the end the customer wants a time window that is not on the list, the customer will not confirm the order. The fact that the customer's behavior has such a large influence, requests further research on this topic.
- Finally, we recommend ORTEC to study the way in which the algorithms in CVRS are configured to solve vehicle routing problems. We now use standard configuration templates, but it may be very beneficial for ORTEC to develop a method that can tune the algorithms according to the need of a certain client. BAS could also make use of such a template, tailored to the need of having a low response time, which is still a problem right now.

Preface

After almost one year, more than 850 hours of editing time of this report consisting of over 65000 words and after writing almost 6000 lines of programming code, this master thesis project marks the end of a period of 5.5 years that I spent studying.

Looking back on this time, I can say that I have learned a lot in many aspects of life. I have had the opportunity to experience many new things during the Industrial Engineering program. I went to Brazil for my minor, where I had a completely new experience of living on my own in a foreign country. Until today I still have many friendships with people I met during that time, which mean a lot to me. Upon my return, I started with an internship to graduate from the bachelor's program.

After graduating, I continued in the same pace as before with my master's. During the first 1.5 year, I spent a lot of time working on projects with my fellow students in the Ravelijn and working as a teaching assistant for Industrial Engineering courses. Finally, I landed at ORTEC for the last step of my "career" as an Industrial Engineering student. Altogether, I have had a lot of great experiences that certainly molded me to be an industrial engineer.

I would like to express my thankfulness to God, Whom I believe capacitated me to complete my study program with good results. Even in difficult times He gave me the strength to carry on. Without my faith in Him I would not be where I am today, which fills me with gratitude toward Him.

I also want to express my gratefulness to the many people who contributed in some way to where I stand today. First of all, I want to thank my parents, my brother, my family and my friends for always supporting me in everything. You know who you are so I don't need to quote any names here, but I'm really grateful for having you in my life and you are extremely important to me!

Second, I want to thank the people who supervised me while I was working at the project of working this master thesis. *Martijn*, you have been a great supervisor! Thanks to your critical input, your detailed feedback and creative ideas for sure the quality of this thesis has increased a lot! *Marco*, although you joined in a later stadium, your participation in the process and your constructive feedback have contributed in a very beneficial way to the quality of my work! *Pim*, for sure this thesis would not be here if you would not have participated in the process (as you kind of invented the subject 😉). Thank you for all the time spent reviewing my ideas and coming up with new ideas and critical remarks always! I thank you and all the other colleagues from ORTEC for your patience in explaining me a lot about how things work within ORTEC, your contribution for sure was indispensable for finishing my thesis.

This thesis marks the end of a phase in my life, but it also marks the beginning of a new phase. As from March this year, I will be starting my career as a Young Professional at Slimstock in Deventer. I do not know yet where this journey will take me, but I'm excited to get to know many new people and to grow further as a professional!

Danny Kuiper
Deventer, January 2019

Abbreviations

BAS *BAlanced Strategy* A smart dynamic strategy developed to deal with the decision whether to accept a customer order.

COMTEC *COMponent TEChnology* A unified technology framework of loosely-coupled components, which ORTEC uses to develop enterprise-ready solutions and real-time planning applications on to sell those to its customers.

CVRS *COMTEC Vehicle Routing Service* An ORTEC service that can be used (standalone as well as in the cloud) to construct solutions for vehicle routing problems.

KPI *Key Performance Indicator* An indicator that can for instance be used to quantify the performance of a strategy and compare it to the performance of other strategies.

MYS *MYopic Strategy* A myopic dynamic strategy developed to deal with the decision whether to accept a customer order.

OBS *ORTEC Base Strategy* A static strategy developed to deal with the decision whether to accept a customer order.

ORD *ORTEC Routing and Dispatch* An ORTEC product that offers advanced planning solutions for dispatch and execution of vehicle routes.

OTS *ORTEC Timeslotting Service* An ORTEC cloud service that can be used to determine which time windows are available to be offered to customers.

TSM *Time Slot Management* Time slot management encompasses all decisions on a strategical, tactical and operational level that are necessary to facilitate the process of offering time windows and assigning one to a customer when an order is placed.

TSP *Traveling Salesman Problem* A combinatorial optimization problem in which the shortest tour must be determined through a set of n points, which all need to be visited once.

VRP *Vehicle Routing Problem* A combinatorial optimization problem in which optimal routes need to be determined for a set of vehicles, given a set of customer orders that need to be delivered.

Contents

1. Introduction

In this chapter we introduce this research and explain its context. Section 1.1 briefly describes the organization where the research takes place. In Section 1.2 we give a description of the case we consider, and we explain why it is relevant for practice. Section 1.3 defines the scope and the goal of this research, and to conclude we present the research framework in Section 1.4.

1.1. About ORTEC

ORTEC started in 1981 as a small company founded by a few Dutch students, who wanted to show the world the value of mathematics. Through the years ORTEC became one of the world's leading suppliers of advanced planning and mathematical optimization solutions, as well as a provider of logistics consultancy services. The company currently has nineteen offices around the globe, and around 900 employees, most of which work at the headquarters in Zoetermeer, The Netherlands.

By optimizing the performance of some of the most iconic businesses in the world, ORTEC gained the respect of industry leaders globally. ORTEC offers software solutions for several industries, such as retail, consumer goods, food & beverage, transportation, and more. The software solutions address different challenges businesses face, in areas such as load assignment, routing, workforce planning and scheduling, field services and warehousing. By offering those solutions, ORTEC aims to increase the economic and social value of their clients, and simultaneously reduce their environmental impact. This contributes to ORTEC's mission, which is to optimize our world using world class mathematics and engineering (ORTEC B.V., 2018).

1.2. Case Background and Description

In the past few years, we see a trend that retail e-commerce sales are growing strongly worldwide. Forecasts indicate that this trend will continue in the years to come (eMarketer, 2018). This growth in retail e-commerce sales imposes logistical challenges on e-retailers when fulfilling the online demand. As the business is competitive and profit margins are small, e-retailers want to minimize their logistics costs. However, at the same time, customers become more and more demanding, which forces e-retailers to comply with high service levels and other restrictions to prevent losing their customers.

In this research, we focus on e-retailers that face demand that requires attended home delivery. In attended home delivery customers need to be at home when the product or the service, which they ordered online, is delivered to them. Examples of those products and services may include groceries, large electronic devices, such as washing machines or dishwashers, but also repairs or installations that need to be conducted at people's homes.

E-retailers may offer the customer the option to select a time window. This obliges an e-retailer to deliver the order to the customer's location not earlier than the start time of the time window, and not later than its end time. The primary objective is to increase customer satisfaction by offering more flexibility to a customer. An important side-effect of offering time windows to customers is that e-retailers can prevent that customers are not at home when the order is delivered. Reaching those objectives comes at a certain cost for an e-retailer, as offering more flexibility to the customer results in more complex restrictions when planning the orders in delivery routes. For instance, without offering time windows to customers, the distributor can decide himself when to deliver an order to a customer. There are no restrictions regarding the sequence in which the orders are delivered. This makes the routes more flexible for changes in the route sequence, which may be desirable when unexpected delays happen. However, when offering time windows, the routes must of course be formed in such a way that the selected time windows are respected as much as possible. This may in practice require additional vehicles compared to the case in which the distributor does not have to cope with time restrictions for delivery. Besides that, the routes may become less efficient, e.g. because a certain neighborhood must be visited twice in one route in different time windows, which may cause detours.

Managing this whole process in a profitable way imposes many challenges on e-retailers. Therefore, this topic offers many incentives for research, to help e-retailers improve the way in which they deal with the process of offering and assigning time windows to their customers. As ORTEC has many (potential) clients in retail e-commerce, ORTEC wants to provide software solutions that help these clients to tackle the challenges. A specific challenge that arises in practice for ORTEC's clients has to do with the fact that they have a heterogeneous fixed delivery fleet. Having a heterogeneous fleet means that they own a fleet consisting of several types of vehicles, which may differ in for instance load capacity, driving speed or other factors. ORTEC's clients need to decide for each day how to use this fleet to deliver customer orders. Especially in cases where the clients employ less drivers than the number of vehicles that they own this decision becomes complex. The drivers must be assigned to one vehicle each, so some vehicles cannot be used for delivery because no driver is assigned to them. The fleet composition that results from the assignment of the drivers to the vehicles determines how many vehicles of each type are used to deliver the customer orders. The fleet composition can have a large influence on the decision of which time windows to offer to customers, as explained in the following sections. This practical challenge serves as motivation of our research topic, which is presented in the remainder of this chapter.

1.2.1. Time Slot Management

Time slot management encompasses all decisions on a strategical, tactical and operational level that are necessary to facilitate the process of offering and assigning a time window to a customer. In this section we give insight into some important decisions for the different levels of control in time slot management, in the context of e-retailers. In Chapter 2 we present an overview of relevant literature about time slot management.

An example of a strategical decision is the definition of the set of time windows that an e-retailer uses. An e-retailer must for instance decide on the number of different time windows to use, the length of each time window (different lengths or equal lengths) and on whether the time windows should have overlap with each other or not. Those decisions are typically fixed for a longer period of time, which makes them important on a strategical level of control.

An important tactical decision is which time windows to make available in a certain region during a certain period. An e-retailer can base this choice on historical data of customer demand for the considered region during similar periods in history. For some regions it may not be desirable to offer narrow time windows. As an example, we may think of areas with high variations in travel times. If time windows would be narrow, the risk of arriving late or early at the customer would be high. This decision may be subject to change, but it is not desirable to take this decision for instance every day. Therefore, we consider this decision on a tactical level of control.

An example of an operational decision is which time windows to assign to a customer when an order comes in. We distinguish between two ways of taking this decision, which both are applied in practice:

1) The customer is offered the option to select a time window from a set of available time windows, composed by an e-retailer.
2) The customer is just assigned a time window by an e-retailer, but at least the customer knows when to be at home to receive the order.

The first option imposes more logistical challenges on e-retailers, but in return the customer satisfaction will be higher compared to the second option. In our research context the focus is on the first option. An e-retailer must then determine which time windows are available for a customer to select from when an order comes in. Being available means that an e-retailer has, or expects to have, enough route capacity to deliver the order of the customer within the considered time window. The customer location and the order quantity (or an estimation of it) serve as inputs here, as well as the planning so far, the remaining route capacity and the composition of the delivery fleet. An e-retailer may take several factors into account, such as forecasts of future demand or restrictions regarding driver capabilities or fleet composition. After deciding which time windows are available, an e-retailer needs to make a choice of whether to present the whole set of available time windows or only a subset. In some cases, it may be desirable to influence customer behavior by not offering certain time windows, which are in fact available. A reason for this may, among others, be that an e-retailer wants to reduce the risk of ending up with delivery routes that are so inefficient, that the profits of increased customer satisfaction do not outweigh the losses due to route inefficiency.

We define the result of all decisions made in time slot management as the *time slotting strategy*. The main focus in this research is on operational decisions in the time slotting strategy. The time slotting strategy should contribute to appropriately balancing the route efficiency and the customer satisfaction. There are many ways to define these two factors. This subject is addressed in a more detailed way in Chapter 2 and Chapter 3.

1.2.2. The E-Retailer Case

In this section we describe some characteristics of e-retailers that are similar to typical ORTEC clients in the retail e-commerce area. After describing our case, we provide some practical examples of typical application contexts, to illustrate the relevance in practice for ORTEC's clients.

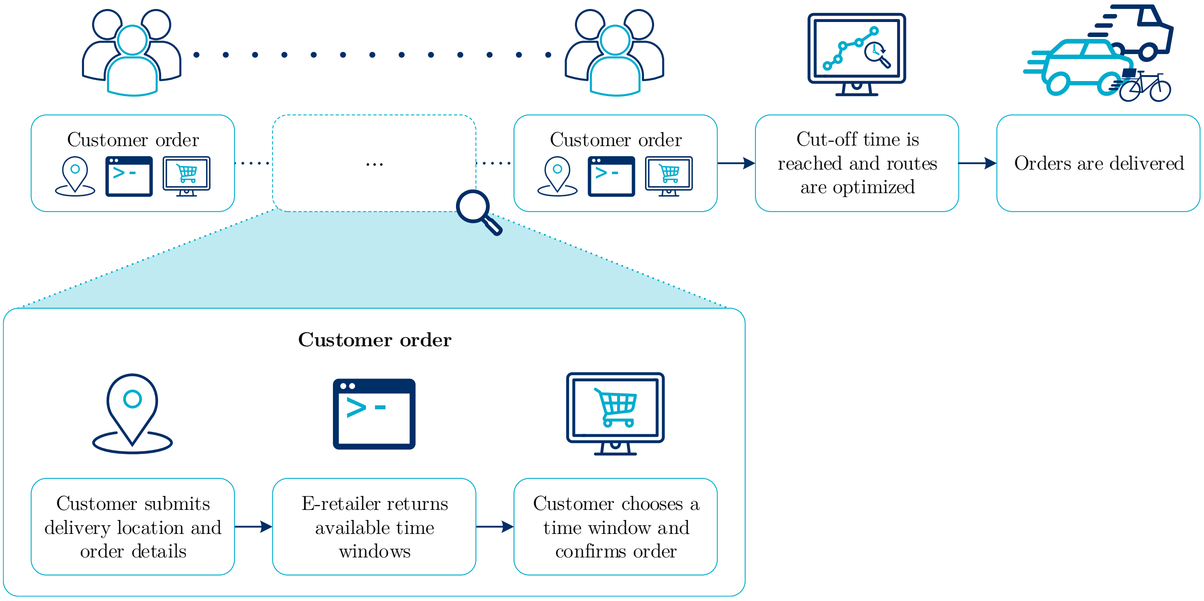

Customer order
...
Customer order
Cut-off time is reached and routes are optimized
Orders are delivered
Customer order
Customer submits delivery location and order details
E-retailer returns available time windows
Customer chooses a time window and confirms order

Figure 1.1. *General overview of ordering and delivery process at an e-retailer*

The e-retailers we consider offer home delivery to their customers. The companies typically own one or more central depots, at which a delivery fleet is available. The composition and the size of the delivery fleet has been determined after thorough analysis and is fixed. The e-retailers have a heterogeneous fleet, which means that vehicle types may differ for instance in load capacity, vehicle speed or so-called capabilities. Examples of capabilities (sometimes referred to as skills) could be suitability for refrigerated transport, or suitability for accessing restricted areas, such as emission zones where only electric vehicles are allowed. The e-retailers in our context employ a fixed number of drivers, that are in most cases capable to drive all vehicle types, but the number of drivers is smaller than the number of vehicles in the fleet. For that reason, not all vehicles can be used at the same time and the e-retailers must determine how many vehicles of each type to use to deliver the customer orders for each day. The e-retailers want to achieve a high customer satisfaction, and therefore they offer the customers the possibility to select a time window in which the order is to be delivered. When customers want to place an order on the website, they first get an overview of the available time windows, based on the delivery location and the order details. After selecting a time window, the customers can pay and confirm their order. For every delivery day, customers can place orders until a so-called cut-off time. This means that after the cut-off time, the e-retailers know all demand details. Then the delivery routes can be optimized, taking into account the restrictions regarding fleet size and composition, vehicle capabilities and the number of drivers that are available. Figure 1.1 gives a general schematic overview of the whole ordering process at an e-retailer in our context for any delivery day.

Just as many of ORTEC's clients do, the e-retailers face some struggles when taking the operational decisions for their time slotting strategy. Especially the decision of how to determine the available time windows for a certain customer order causes some difficulties. Since the number of drivers is fixed and smaller than the number of vehicles available, the e-retailers must decide which vehicles to use and which vehicles not to use for delivery. When the customer order is received, the e-retailers do not yet know what the ideal composition (given all customer orders that will still arrive after this order) of the delivery fleet would be. In other words, they do not know how many vehicles of each type to use to ensure an as efficient as possible delivery process. Therefore, the e-retailers need to determine the expected optimal fleet composition, for instance based on historical data. The reason that this expectation is required, is that the composition of the delivery fleet strongly influences the operational decision regarding which time windows to offer to customers. When making this operational decision, the available time windows are, among others, based on the capabilities, the speed and the capacity of the vehicles in a given delivery fleet. Exchanging a vehicle of a certain type in the delivery fleet for a vehicle of a different type may cause that a certain time window cannot be offered anymore to customers from a certain location. This may for instance be due to the fact that the new vehicle type cannot reach the location in time, whereas the original vehicle type could have reached the location in time.

Therefore, the e-retailers need to take many different aspects into account when determining the operational time slotting strategy. If the e-retailers do not do this in an appropriate way, this may cause unnecessary lost sales, which the e-retailers want to prevent. However, the struggle remains how to deal with the situation in an appropriate way. The e-retailers could make use of a sophisticated but time-consuming algorithm to provide the customer with a list of available time windows. However, there are restrictions on its running time, because customers will not be satisfied when they need to wait long for a response when they request the available time windows. Therefore, it may be better to use a more simplified algorithm, which disregards several restrictions but has a short running time. The downside of using the simplified algorithm is that it increases the risk of not being able to make efficient routes after the cut-off time. Or even worse, some orders may turn out to be unplannable for delivery within the selected time windows. If that happens, either expensive extra workforce capacity must be hired, or customers are left dissatisfied. Both are undesirable. Many of ORTEC's clients deal in some way with challenges that are similar to the ones the e-retailers we just described face. They want ORTEC's software to provide them with an operational time slotting strategy that helps to appropriately balance route efficiency and customer satisfaction. Below we provide two possible examples of applications of the e-retailer case in practice.

Online Grocery Store

An online grocery store called OGS sells several kinds of products. Those products vary from food to beverages, personal care to household products, and other products that can be found in a regular grocery store as well. OGS delivers the orders to the customers' homes, within the time windows that the customers selected. To this end, OGS owns a delivery fleet, composed of a fixed number of vans. Some vans are suitable for refrigerated transport, but other vans are not. OGS also employs a fixed number of drivers, smaller than the number of vehicles in the fleet, who can drive both types of vans. The fleet is located at a central depot, close to the center of a large city. The vans that are suitable for refrigerated transport may also be used for transporting products that do not need cooling. OGS does not apply the practice of order splitting, which implies that if a customer places an order that contains any product that needs refrigerated transport, a van suitable for this type of transport is needed. If no such type of van is available, OGS is forced to reject the order of the customer. Rejecting *in this case means that OGS is not able to offer the customer any time windows, given the vehicles that are available in the fleet and the availability of additional drivers. To prevent undesirable rejections of profitable customers, OGS needs to determine an appropriate composition of their delivery fleet, to prevent rejecting certain profitable customers.*

Food Delivery Service

A food delivery service named FDS offers customers the possibility to order food online, and have it delivered to their homes at the time they selected. To reach a wide range of customers, FDS offers many different types of food. Among others, customers can order Italian food, American food, Brazilian food, Japanese food, Chinese food and Dutch food. FDS is located in a modern metropolis, close to downtown. To deliver the customer orders, FDS owns a fleet of several electric vans, as well as petrol vans. The petrol vans have a larger reach and more capacity in comparison to the electric vans. However, the municipality decided that some areas in the city center are not accessible for non-electric vehicles anymore, which they call emission zones. With this measure the municipality aims to reduce the impact of the emission of exhaust gases, which should contribute to a cleaner environment in the city. To serve the many customers in these emission zones, FDS needs the electric vehicles. FDS employs a limited number of drivers, which is smaller than the number of vehicles they own. Since FDS has a good reputation, customer demand is always larger than FDS' capacity. Customers must therefore order in time to be sure that their order can be fulfilled, due to the limited delivery capabilities. Based on their location and the order quantity, FDS then offers them narrow time windows in which their food can be delivered. FDS wants to be able to serve as many customers as possible. Therefore, it is important for FDS to make use of the right vehicles, taking into consideration the limitations of the vehicles and the impact of the fleet composition on the time windows that can be offered to the customers.

We can define many use cases of the e-retailer case based on what we observe in practice. We limit our analysis in this project to some of the most common use cases of ORTEC's clients. In Chapter 3 we present a more elaborate overview and definition of the use cases of the e-retailer case that we distinguish.

1.3. Research Scope and Research Goal

Our research closely relates to the area of vehicle routing. To be more precise, it is in the realm of the application of time slot management in the context of vehicle routing with a heterogeneous fleet, time windows and some additional restrictions. As mentioned before, the main focus of our research lays on the operational decisions in the time slotting strategy. There are many interesting decisions on this level of control, but we focus specifically on the influence of the fleet composition (given the limit imposed by the number of drivers available) on the decision which time windows to offer to a customer. Several clients of ORTEC deal with problems related to this topic, which makes it a relevant topic for practice. Within ORTEC not much attention has been paid to this subject yet. We study the impact of several ways to deal with an unfixed fleet composition during the ordering process, in the context of the use cases of the e-retailer case that we consider as described in Section 1.2 and Chapter 3.

To measure this impact objectively, we need general measures of the quality of a final solution, which we define later in this research (Chapter 3). Important aspects to consider are customer satisfaction and route efficiency, as mentioned before. The goal of this research is to provide insights into how ORTEC can deal with a fleet composition that is unfixed during the ordering process, given the restrictions imposed by the number of drivers that are available. These insights can then be used when ORTEC implements strategies for operational time slot management for its clients. As explained in Section 1.2, the decision of which time windows to offer to the customers is based on a choice regarding the fleet composition. If the fleet composition (that was assumed to be optimal) changes, the resulting time windows that are offered for a certain customer order may change as well. With this research we aim to increase the knowledge within ORTEC regarding this topic.

To attain our research goal, we seek to design several strategies that can be used to determine what the ideal fleet composition would be at a certain time during the ordering process. This can for instance be based on estimations made with historical data, already known demand and other factors. Those strategies should take into account that it is undesirable to have high running times, because this leaves customers dissatisfied. We analyze the impact of the strategies that we design for different scenarios and different use cases of the e-retailer case. By doing so we seek to provide insights that ORTEC may use to configure operational time slotting strategies for clients that are similar to the e-retailers we discussed in our case description. We make use of simulation techniques to evaluate the performance of the strategies we design. We formulate our main research question in the following way:

> *How can ORTEC deal in a proper way with an unfixed fleet composition when implementing a strategy for operational time slot management for its clients?*

In the next section we describe the way in which we aim to reach our goal. We formulate several sets of research questions, which guide us to an answer to our main research question as defined above.

1.4. Research Framework

To work in a structured way toward our research goal, we define a research framework in this section. To find an answer to our main research question, we need to answer several other questions on the way. We start exploring what is already known in literature about the relevant aspects of the vehicle routing problem for our context. Besides that, we study what has been found in literature about time slot management in attended home delivery or similar fields, with a special focus on what is common practice for the way in which available time windows are offered to the customers. Finally, we seek to increase our knowledge regarding ways to model customer choice behavior. Therefore, we study what is known about this subject as well. Besides the substantive knowledge we obtain about these subjects, we also aim to obtain more information about how similar studies have been conducted in the past. Our first set of research questions is the following:

1) What can we learn from literature…
   a. …about vehicle routing problems in the context of a heterogeneous fleet and time windows?
   b. …about time slot management in attended home delivery or similar application contexts?
   c. …about the way of measuring the solution quality of vehicle routing problems in the context of time slot management in attended home delivery?
   d. …about modeling customer choice behavior in time slot management?

After obtaining background knowledge, we make the step back to our own context. As the e-retailer case is a very general case that can be applied in many contexts, we first need to define the use cases we consider. We also need to find out how we can properly model the e-retailer case, so in the end we can simulate the performance of different strategies that we design. Therefore, we seek an answer to our second set of research questions:

2) How can we model different use cases of the e-retailer case, to test the performance of the different strategies that tell us how to deal with an unfixed fleet composition during the ordering process?
   a. Which use cases of the e-retailer case should we consider?
   b. How can we formally define the problem we are tackling?
   c. How do we measure the quality of a final solution?
   d. Which hypotheses do we put to a test in our simulations?

The next step in our research consists of designing the new strategies we want to compare to each other. The different strategies should of course be representative for different approaches of solving the problem of how to deal with an unfixed fleet composition in operational time slot management. Our third set of research questions addresses this topic and is as follows:

3) Which solution strategies do we design to deal with an unfixed fleet composition during the ordering process?
   a. What are the problems for which our solution strategies should come up with a decision?
   b. How can we deal with these problems in the solution strategies that we design?

After designing the solution strategies, we simulate their performance for different scenarios based on the use cases of the e-retailer case. We perform simulation runs to find out whether the hypotheses as defined in 2d can be confirmed. Of course, we need input data to define the different scenarios that we consider, which we use to run the simulations. We therefore design experiments that we perform to put our hypotheses to a test. This results in our fourth set of research questions:

4) How can we simulate the ordering process from the e-retailer case?
   a. What are the inputs that we need for our simulations and how do we process them?
   b. How can we run our simulations?
   c. Which scenarios are we going to use as input data for our simulations?
   d. Which experiments do we define to validate our hypotheses?

Finally, after performing all the simulation runs, we analyze the results in order to confirm or reject our hypotheses. For that reason, our last set of research questions is the following:

5) What are the insights that we can obtain from the results of our simulations?
   a. Do the simulation results confirm the hypotheses defined earlier?
   b. What general results do we observe from our experiments?

Answering these five sets of research questions will give us valuable insights into how ORTEC should deal with an unfixed fleet composition when implementing an operational time slotting strategy for a client. Figure 1.2 shows a schematic overview of the research framework to give an idea of the research structure and to link each part of the research to the corresponding chapter(s) and research questions.

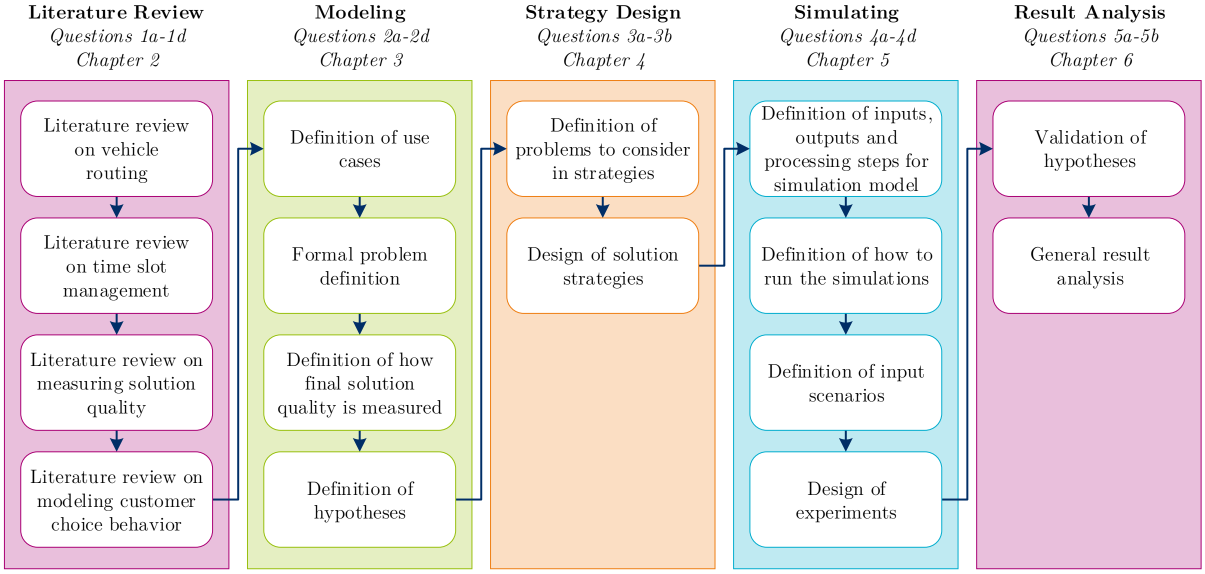

Literature Review
Questions 1a-1d
Chapter 2
Literature review on vehicle routing
Literature review on time slot management
Literature review on measuring solution quality
Literature review on modeling customer choice behavior
Modeling
Questions 2a-2d
Chapter 3
Definition of use cases
Formal problem definition
Definition of how final solution quality is measured
Definition of hypotheses
Strategy Design
Questions 3a-3b
Chapter 4
Definition of problems to consider in strategies
Design of solution strategies
Simulating
Questions 4a-4d
Chapter 5
Definition of inputs, outputs and processing steps for simulation model
Definition of how to run the simulations
Definition of input scenarios
Design of experiments
Result Analysis
Questions 5a-5b
Chapter 6
Validation of hypotheses
General result analysis

Figure 1.2. *Schematic overview of the research framework*

2. Literature Review

In this chapter we present an overview of some relevant findings in literature, which help us better understand the context of our problem. In Section 2.1 we investigate the subject of the vehicle routing problem. Section 2.2 gives an overview of what is currently known about time slot management in attended home delivery. Finally, Section 2.3 presents our findings about modeling customer choice behavior in time slot management.

2.1. The Vehicle Routing Problem

In this section we give a summarized overview of the vehicle routing problem. More specifically, we first present the basic vehicle routing problem in Section 2.1.1. We then take a closer look into some of its different variants which arose over the years in Section 2.1.2. Finally, we give insight into different solution methods that have been developed, and into measures for solution quality in Section 2.1.3.

2.1.1. Basic Problem Definition

The *Vehicle Routing Problem* (VRP) is an NP-hard combinatorial optimization problem (Lenstra & Rinnooy Kan, 1981; Vigo & Toth, 2014), which was introduced many years ago, under the name *Truck Dispatching Problem* (Dantzig & Ramser, 1959). Dantzig & Ramser introduced the problem as a generalization of the well-known *Traveling Salesman Problem* (TSP), which was formulated by Flood (1956) a few years earlier. Ever since its introduction, the VRP has been a problem which intrigued many researchers, due to its relevance for practice.

The objective of the classical VRP, as introduced by Dantzig & Ramser, is basically to find the shortest route that passes once through a set of n given locations, just as in the TSP. However, the VRP differs from the TSP regarding capacity restrictions. In the TSP, the assumption is that one route can cover all n points. The VRP takes into account that several carriers may be required to serve all delivery points, due to a limited capacity of the individual carriers. Dantzig & Ramser came up with an iterative computational procedure to solve their VRP. Their formulation of the problem and a solution method paved the way for many others to come up with improved solution methods and different versions of the VRP. Clarke & Wright (1964) were the first ones to come up with an improved solution method, which resulted in a distance reduction of almost 20% compared to the solution obtained with Dantzig & Ramser's method for their test case. The method Clarke & Wright used became known as the savings algorithm. The papers written by Dantzig & Ramser and Clarke & Wright are considered as pioneering papers for studies of the VRP (Vigo & Toth, 2014).

Nowadays the VRP is often represented using graphs (Chang & Chen, 2007; Cordeau, Laporte, Savelsbergh, & Vigo, 2007; Eilam Tzoreff, Granot, Granot, & Sošić, 2002; Jiang, Ng, Poh, & Teo, 2014; Munari, Dollevoet, & Spliet, 2016). To give an example, the classical VRP is defined on an undirected graph $G = (V, A)$. The assumption here is that we have a symmetric problem, in the sense that the direction in which we cross an edge does not matter. In case the sequence in the routes does matter, we can represent the classical VRP on a directed graph. The vertex set $V = \{0, 1, ..., n\}$ contains nodes $i \in V \setminus \{0\}$, which represent customers with demand $q_i > 0$ (for instance expressed as the weight in kg). Vertex 0 represents a depot. The set A consists of the edges (i, j) between each pair of nodes $i, j \in V \setminus \{0\}$. To every edge a travel cost of c_{ij}, for instance the distance in km, is associated. At the depot a fleet of m identical vehicles is available, which all have capacity Q. The objective of solving the VRP is to find a set of m routes with minimized total costs, in such a way that each route starts and ends at the depot. Furthermore, all customers must be visited exactly once in one of the routes, in such a way that the total demand of all customers visited in a route does not exceed the vehicle capacity. Figure 2.1 shows a graphical example of a possible solution to a random instance of the classical VRP.

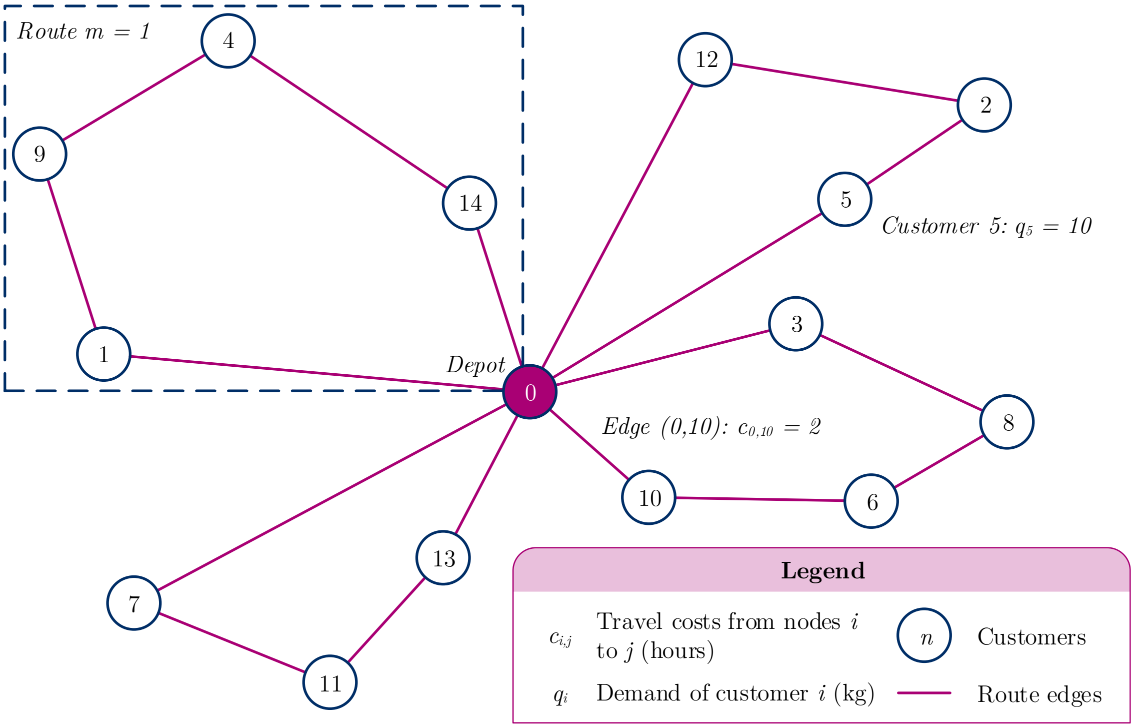

Route m = 1
Depot
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
Customer 5: $q_5 = 10$
Edge (0,10): $c_{0,10} = 2$
Legend
$c_{i,j}$ Travel costs from nodes i to j (hours)
q_i Demand of customer i (kg)
n Customers
Route edges

Figure 2.1. *Graphical representation of a solution to an instance of the classical VRP*

2.1.2. Variants of the Vehicle Routing Problem

Over the years many different variants of the classical VRP have been proposed. Dantzig & Ramser (1959) considered a homogeneous delivery fleet when they first introduced the VRP, in the sense that all vehicles they considered had the same capacity. However, they already mentioned the possibility of considering a heterogeneous delivery fleet, in the sense that the carriers have different capacities. This extension is relevant for our research, in which we consider vehicles that, by definition, have different characteristics. Besides this variation in fleet mix, many other characteristics of the VRP can be varied. Examples are variations in fleet size (fixed, unfixed), nature of demand (deterministic, stochastic), restrictions on delivery times, type of demand (pickup, delivery, both), service times, vehicle characteristics, travel times, and many other features. We briefly discuss a few types of the VRP here, which are relevant for our research. There are many other interesting variants, but we do not report extensively on those here. For anyone who would like to obtain more knowledge on the different variants, rich literature is available on this subject. Braekers, Ramaekers, & Van Nieuwenhuyse (2016) provide a good starting point with their state of the art classification and review of the VRP.

The *Heterogeneous Fleet VRP* (HVRP) is the first variant of the VRP we consider. The HVRP considers a fleet of different types of capacitated vehicles, where each vehicle type has a fixed cost. Those vehicles have to serve a set of customers of which the demand is known. The heterogeneity of the fleet can be determined by various factors. In most studies we see that the heterogeneity is characterized by vehicle capacity and vehicle costs (Koç, Bektaş, Jabali, & Laporte, 2016; Pessoa, Uchoa, & De Aragão, 2009). Two common variants of the HVRP are the *Fleet Size and Mix VRP* (FSMVRP) (Golden, Assad, Levy, & Gheysens, 1984) and the *Heterogeneous Fixed Fleet VRP* (HFVRP) (Taillard, 1999). The FSMVRP is a special instance of the HFVRP in which the number of vehicles per type is infinite. The FSMVRP is studied more extensively in literature, probably because it is easier to solve than the HFVRP. The FSMVRP is relevant for strategic decisions regarding fleet dimensioning, so it is generally applied in situations where long-term decisions must be taken. The HFVRP is mainly applied in an operational context, when a company already owns a delivery fleet corresponding to their strategic decisions. The HFVRP is then solved to determine which of the vehicles to use to serve customer demand on an operational level (Brandão, 2011; Paraskevopoulos, Repoussis, Tarantilis, Ioannou, & Prastacos, 2008).

For us, the most relevant version of the HVRP is the HFVRP, because the fleet dimensioning decisions are given as input in the context we consider. In the HFVRP, we deal with a fixed number of different vehicle types. The objective is to find out how to make the best use of this fleet to fulfil customer demand. Companies need a heterogeneous fleet to be able to cope with different demand characteristics. For some customers it may for instance be required to have vehicles with large capacities to fulfil demand, while for other customers this may not be the case. Some customer locations may impose access restrictions, or they may be out of reach for certain vehicles types, where using a different vehicle type may solve the problem. But of course, this may come at a certain additional cost for using that vehicle. These examples illustrate the practical applicability of the HFVRP (Li, Golden, & Wasil, 2007).

The *VRP with Time Windows* (VRPTW) is another interesting variant of the VRP. It was first introduced in some case studies, around 50 years ago (Cook & Russell, 1978; Knight & Hofer, 1968; Pullen & Webb, 1967). Later, a more general solution method was proposed (Solomon, 1987), aiming to provide a well-performing approach for practical sized problems and benchmarks for future research. The VRPTW is essentially equal to the VRP with one additional restriction. This restriction states that the service at a customer must not start earlier than the start time of the time window provided by the customer and must not start later than the end time of this time window. There are two types of time window restrictions. Time window restrictions can be soft, which implies that the service to a customer can start before or after the selected time window, but this violation comes at a certain cost (penalty). Hard time window restrictions imply that no violations of the selected time windows are allowed. The VRPTW has been intensively researched over the years, which resulted in many reviews of this problem (Bräysy & Gendreau, 2005a, 2005b; Cordeau et al., 2007; Desrosiers, Dumas, Solomon, & Soumis, 1995; Gendreau & Tarantilis, 2010; Kallehauge, Larsen, Madsen, & Solomon, 2005). The reviews provide a formal definition of the VRPTW as well as an overview of the many solution techniques that were proposed over the years.

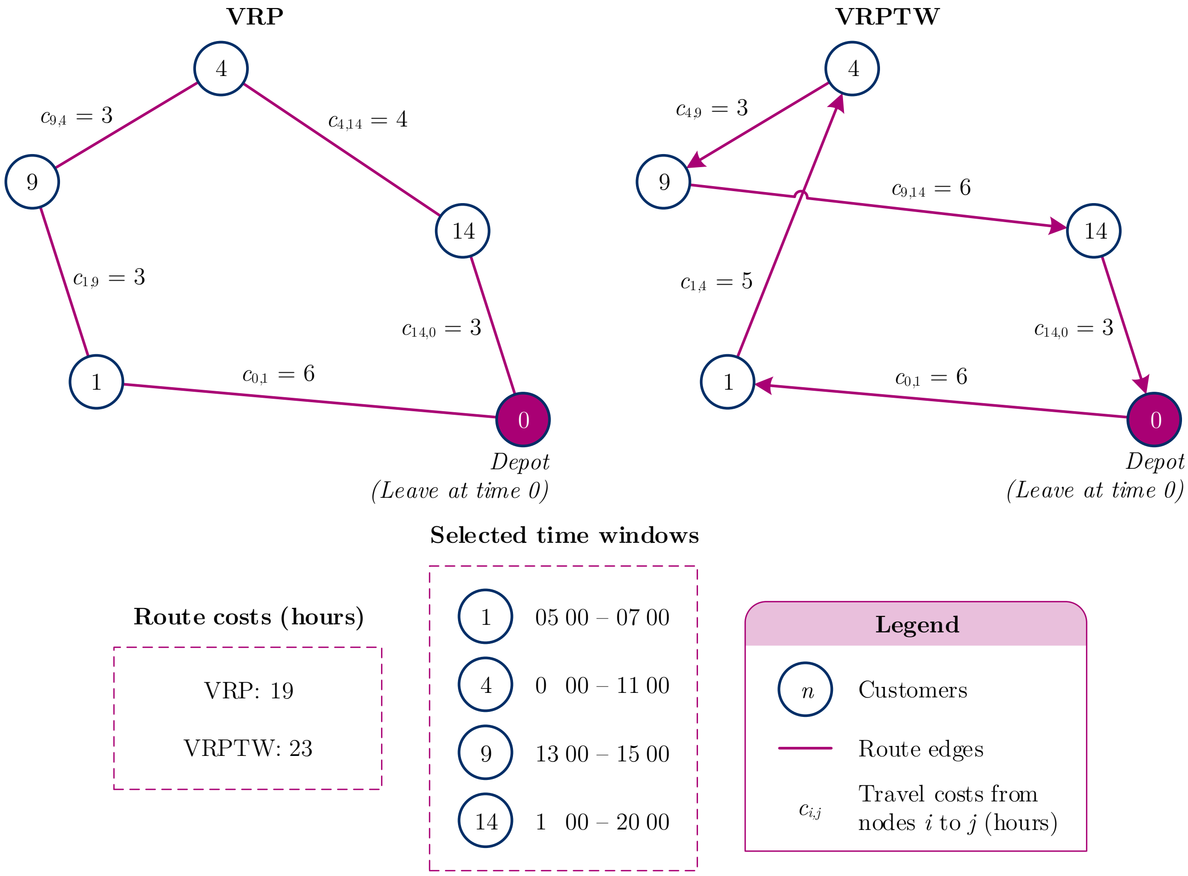


Figure 2.2. *The effect on route costs for route 1 from Figure 2.1 when adding time window restrictions to VRP*

It is much more difficult to find an efficient solution in terms of driving time and driving distance for the VRPTW compared to the classical VRP. To illustrate the effect, we consider the VRP as presented in Figure 2.1. We assign travel costs expressed in hours (c_{ij}) to all the edges of the routes, as displayed in Figure 2.2. For the classical VRP, the sequence in which the customers are served does not matter, because there are no time window restrictions. This gives us a cost of 19 for traveling across all the edges of the route. But as soon as we take the time window restrictions into account, we cannot drive the same route anymore. The time windows for each customer node as displayed in Figure 2.2 force us to drive the route in a different sequence, as displayed by the arrows. Note that the solution to the VRPTW is defined on a directed graph instead of on an undirected graph, which is the case for the classical VRP. We see, that serving the same customers now implies a route cost of 23. It becomes clear that we can never obtain an optimal solution to the VRPTW that is more efficient than an optimal solution to the classical VRP for the same instance but without time window restrictions. The best possible result is an optimal solution that is just as efficient as an optimal solution to the classical VRP, but in most cases optimal solutions to the VRPTW are worse. Therefore, as we already mentioned in Chapter 1, time window restrictions are only added because offering time windows to customers increases customer satisfaction.

For our research, a relevant problem lies in the combination of the HFVRP and the VRPTW. Literature on this variant of the VRP, the *Heterogeneous Fixed Fleet VRP with Time Windows* (HFVRPTW) is scarce compared to other variants. The main focus of publications about this variant of the VRP, is to provide heuristic techniques that can solve an offline version of the HFVRPTW (Brandão, 2011; Jiang et al., 2014; Paraskevopoulos et al., 2008; Yepes & Medina, 2006). The reason for this is that exact methods do not provide satisfactory solutions within a reasonable amount of time. Just as in most of the literature about solving the VRP variants, the assumption is that the problem is solved only after the moment that all customer orders are placed, when the selected time windows are all known. Therefore, we call it the offline version of the problem (see Figure 2.3).

This is an essential difference with the context we consider in this research, in which we deal primarily with an online version of the HFVRPTW. Instead of finding a solution given the time windows selected by customers, we provide available time windows to a customer, based on the solution we built up until the order comes in. In other words, the demand becomes known during the construction of routes, instead of constructing routes after all demand has become known. Although the perspective is different, what remains the same is that in the end we need to come up with an as good as possible solution of the HFVRPTW with the confirmed customer orders and the corresponding time windows as input. Therefore, publications about the HFVRPTW may provide valuable insights for the construction of our simulation model and the choice of our solution methods.

2.1.3. Solution Methods

The main focus of studies of the VRP over the years has been to develop solution techniques that provide efficient solutions in a reasonable amount of time. As mentioned before, the VRP (and most of its variants) is NP-hard and therefore only small instances can be solved to optimality in polynomial time. For the classical VRP, many solution methods have been developed since the introduction of the problem. We see that several exact methods have been developed. Those methods make use of branch-and-bound, set partitioning, dynamic programming and branch-and-cut algorithms to solve the classical VRP. Besides those exact methods, we see two types of heuristics that have been developed, classical heuristics and metaheuristics. Classical heuristics include route construction heuristics, two-phase heuristics consisting of both a clustering and a routing phase, and route improvement heuristics. Metaheuristics include local search heuristics, population search heuristics and learning mechanisms. Metaheuristics are less likely to get stuck in local optima than classical heuristics. Cordeau et al. (2007) provide a detailed overview with more information about these solutions methods.

For the HVRP, Koç et al. (2016) provide an overview of the different solution methods that have been developed over the years. They point out that we have to distinguish between the FSMVRP and the

HFVRP when it comes to solution methods. For the FSMVRP, much more methods were developed, including lower bound and exact algorithms, as well as continuous approximation models and heuristics. For the HFVRP, they conclude that there is no exact algorithm available yet, but there are several heuristics methods that have been published, e.g., tabu search heuristics.

Cordeau et al. (2007) also provide an overview of solutions methods for the VRPTW, including both exact methods and heuristics. The exact methods developed for the VRPTW include algorithms based on Lagrangian relaxation, column generation and branch-and-cut. Heuristics for solving the VRPTW include both construction and improvement heuristics, as well as metaheuristics, for instance based on tabu search or genetic algorithms. Gendreau & Tarantilis (2010) point out that for large instances of the VRPTW it is not worthwhile, or even impossible in polynomial time, to make use of exact methods. They provide an overview of metaheuristics, parallel and cooperative search methods and hybrid optimization algorithms that are used when dealing with large instances of the VRPTW in practice. Another overview of solution methods for the VRPTW can be found in the work of Bräysy & Gendreau (2005a, 2005b), where they discuss both route construction and local search algorithms, as well as metaheuristics.

For the HFVRPTW, to our knowledge no exact algorithms exist to solve the problem. Most contributions in literature about this problem focus on developing heuristics to solve the HFVRPTW, or variants of it (Belfiore & Yoshida Yoshizaki, 200 ; Brandão, 2011; Dell'Amico, Monaci, Pagani, & Vigo, 2007; Jiang et al., 2014; Koç, Bektaş, Jabali, & Laporte, 2015; Kritikos & Ioannou, 2013; Paraskevopoulos et al., 2008; Yepes & Medina, 2006). Those heuristics include for instance tabu search heuristics, scatter search heuristics or regret-based heuristics.

2.2. Time Slot Management in Attended Home Delivery

In Chapter 1 we already saw how *Time Slot Management* (TSM) can be defined. In this section we study its application in the context of attended home delivery. A pioneering study in the area of TSM in attended home delivery is the one of Campbell & Savelsbergh (2005), in which they introduce the *Home Delivery Problem*, which is in some way similar to our e-retailer case. As we have shown in Chapter 1, we can distinguish between strategical, tactical and operational levels of control in TSM. In literature we see that not always a distinction is made between the strategical and the tactical level, but we do see a clear distinction between the operational level and the other two levels.

At the two highest levels of control, strategical and tactical, we deal with decisions regarding the design of time windows, for instance the total number of time windows to use, their length, whether they should overlap or not and their pricing (Agatz, Campbell, Fleischmann, & Savelsbergh, 2008; Ehmke & Campbell, 2014). Several studies have been conducted in which the impact of decisions at these levels of control in TSM is investigated (Agatz, Campbell, Fleischmann, & Savelsbergh, 2011; Hernandez, Gendreau, & Potvin, 2017; Punakivi & Saranen, 2001). For example, Agatz et al. (2011) consider the tactical problem of selecting the set of time slots to offer in each of the zip codes in a service region. Hernandez et al. (2017) consider a tactical problem where a time slot combination for delivery service over a given planning horizon must be selected in each zone of a geographical area. Our focus is on the operational level of control in TSM, therefore we do not look further into the strategical and tactical level here. However, the topics are closely related and because the strategical and tactical decisions do strongly influence the operational flexibility, it is useful to have an idea about what is known in literature about the impact of strategical and tactical decisions in TSM.

Operational TSM is characterized by two periods in which different decisions must be taken: a so-called booking period and a service period (Bühler, Klein, & Neugebauer, 2016). During the booking period the provider and the customers that place an order agree on a time window for the delivery of the order. During the service period the orders are physically delivered. Studies of operational TSM have focused on decisions that need to be taken during the booking period. This is because decisions that are to be taken for the service period boil down to solving a variant of the VRP, which has already been studied extensively over the years. During the booking period the service provider needs to decide among others for each customer order which time windows to offer, whether to accept or reject the customer and whether to change the pricing of the time windows. Several studies have been carried out to get insight into the impact of those decisions. Some focus on the decision about the offering of time windows (Azi, Gendreau, & Potvin, 2011; Bent & Van Hentenryck, 2004; Campbell & Savelsbergh, 2005; Cleophas & Ehmke, 2014), where others focus on deciding how to price time windows dynamically for each customer to influence customer behavior and smoothen demand (Asdemir, Jacob, & Krishnan, 2009; Campbell & Savelsbergh, 2006; Chen & Chen, 2014; Klein, Neugebauer, Ratkovitch, & Steinhardt, 2017; Yang & Strauss, 2017; Yang, Strauss, Currie, & Eglese, 2016).

A study with an approach similar to ORTEC's current approach, considers how to solve an online capacitated vehicle routing problem with structured time windows (Hungerländer, Maier, Pöcher, Rendl, & Truden, 2018). Hungerländer et al. show that for the booking period we deal with an online problem, whereas for the service period we deal with an offline problem. Their online problem encompasses two steps: an insertion step and an improvement step. The insertion step consists of three parts. First a selection of available time windows is offered to a customer when the customer places an order for a certain day. From the options the customer selects a time windows which suits best. Finally, the order is inserted in the delivery routes for that day, in such a way that the order can be delivered within the selected time window. The improvement step takes the routes that have been built up until now and tries to rebuild them in a more efficient way, so more capacity is available to serve new customers. The offline problem consists of finding a more efficient way to schedule the delivery routes, with schedule built up so far as input. The offline problem is solved after the cut-off time. In other words, after all customers have placed their orders and no new order can be placed anymore. In Figure 2.3 we show the relation of several concepts in operational TSM which we encounter in literature on a timeline.

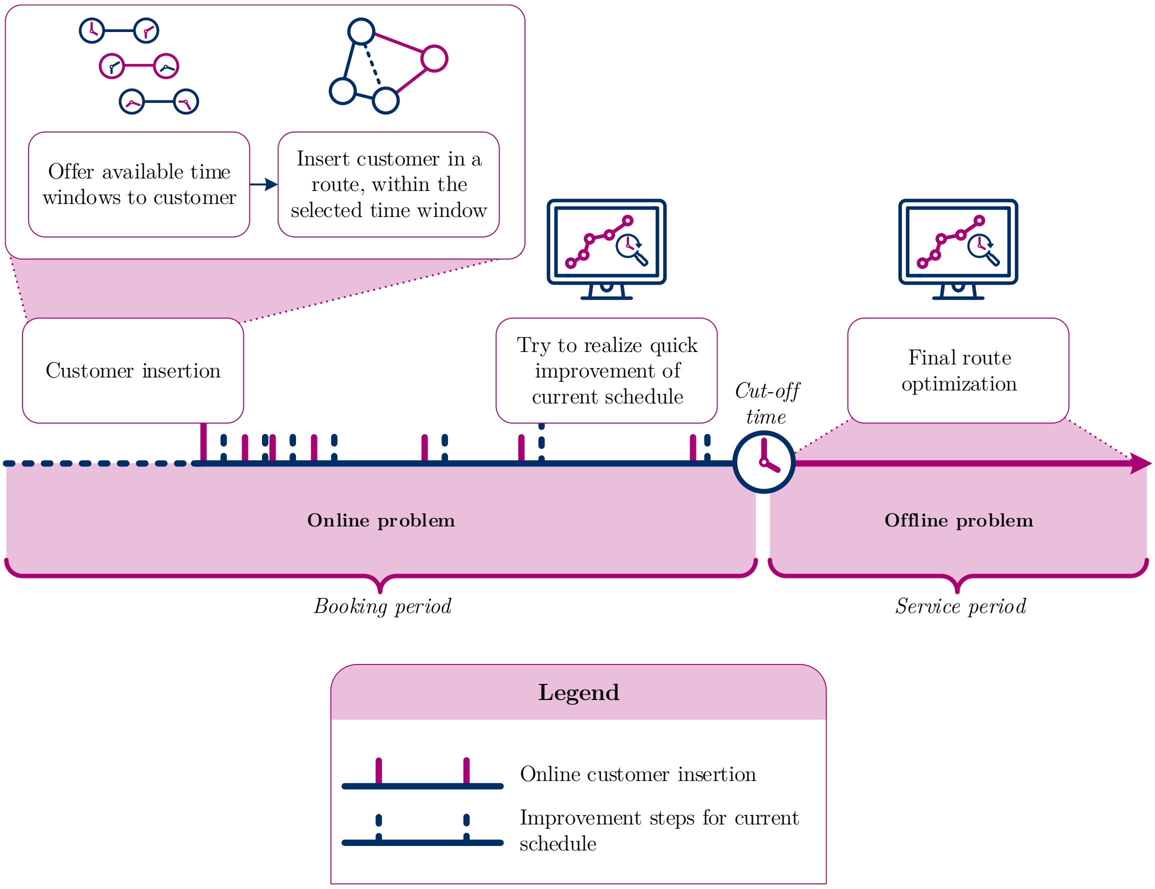

Offer available time windows to customer
Insert customer in a route, within the selected time window
Customer insertion
Try to realize quick improvement of current schedule
Cut-off time
Final route optimization
Online problem
Offline problem
Booking period
Service period
Legend
Online customer insertion
Improvement steps for current schedule

Figure 2.3. *Schematic overview of studied concepts in operational TSM on a timeline*

Another study with an approach that is similar to the set-up of our research project is focusing on developing a solution method to address an attended home delivery problem faced by an Italian provider of gas, electricity, and water services (Bruck, Cordeau, & Iori, 2017). Although the context is somewhat different from our context, many other characteristics are similar. The problem that Bruck et al. consider consists of three subproblems. First a so-called "time slot table" is created, with an assignment of resources per region for each time window. This assignment is based on expected service times for each region and is done offline. The second subproblem is the booking of the time slots, which is an online problem in which customers request service and select a time window according to their preference. The final subproblem consists of designing a routing plan offline, after all demand has become known. The difference with our project is that we build up the routing plan online already. Bruck et al. divide their demand area in regions instead, to which they assign resources. In that way, they try to ensure that feasible routes can be made for the smaller regions, after customers have booked their time windows based on the resource capacity for a time window in their region. By introducing penalties, representing the costs of outsourcing the service of the unserved customers, they take into account the possibility that not all customers can be served as a consequence of their approach. Bruck et al. use simulation techniques to evaluate the performance of their solution methodology, which we also plan to do.

Therefore, studies such as these of Bruck et al. (2017) and Hungerländer et al. (2018) provide valuable insights into how others approach problems similar to the one we are tackling.

2.2.1. Objectives Used for Measurement of Solution Quality

In this section we aim to give some insights into what literature tells us about how to compare different strategies or solution methodologies with regard to their impact on the solution quality. Our focus is on literature that considers problems in the context of TSM in attended home delivery. As mentioned in Chapter 1, in TSM we seek to find a proper balance between customer satisfaction and route efficiency. We also find these two concepts in literature about the measurement of the solution quality for problems in the context of TSM in attended home delivery.

In papers about the VRP that are not specifically linked to TSM, in most cases route efficiency is incorporated in the objective function, either in the form of driving times, or driving distances (Braekers et al., 2016). These two are generally assumed to be directly linked to the routing costs. For the HFVRPTW we see that besides the variable routing costs that are linked to driving times or distances, fixed vehicle costs are incorporated as well. These costs become interesting, due to the fact that the HFVRPTW considers a heterogeneous fleet, in which the vehicles have different fixed costs and other characteristics (Paraskevopoulos et al., 2008). Dondo & Cerdá (2007) point out that it is indeed important to consider both fixed vehicle utilization costs and variable operational costs such as driving times and distances. However, our research focusses on the application of the VRP in TSM, for that reason we now take a look into which objectives are common to use in this area. We focus specifically on how customer satisfaction and route efficiency are incorporated in those objectives.

We first take a look at the study of Bruck et al. (2017). They minimize routing costs to optimize route efficiency, and they do not explicitly optimize customer satisfaction. However, they adopt a certain threshold for quality of service to make sure to have a satisfactory performance regarding customer satisfaction. This threshold is determined upfront and serves as input to the problem.

The study of Hungerländer et al. (2018) adopts a somewhat different approach. They report several possible measures for quality of solutions, but they leave it to the reader to determine what they think is more important and what weights they give to the different performance indicators. At the end of their paper they summarize some findings based on the result tables, aiming to give the reader insight into their most interesting contributions.

In other studies of operational TSM in attended home delivery that we cited before, we see different ways of dealing with the balance between customer satisfaction and route efficiency. Some studies do not make use of an objective function, but they give resulting values for different performance indicators in result tables. These performance indicators give insight into for instance the percentage of served customers as an indication of the customer satisfaction, and the costs or profits as an indication for the route efficiency (Azi et al., 2011; Campbell & Savelsbergh, 2005; Yang et al., 2016). Other studies either focus on optimizing customer satisfaction, for example by maximizing the number of served customers (Bent & Van Hentenryck, 2004), or they focus on optimizing route efficiency, for instance by maximizing profits or minimizing routing costs (Campbell & Savelsbergh, 2006; Klein et al., 2017). An alternative objective is used by Cleophas & Emke (2014), who propose to maximize the overall value of orders given the transport capacity.

For our project the most suitable option seems to be the option in which we evaluate several performance indicators for different scenarios. It is then up to the reader to determine which strategy performs best in a specific practical realization of our context. In Chapter 3 we present the performance indicators that we use in this research.

2.3. Modeling Customer Choice Behavior in Time Slot Management

An important aspect for research in the context of TSM in attended home delivery is how to model the choice behavior of customers. In real life, every customer chooses for him- or herself and we just receive their choice as input. However, when we want to simulate the ordering process, we cannot look inside a customer's head to know which time window he or she would select in any given situation. Therefore, we must make assumptions to model the customer choice behavior in a proper way. Several studies that have been conducted in the field of TSM in attended home delivery report different methods to model this behavior. In this section we aim to get insights into what different methods have been used in literature, which enables us to make a well-founded choice of how to model customer choice behavior in our research project.

In their pioneering paper, Campbell & Savelsbergh (2005) use a quite straightforward method to model customer choice behavior. They assume that for each customer a so-called "time slot profile" is known, that identifies which time windows are acceptable for the customer. This profile is compared to the available time windows and if one or more available time windows are in the customer's profile, the time window with the highest expected profit is assigned to the customer.

In a later publication Campbell & Savelsbergh (2006) use a different approach. In this work they consider for each customer a known probability that the customer will select a certain time window for delivery, for all time windows. The probabilities for all time windows sum to 1 for each customer. This probability may be influenced by incentives in the pricing of time windows. A similar probability-based model is used by Yang & Strauss (2017).

Some other studies do not explicitly model customer choice behavior. In those studies the implicit assumption is that a customer only wants to select one time window, which is known, and that the only decision to be taken is whether to accept the customer or not (Azi et al., 2011; Bent & Van Hentenryck, 2004; Cleophas & Ehmke, 2014).

Studies that focus on dynamic pricing of time windows tend to make use of customer choice modeling concepts from the area of revenue management. Asdemir et al. (2009) for instance make use of utility models to derive the choice probability for each customer of selecting a certain time window. In their study about choice-based demand management and vehicle routing in e-fulfillment, Yang et al. (2016) also use this method, which they call the multinomial logit choice model. This model assumes that customers are utility maximizers, therefore the customers select the available time window that gives them the highest utility.

There are still many other methods to model customer choice behavior. For instance, Klein et al. (2017) use a general nonparametric rank-based choice model. In an extensive review, Nguyen, De Leeuw, & Dullaert (2018) give insights into the relation between customer behavior and order fulfillment in online retailing. Bruck et al. (2017) use four different simulation techniques. Two of those strategies assume that all time windows are equally popular among customers. A third strategy models customer preference by seeking for time windows that are as close as possible to the time window a customer selected in real life. The implicit assumption is here that a customer selected the time window that has his or her highest preference. A fourth strategy attributes a popularity translated into a choice probability to each time window. These choice probabilities are assumed to be equal for all customers.

We see that there is not really a standard way of modeling customer choice behavior for TSM in attended home delivery. All papers that we encountered choose, either implicitly or explicitly, some way to deal with customer choice behavior and then focus on other aspects of TSM in attended home delivery. Generally, no or very little attention is payed to the impact of the way in which customer choice behavior is modeled.

2.4. Conclusion

In this chapter we presented an overview of what is known in literature about the VRP (Section 2.1) in the context of a heterogeneous fleet and time windows and about TSM in attended home delivery (Section 2.2). Besides that, we saw what literature tells us about ways of measuring the solution quality of vehicle routing problems in the context of TSM in attended home delivery and how customer choice behavior can be modeled in this context. The insights we obtained answer our first set of research questions:

1) *What can we learn from literature…*
   - a. *…about vehicle routing problems in the context of a heterogeneous fleet and time windows?*
   - b. *…about time slot management in attended home delivery or similar application contexts?*
   - c. *…about the way of measuring the solution quality of vehicle routing problems in the context of time slot management in attended home delivery?*
   - d. *…about modeling customer choice behavior in time slot management?*

We found that in our context we deal with a version of the VRP that combines a heterogeneous fixed fleet and time windows, the HFVRPTW (Section 2.1). To solve this problem, usually heuristic techniques are used, because exact methods generally do not provide a satisfactory solution within a reasonable amount of time.

In our research we focus on the application of the HFVRPTW in TSM. We solve an offline version of the HFVRPTW to determine the final delivery routes at the beginning of the service period. Before this, during the booking period, we deal with an online version of the problem. When a customer order comes in during this period, we provide the customer with a choice of available time windows. The set of available time windows is based on the solution to the HFVRPTW built op until the moment the customer order comes in. Therefore, we consider the problem with which we deal during the booking period an online version.

The way in which studies of TSM measure the solution quality of solutions they find differs. Some studies focus on optimizing quantified indicators of customer satisfaction, whereas others focus on optimizing route efficiency. The option that seems most suitable in our context is evaluating several performance indicators for both customer satisfaction and route efficiency for different scenarios. Our results can then give insight into which strategy would perform well in different business scenarios, depending on which performance indicators are considered as the most important ones.

Customer choice behavior in TSM is modeled in different ways. Most methods incorporate choice probabilities for each time window for each customer in some way. These probabilities then determine whether a customer selects a time window to confirm the order or the customer chooses to withdraw the order instead.

3. Modeling the E-Retailer Case

In this chapter we present our approach for modeling the e-retailer case (Chapter 1). In Section 3.1, we present the use cases of the e-retailer case that we distinguish, and we provide practical examples for each of them. Then, in Section 3.2, we explain something about ORTEC's software solutions that we use in our model. Subsequently, in Section 3.3 we provide a formal definition of our problem. Finally, in Section 3.4 we discuss the hypotheses that we investigate in our computational experiments.

3.1. Use Cases

As already mentioned in Chapter 1, the e-retailer case has many applications in practice. There are several factors that can vary, such as the number of vehicle types, the number of order types, the driver capabilities, the number of central depots or the length of the booking period. These are only a few of the many factors of which we can study the impact. We use this section to define some use cases that we distinguish among ORTEC's clients, and therefore are most relevant to us given the scope of this research. Within the scope of this research it is not possible to analyze all use cases that we can think of for the e-retailer case. Therefore, we aim to construct our use cases in such a way that at least most practical situations encountered at ORTEC's clients can be categorized under one of the defined use cases.

We consider the e-retailer case with three vehicle types (1, 2, 3) for the delivery fleet, two order types (A, B) and one central depot for all use cases. For all use cases we consider drivers that can drive all vehicle types, so the only restriction that the drivers impose is that we may not use more vehicles than drivers that are available. Of course, in practice it may occur that an e-retailer has more than three vehicle types. However, in many cases these vehicle types show similar characteristics, for instance with regard to vehicle capacity, costs or vehicle speed. Therefore, we believe that for an initial exploration of our subject it is a good approximation to consider three vehicle types. In case a specific e-retailer has more vehicle types, these types may be categorized in such a way that the categories fit to the three vehicle types that we define below. Analogously, in many cases the customer orders can also be classified according to two order types in practical situations for ORTEC's clients that are similar to an e-retailer in our context. We restrict our analysis to only one central depot, because having to plan for multiple depots at the same time would considerably increase the complexity of the problem. As we are in an initial phase of exploring the field of operational TSM, considering multiple depots goes beyond our research scope. Also, it would be interesting for only some of the larger e-retailers that ORTEC has as (potential) clients. In most cases having one central depot is a realistic representation of reality for ORTEC's clients.

We construct the use cases by varying between vehicles that are dedicated and non-dedicated to the two order types that we consider. When we speak about a dedicated vehicle type, this means that vehicles of this type can only be used to deliver orders of either Order Type A, or Order Type B. Contrarily, vehicles of a non-dedicated vehicle type can deliver orders of any order type. Table 3.1 displays the main characteristics of the different vehicle types that we consider.

Table 3.1. *Main vehicle characteristics for each vehicle type*

Characteristic	Vehicle Type 1	Vehicle Type 2	Vehicle Type 3
Dedication	Dedicated to Order Type A	Dedicated to Order Type B	Non-dedicated
Capacity	Small vehicles	Small vehicles	Large vehicles
Costs	Cheap vehicles	Cheap vehicles	Expensive vehicles

Given the three vehicle types that we consider, we can distinguish between four combinations of vehicle types with which we can deliver all orders both from type A and type B. These combinations form the basis of our use cases and are shown in Table 3.2. As explained in Chapter 1, we consider e-retailers that employ a fixed number of drivers. As mentioned, we focus especially on cases where an e-retailer owns more vehicles than the number of drivers employed. In those cases, an e-retailer has to make a decision of how to assign the drivers to the delivery vehicles, to determine the composition of the delivery fleet that is available to deliver customer orders. The vehicles that are available to assign the drivers to are usually a result of strategical and tactical decisions that cannot be changed on an operational level. However, the decision of how to assign the drivers to the delivery vehicles is taken on an operational level and is not fixed until after the booking period. In this research we study the impact of the fleet composition resulting from the assignment of the drivers to the delivery vehicles, so the main bottleneck in our context is the number of drivers employed by an e-retailer. Therefore, we assume at least for the cheap and small vehicle types (Vehicle Type 1 & Vehicle Type 2) that if an e-retailer owns any vehicles of those types, the e-retailer owns a number that is equal to or greater than the number of drivers that the e-retailer employs. In that way we prevent that the number of cheap and small vehicles that is available, resulting from earlier decisions on a higher level of control, becomes a bottleneck in our problem. For the expensive vehicles (Vehicle Type 3) we do not make this assumption, because in the context of ORTEC's clients it is not realistic to assume that they always have enough large expensive vehicles available.

Table 3.2. *Overview of whether vehicles of each type are available (1) or not (0) for each use case*

Use Case	Vehicle Type 1	Vehicle Type 2	Vehicle Type 3
1	0	0	1
2	1	1	0
3a	1	0	1
3b	0	1	1
4	1	1	1

Use Case 3 is split up in 3a and 3b, because both cases are characterized by having one dedicated vehicle type and one non-dedicated vehicle type. For our context it does not really make a difference which of the two dedicated vehicle types we use, so for that reason we categorized both options under Use Case 3. For the sake of completeness, we now briefly discuss the four use cases from Table 3.2. However, in the remainder of our research we do not focus on all use cases, as we explain further in this section and in Section 3.4.

3.1.1. Use Case 1

Use Case 1 is a special use case in the sense that it is the only use case in our context in which we do not have a heterogeneous fleet. In this use case we only have vehicles of Vehicle Type 3, that can deliver both Order Type A and Order Type B. This basically reduces our use case to a version of the VRP with a homogeneous fleet. Instead of making all vehicles available during the booking period, we can just choose a number of vehicles equal to the number of drivers and use that subset of the delivery fleet as available vehicles. By doing so, we cope in a simple way with the restriction imposed by the number of drivers, which is by definition smaller than the number of vehicles in the delivery fleet. ORTEC's software solutions offer plenty of options for solving (rich) VRPs with a homogeneous fleet already. Therefore, this use case is not very relevant for us to consider, although it is a very relevant case in practice. Below we provide a practical example of this use case from the context of an e-retailer reselling electronics.

Online Electronics Reseller

An online electronics reseller called OER sells several kinds of electronic devices. The electronic devices can be divided into two categories, large electronic devices and small electronic devices. Due to the high value of the products sold, OER requires the customers to be at home when their order is delivered. The electronic devices are installed by specialized operators, of which OER employs a fixed number. OER owns several vehicles (equipped with installation tools) that may be used for the delivery of large electronic devices as well as small ones. On a daily basis OER plans the delivery routes, by assigning customer orders to the different delivery vehicles while taking into consideration how many operators are available.

3.1.2. Use Case 2

In Use Case 2 we consider the restriction that orders from Order Type A can only be delivered by Vehicle Type 1 and orders from Order Type B by Vehicle Type 2. All drivers can drive any vehicle type, which implies that we have a restriction on the sum of vehicles that can be used from Vehicle Type 1 and Vehicle Type 2. As a practical example of this use case we consider the example of a food delivery service from Chapter 1.

Food Delivery Service

A food delivery service called FDS offers customers the possibility to order food online, and have it delivered to their homes at the time they selected. To reach a wide range of customers, FDS offers many different types of food. To deliver the customer orders, FDS owns a fleet of several electric vans, as well as petrol vans. The petrol vans have a larger reach compared to the electric vans, and they are cheaper to purchase as well. Therefore, FDS prefers to use petrol vans whenever that is possible. However, the municipality decided that some areas in the city center (which they call emission zones) are not accessible for non-electric vehicles anymore. To serve the many customers in these emission zones, FDS needs the electric vehicles. FDS employs a limited number of drivers, which is smaller than the number of vehicles they own. To facilitate the planning process for the order deliveries, FDS decided to split the customer area into two parts. For customers that are located in the emission zones and their adjacent districts FDS always uses the electric vehicles to deliver the orders. In all other areas FDS uses the petrol vans to deliver the orders to their customers.

3.1.3. Use Case 3

Use Case 3 differs from the other use cases in the sense that for the delivery of one order type only non-dedicated vehicles can be used and for the other order type both dedicated and non-dedicated vehicles can be used. To be precise, we either have vehicles of Vehicle Type 1 and Vehicle Type 3, or vehicles of Vehicle Type 2 and Vehicle Type 3 available. Vehicles of Vehicle Type 3 offer more flexibility, but on the other hand they are more expensive. Therefore, we preferably use as many vehicles from Vehicle Type 1 or Vehicle Type 2 as possible. Recall the practical example of an online grocery store from Chapter 1 as a practical example for this use case.

Online Grocery Store

An online grocery store named OGS sells several kinds of products. Those products vary from food to beverages, personal care to household products, and other products that can be found in a regular grocery store as well. OGS delivers the orders to the homes of the customers, within the time windows that the customers selected. To this end, OGS owns a delivery fleet, composed of a fixed number of vans as well as a fixed number of small trucks that have a larger capacity than the vans. The vans are not suitable for refrigerated transport, but the small trucks have a cooling department and can therefore be used for refrigerated transport. OGS distinguishes between two order types, namely the orders that contain products that require refrigerated transport and the ones that do not contain such products. This implies that the first order type must always be delivered with a small truck, because OGS does not apply the practice of order splitting. The second order type can be delivered with any of the delivery vehicles that OGS owns. OGS employs a fixed number of drivers, smaller than the number of vehicles in the fleet, which can drive both vans and trucks. For each day on which customer orders must be delivered, OGS tries to make delivery routes that are as efficient as possible, while ensuring an as high as possible customer satisfaction.

3.1.4. Use Case 4

Compared to the first three use cases, Use Case 4 introduces the concept of having more than one vehicle type that can be used to deliver the orders for all order types. In Use Case 3 we needed the vehicles from Vehicle Type 3 to deliver orders from one of the order types. For Use Case 4 we have the choice of using only dedicated vehicles (just as in Use Case 2) or using a mix of dedicated and non-dedicated vehicles. An important decision for this use case regards whether to increase the fleet capacity for the current fleet composition by switching from a smaller vehicle to a larger one. Recall that the number of drivers available determines the number of vehicles available in the current fleet composition. Therefore, we do for instance not consider exchanging two small vehicles for one larger one, because in our context this is not relevant. Use Case 4 is the most complex use case in practice, and therefore the focus of our research is mostly on this use case. Below we illustrate this use case with a practical example from the context of an online sports store.

Online Sports Store

An online sports store named OSS sells many types of sport products, which can be divided into two categories. The first category contains general sports products, and the second category contains fan products of the local soccer club, which has a huge fan base internationally. OSS is one of the main sponsors of this soccer club and therefore OSS owns several delivery vans which are covered with advertisements for the club. OSS prefers to use these delivery vans to deliver any order that contains a product from their second category of products. For the other orders they prefer to use delivery vans that are covered with general advertisements of OSS, but do not differ in any other way from the vans with advertisements for the soccer club. However, sometimes there is a lot of customer demand and it becomes more efficient to use the small trucks that OSS owns, which can be used to deliver products from any category. Those trucks have more capacity than the vans, but they are more expensive. The drivers OSS employs are all licensed to drive both vans and trucks. OSS searches on a daily basis for the optimal composition of their delivery fleet to deliver the customer orders in the best way.

Figure 3.1 gives a visual representation of the configuration of each use case of the e-retailer case.

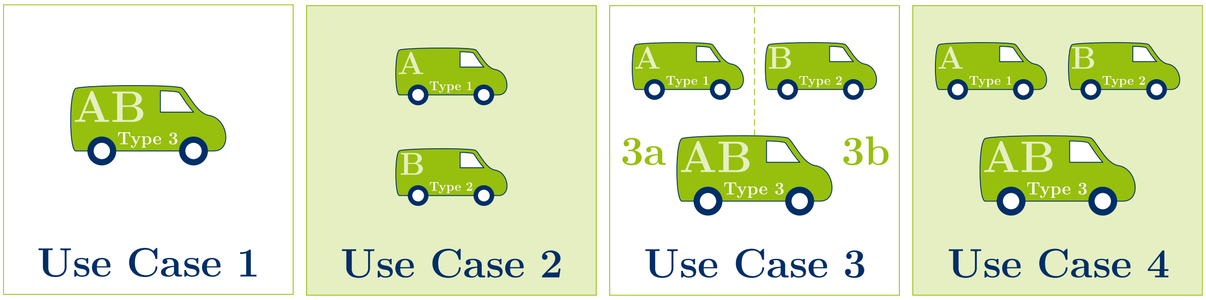


Figure 3.1. *Visualization of the characteristics of each use case*

3.2. ORTEC's Timeslotting Solution

In the previous section we already touched on what ORTEC's software solutions can do. In this section we explain how ORTEC's software solutions can be used during the ordering process of an e-retailer from our case. We discuss this before the formal definition of the problem we consider, because in our model we make use of ORTEC's software solutions for several decisions we need to take. Therefore, it is useful to know how these decisions are taken in ORTEC's software solutions.

This section is marked as confidential for the public version.

This section is marked as confidential for the public version.

3.3. Formal Problem Definition

In Chapter 1 we presented an overview of the background and the relevance of our problem in practice, as well as a short case description with practical examples. In Section 3.1 we presented different use cases of our problem. In this section we formally define the problem we are tackling in this research. We first present the problems that we consider our context, in Section 3.3.1. Then we present the entities that we use to model our problem in Section 3.3.2. After that, we describe the restrictions for our problem in Section 3.3.3. To conclude, in Section 3.3.4 we discuss what the objectives are that we consider when trying to find a solution to our problem. We express the objectives in terms of *Key Performance Indicators* (KPIs) that we use to evaluate the impact of the decisions we take based on the strategies we design.

3.3.1. Main Problems in the E-Retailer Context

In our context we deal with two main problems. We consider an online problem during the booking period and an offline problem at the start of the service period (see Figure 2.3 for an overview of these concepts in operational TSM). We consider a booking period of one day and a service period of one day. The online problem needs to be solved first and consists of a sequence of decisions, triggered upon each arrival of a customer order, as also displayed in Figure 3.3:

1) Do we accept or reject the customer?
2) In case we accept a customer, which time windows do we offer to the customer?
3) If the customer selects a time window, how do we insert the customer order into a delivery route?
4) Do we call an improvement algorithm to improve the delivery routes built up so far or not?

The first decision in the sequence of the online problem is the decision that we focus on in our research. The strategies that we design in Chapter 4 approach this decision in different ways, therefore we discuss this decision in more detail in that chapter. For all other decisions, we fix the way in which we take them, mainly based on how ORTEC approaches them currently in OTS, as also described in Section 3.2. Optimizing the way in which those decisions are now taken in OTS is out of scope for our research. Therefore, we discuss those decisions here, as by fixing the way in which they are taken they become part of our problem definition.

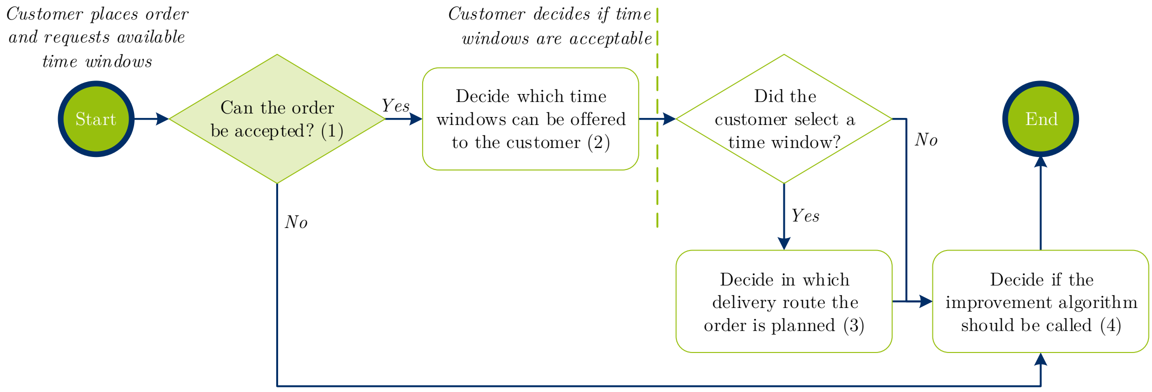

Customer places order and requests available time windows
Customer decides if time windows are acceptable
Start
Can the order be accepted? (1)
Yes
No
Decide which time windows can be offered to the customer (2)
Did the customer select a time window?
No
Yes
Decide in which delivery route the order is planned (3)
Decide if the improvement algorithm should be called (4)
End

Figure 3.3. *Decisions that must be taken online, triggered by an incoming customer order*

For the second decision, we offer all feasible time windows to the customer and let the customer select one of them. We make use of OTS to determine which time windows are available, so for each customer we place a time slot request. The time windows that OTS returns are then offered to the customer.

For the third decision we also make use of OTS. The customers are inserted in the cheapest way into our delivery routes, while respecting the time window the customer selected. The cheapest delivery route is determined by the taking the delivery route with the smallest amount of additional costs when inserting the customer into that delivery route.

The fourth decision we model by approximating the number of customers after which an improvement algorithm (or optimization algorithm) is called. Currently OTS calls this algorithm after a fixed period. If we would simulate this in the same way, we would need to perform real-time simulations that consume a lot of time. Therefore, we call the optimization algorithm after a fixed number of customers, which saves a lot of running time and enables us to do more experiments. We configure the optimization algorithms in CVRS to first maximize the number of customers that is served, then minimizing the total delivery costs and then minimizing the total distance of the delivery routes.

The offline problem that we solve is the HFVRPTW (see Chapter 2). As input we use delivery routes and customer orders that result from the online problem. We then improve the delivery routes by making use of CVRS, which implements ruin and recreate techniques, considering the specific restrictions for our context. We configure the algorithms in the same way as ORTEC currently does for clients similar to the e-retailers from our context (Section 3.2). The configuration therefore becomes part of our problem definition, as we take it as a given input, which we cannot modify. Therefore, we do not elaborate on this part of our simulation, neither do we try to improve the solution methods used. This configuration is out of scope for our research.

3.3.2. Model Entities

In our model of the e-retailer case we consider several entities, which we define in this section. These entities formally define our problem and serve as a basis for our simulation model, which we present in Chapter 5. The first entity that we consider is a *customer*. Figure 3.4 shows a graphical representation of the customer entity. We have a number of customers that sequentially place requests for available time windows during the booking period. Each customer has the following attributes:

- *Id* – The customer id is a unique number used to identify a customer.
- *Arrival time* – The arrival time represents the percentage of the length of the booking period that has elapsed when the customer places a request for available time windows, as a decimal value.
- *Location* – The customer location is a combination of a longitude value and a latitude value.
- *Time window preferences* – The time window preferences of a customer are represented by an ordered list of time windows, in the sequence of the customer's preference. The length of this list, i.e., the number of time windows that a customer is interested in, may differ for each customer and is therefore not fixed.

- *Order quantity* – The order quantity is expressed in kilograms and is used as a measure of the capacity that is required when delivering the order.
- *Order type* – The order type of a customer order is either Order Type A or Order Type B, as we consider the e-retailer case with these two order types. An order can never be both of Order Type A and Order Type B, so this attribute always has a single value.
- *Time window* – The time window can be used to represent the time window that the customer selected. This attribute does not necessarily have a value for all customers, because some customers may not select a time window and others may not be offered any time windows at all. So, this attribute only has a value for customers that are indeed planned in any delivery route.
- *Order duration* – The order duration is expressed in seconds and represents the time that is required to fulfill the order at the customer's home.

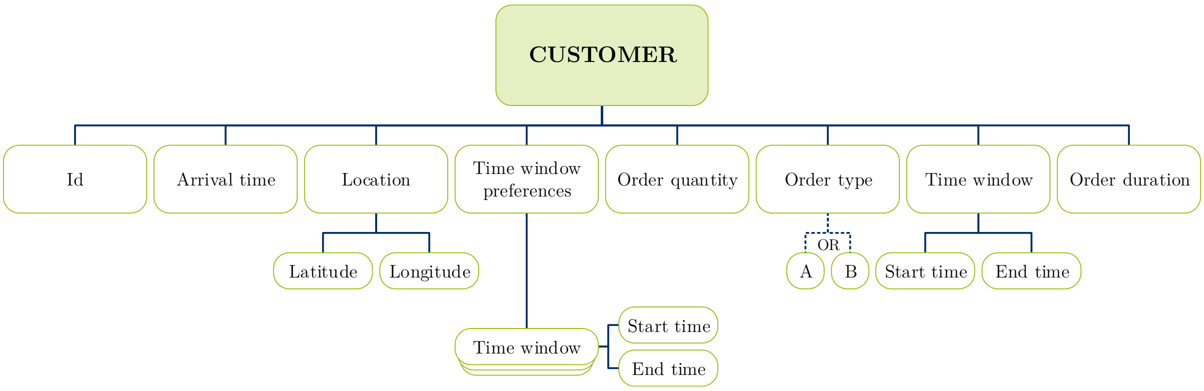


Figure 3.4. *The customer entity*

The second entity that we consider is a *driver*. Figure 3.5 gives a graphical representation of the driver entity. We have a number of drivers, who can drive the vehicles that an e-retailer owns. In our context all drivers are always available, we do not consider employee sickness or vacation. The drivers have two attributes:

- *Id* – The driver id is a unique number used to identify a driver.
- *Skills* – The driver skills are expressed as a list containing vehicle types and are used as an indication whether a driver can drive a certain vehicle type or not. In our context all drivers can always drive all vehicle types, i.e., Vehicle Type 1, Vehicle Type 2 and Vehicle Type 3.
- *Working hours* – The maximum number of hours that a driver is allowed to work. In our context this number is set to 8 hours.

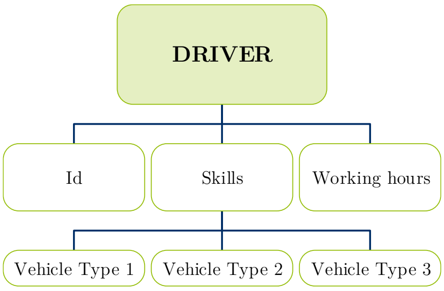


Figure 3.5. *The driver entity*

The third entity that we consider is a *vehicle*. Figure 3.6 presents a graphical representation of the vehicle entity. We have a number of delivery vehicles that can be used to deliver the customer orders that an e-retailer receives. Due to the nature of our problem, we always have more vehicles available than drivers. Therefore, in our context vehicle failures do not really form a problem that need to be considered during the booking period. Of course, during the service period an e-retailer needs to find a solution if a delivery vehicle breaks down while serving the customers, but that is out of scope for our research. So, all vehicles are considered to be always available for an e-retailer in our context. The vehicles have the following attributes:

- *Id* – The vehicle id is a unique number used to identify a vehicle.
- *Vehicle type* – The vehicle type is a number that indicates of which type from Section 3.1 the vehicle is. This can be either Vehicle Type 1, Vehicle Type 2 or Vehicle Type 3.
- *Costs* – The costs for using a vehicle are expressed in euros and are categorized as variable costs (per hour and per kilometer) and fixed costs (setup costs).
- *Capabilities* – The capabilities of a vehicle are used to indicate which order type can be delivered with this vehicle. They can be deduced from the vehicle type, but for the sake of clarity we consider this as a separate attribute. Vehicles from Vehicle Type 1 can only orders from Order Type A, vehicles from Vehicle Type 2 only orders from Order Type B. Vehicles from Vehicle Type 3 can deliver both order types.
- *Customers* – The list of customers contains all customers of which the orders are planned in this vehicle. If this list does not contain any customers, the vehicle is not used for delivery.
- *Route legs* – The list of route legs contains all the separate trips that the vehicle has to make to deliver the orders. If there are no customers in the list of customers, the list of route legs will also be empty. Each leg has a departure location, an arrival location and an associated distance in kilometers and a driving time in seconds. The driving time depends on the speed limit of the roads that the vehicle takes to go from the departure location to the arrival location. In our context we consider the speed limits as imposed for cars, and we use the distance and the driving times as provided by ORTEC's software solutions. If one or more customer orders are planned in the vehicle, the first leg always departs at the central depot (our next and last entity) and the last leg always arrives at the central depot.
- *Capacity* – The capacity of a vehicle is expressed in kilograms.

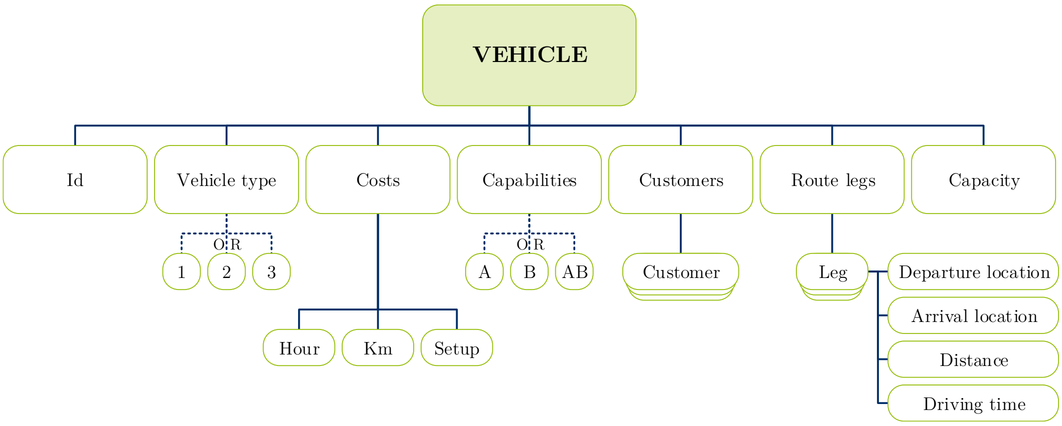


Figure 3.6. *The vehicle entity*

The last entity that we consider is a *depot*. Figure 3.7 shows a graphical representation of the depot entity. We have one central depot that we consider for the e-retailer case. In practical applications of the e-retailer case it is of course possible that an e-retailer has multiple depots, especially when there is a large spread of customer locations or there are too many customer orders to serve with only one depot. For now, we assume that this is not the case and we consider only one central depot, as we explained earlier in Section 3.1. A depot has the following attributes:

- *Id* – A unique name that identifies the depot
- *Location* – The depot location is expressed as a combination of a longitude value and a latitude value.
- *Time windows* – The list of time windows of a depot contains all the time windows that an e-retailer possibly offers to the customers when they want to place an order.

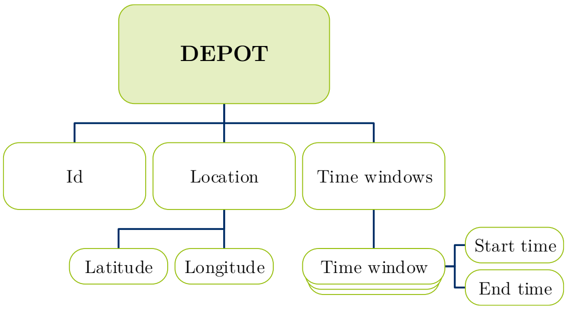


Figure 3.7. *The depot entity*

3.3.3. Restrictions

In our problem context we consider some important restrictions that may not be violated by the solutions we find:

- *A customer only places an order when at least one of the offered time windows matches a time window on the customer's preference list.*
- *Every customer that placed an order must be visited once in one delivery route* – This restriction prevents for instance the possibility to make use of order splitting.
- *Every customer order must be delivered within the time window selected by the customer* – Whether a customer order is delivered within the selected time window or not is determined by the moment that a delivery vehicle arrives at the customer.
- *Once a customer order is confirmed, the customer cannot be rejected anymore* – This restriction is essential during the booking period as well as after the booking period when the final delivery routes are formed. If we would not consider this restriction, we could simply "throw away" customer orders that were accepted in an earlier stage because they cause the delivery routes to be inefficient. Therefore, the fleet composition may only be changed if with the new composition all confirmed customer orders can still be delivered within their selected time windows.
- *Each vehicle in which one or more customer orders are planned requires exactly one driver* – This restriction prevents that a driver can be assigned to multiple vehicles, or multiple drivers can be assigned to one vehicle.
- *The total number of drivers used to drive vehicles may not exceed the number of drivers employed by an e-retailer* – If this turns out to be more efficient, the final solution may contain less delivery routes than drivers. However, this restriction makes sure that at least not more drivers are required than the number of drivers employed.
- *For each vehicle type, the number of vehicles that are used of that vehicle type may not exceed the total number of vehicles available from that type.*
- *Each vehicle that is used for delivery starts at the central depot* – This restriction ensures that before delivering any customer orders, a vehicle is loaded with the products that the customers ordered. The plan for a vehicle may be split up in trips, in that case the vehicle needs to return to the central depot at the end of a trip to load the products that the customers of the next trip ordered.
- *The sum of the order quantity for all customer orders planned in a vehicle may not exceed the capacity of the vehicle* – In case an e-retailer makes use of multiple trips of a vehicle during the service period, this restriction holds for each trip.
- *The duration of a delivery route carried out by a vehicle may not exceed the working hours of the driver* – The duration of a delivery route is defined as the sum of the order durations of all customers planned in a vehicle and the driving times for all legs of that vehicle. As mentioned, we consider drivers that are all available full-time (i.e., 8 hours per day).

3.3.4. Objectives and Performance Indicators

As we already mentioned several times, one of the most important challenges in TSM is to find a balance between customer satisfaction and route efficiency. Ideally our objective would be to maximize both of these, but that is an illusion in practice. Therefore, we try to find a satisfying balance between the two. As we pointed out in Chapter 2, many different KPIs can be used to quantify both the customer satisfaction and the route efficiency. We chose to evaluate our solution quality based on two main KPIs.

We use the first KPI, the percentage of customers that can be served, as a measure of customer satisfaction. Obviously, we want to maximize this percentage, as we assume that the more customers we can serve, the higher our customer satisfaction is.

The second KPI, the delivery costs per customer that is served, is the measure we use to quantify the route efficiency. As the revenues are also coming from the customers that are served, we decided to take the delivery costs per customer that is served as the main measure for route efficiency. Unlike for the percentage of customers that is served, we want to minimize this KPI.

When designing solution methods for our problem we focus in the first place on serving as many customers as possible, and in the second place on keeping the delivery costs per customer that is served as low as possible. So, whenever we speak about a better performance in the remainder of this report, we refer to a higher percentage of customers served and lower delivery costs per customer served. To calculate our two main KPIs and some other KPIs as well, we define the following variables and sets:

n	The number of customers that arrives during the booking period
C	The set of all the customers $i \in \{1, 2, ..., n\}$ that arrive during the booking period
q_i	The order quantity in kilograms of customer i
m	The total number of drivers employed by an e-retailer
V	The set of the vehicles v that are used to perform the delivery routes
q^v	The capacity in kilograms of vehicle v
V_j	The set of the vehicles of Vehicle Type $j \in \{1, 2, 3\}$ that are used to perform the delivery routes
C_v	The set of all the customers whose orders are planned in vehicle v
c_{tv}	The variable costs per hour for vehicle v
c_{dv}	The variable costs per kilometer for vehicle v
c_{sv}	The fixed setup costs for vehicle v
t_v	The duration in hours (order duration, driving time) of a delivery route performed by vehicle v
t_{vd}	The driving time in hours of a delivery route performed by vehicle v
d_v	The distance in kilometers of a delivery route performed by vehicle v

The percentage of customers that can be served is then given by Equation (3.1), in which we divide the number of customers that can be served by the total number of customers that arrives during the booking period:

$$\frac{\sum_{v \in V} |C_v|}{n} \times 100\% \tag{3.1}$$

To be able to compute the delivery costs per customer, we first need to compute the duration and the total distance of the delivery routes carried out by all vehicles in the set V. The duration t_v for a vehicle $v \in V$ is given by the sum of the order durations of all customers planned in that vehicle and the driving times for all legs of that vehicle (t_{vd}). The total distance d_v for a vehicle $v \in V$ is given by the sum of the distances for all legs of that vehicle. The delivery costs per customer that is served are then given by Equation (3.2):

$$\frac{\sum_{v \in V} (c_{tv} \times t_v + c_{dv} \times d_v + c_{sv})}{\sum_{v \in V} |C_v|} \tag{3.2}$$

Besides these two main KPIs, we keep track of several other KPIs to obtain more insights into the performance of our solution strategies. As measures for route efficiency, we keep track of the following KPIs:

- *The total driving time in seconds* → $\sum_{v \in V} (t_{vd} \times 3600)$
- *The total duration in seconds* → $\sum_{v \in V} (t_v \times 3600)$
- *The total delivery costs* → $\sum_{v \in V} (c_{tv} \times t_v + c_{dv} \times d_v + c_{sv})$
- *The total number of delivery routes that are formed* → $|V|$
- *The number of vehicles used for delivery routes per vehicle type* → $\left|V_j\right|$
- *The average utilization of the vehicles (in terms of load capacity)* → $\frac{\sum_{v \in V} \left(\sum_{i \in C_v} (q_i / q^v)\right)}{|V|} \times 100\%$

As measures for customer satisfaction, we keep track of the KPIs below:

- *The number of customers that are served in the delivery routes* → $\sum_{v \in V} |C_v|$
- *The average number of time windows offered to customers during the booking period*
- *The average response time in seconds when a customer requests available time windows*

Of course, we may keep track of many other KPIs. Nevertheless, most of them are likely to be somehow related to the KPIs we selected. To keep the scope of our analysis manageable we select this subset of KPIs, which in our opinion (based on what we see in literature) give a good indication of the quality of a final solution to our problem.

3.4. Hypotheses

The examples provided for the use cases in Section 3.1 already indicate that the e-retailer case has a wide range of possible applications. Consequently, we could keep ourselves busy researching many of those. To scope our exploration of this subject, we formulate four hypotheses that we investigate in the remainder of this research. We first state the hypotheses in this section, and in Chapter 5 we present the scenarios for which we test the hypotheses by analyzing computational results of our experiments.

Each hypothesis is linked to one or more of the use cases that we explained earlier. The hypotheses focus on Use Case 2 and Use Case 4, because these two use cases are most relevant in practical applications for ORTEC's clients. As explained in Section 3.1, ORTEC's software solutions are already able to deal with Use Case 1 in a good way, so there is no big need for research on this use case. Investigating Use Case 3 may be interesting from a knowledge-gaining point of view but is less relevant for ORTEC's practice. Besides that, preliminary results indicate that strategies for changing the fleet composition can have a more significant impact for Use Case 2 and Use Case 4 than for Use Case 3. Therefore, we scope our hypotheses to Use Case 2 and Use Case 4.

Another important consideration on which the hypotheses are based is that in this research we focus on cases in which on average more customer orders arrive than can be served with the maximum fleet capacity. The maximum fleet capacity is in this case not determined by the total capacity of all vehicles that an e-retailer owns, but by the restriction imposed by the number of drivers that is available. In cases where on average less customers arrive than can be served with the maximum fleet capacity, the incentive to change the fleet composition dynamically during the booking period is less or may even be absent.

We clarify this with the following example: imagine we have a case with incoming customer orders of two types. We have two vehicle types, each of which is dedicated to an order type, just as in Use Case 2. The number of vehicles of each type equals the number of drivers that is available, so we have a number of vehicles in the fleet equal to twice the number of drivers. If we now assume that typically we can serve all customers that want to place an order, there is hardly any need to distinguish between whether a vehicle type can deliver a certain order type or not. We can instead just assume that all vehicles can deliver any order type, and only after the booking period create a routing plan in which we consider whether a vehicle type can deliver an order of a certain type or not. This is due to the fact that, by definition, we have more capacity than needed to deliver all orders, so in most cases our strategy is likely to be able to make a feasible routing plan. Of course, we could investigate this subject in a deeper way, but we scope our research further to cases in which on average an e-retailer is not able to serve all customer demand. These cases have a higher practical relevance given the context of ORTEC's clients.

Finally, in our research we consider the performance of three strategies, that can either be static or dynamic strategies. A strategy is considered to be static in case the strategy does not change the fleet composition during the booking period, i.e., all drivers are assigned to delivery vehicles at the beginning of the booking period and not re-assigned after that anymore. A strategy is considered to be dynamic when the strategy gives the possibility to re-assign the drivers to other vehicles during the booking period, thus changing the fleet composition of the delivery fleet. We consider the following strategies:

1) A static strategy named *ORTEC Base Strategy* (OBS)
2) A myopic dynamic strategy named *Myopic Strategy* (MYS)
3) A smart dynamic strategy named *Balanced Strategy* (BAS), that may decide to reject a customer whenever this is expected to be more profitable compared to accepting the customer (i.e., offering available time windows to the customer). OBS and MYS, on the contrary, only reject customers to whom no available time windows can be offered.

In Chapter 4 we further explain all characteristics of these strategies, but we introduce them here already to keep them in mind when reading the hypotheses.

3.4.1. Hypothesis 1

The first hypothesis is quite straightforward. Although we may be stating the obvious here, we need to show its correctness by doing computational experiments before we continue with testing other hypotheses. The hypothesis sounds as follows:

> *In applications of Use Case 2 where forecasts (based on historical data) for the ratio of customer orders of Order Type A and Order Type B are not accurate, using a myopic dynamic strategy leads to a better performance compared to a static strategy. However, when the forecasts are accurate, using a myopic dynamic strategy does not necessarily improve the performance.*

For this hypothesis we deliberately consider Use Case 2, because in this use case we filter out the effect of having the possibility to increase the capacity of the fleet composition (by re-assigning drivers to larger vehicles). In Use Case 4 we have the possibility to re-assign drivers that were assigned to a vehicle of either Vehicle Type 1 or Vehicle Type 2 to a vehicle of Vehicle Type 3 with a larger capacity. This effect may be a disturbing factor when we want to show only the impact of different ratios between the two order types and dealing dynamically with those. Hence, we focus on Use Case 2 for this hypothesis and we incorporate Use Case 4 in our other hypotheses. When we speak about different ratios, we refer to forecasting a ratio based on historical data that differs from the ratio observed in practice.

The primary goal we want to attain by investigating this hypothesis is proving the need for strategies that are able to dynamically change the fleet composition during the booking period. The objective of this possibility to change the fleet composition is to achieve a better performance in terms of customer satisfaction and route efficiency. Recall that we quantify those measures by keeping track of the percentage of customers that we can serve and the costs per customer that is served. To show the need for dynamic strategies, we investigate both the impact of using a dynamic strategy in cases where forecasts with regard to the ratio of the order types are accurate and in cases where they are not.

3.4.2. Hypothesis 2

The second hypothesis builds further on the first one. It introduces the impact of a dynamic strategy that does not only consider changing the fleet composition during the booking period, but also may decide to reject a customer order when this order is considered not to be profitable enough:

> *In applications of Use Case 2 and Use Case 4, the performance of the myopic dynamic strategy mentioned in the first hypothesis can be improved by introducing a smart dynamic strategy.*

For this hypothesis we again consider both cases with accurate forecasts and inaccurate forecasts. Considering Use Case 4 introduces the impact of being able to increase or decrease the fleet capacity implied by the fleet composition, as explained in Section 3.4.1, by exchanging smaller vehicles for larger ones or vice versa. This may be very interesting for practical applications for ORTEC's clients. Our main goal when putting this hypothesis to a test is to see whether we can justify the expectation that a smart dynamic strategy shows a better performance compared to a myopic dynamic strategy, by taking into account forecasts regarding the remainder of the booking period.

3.4.3. Hypothesis 3

The third hypothesis is relevant from a more practical point of view. It introduces the concept of an initial fleet composition. The initial fleet composition is determined by the assignment of the drivers to delivery vehicles before the start of the booking period. As long as we do not exchange any vehicle yet during the booking period, the vehicles in the initial fleet composition determine whether we can offer available time windows to the customers that arrive.

Especially when we have a static strategy that does not change the fleet composition during the booking period, the initial fleet composition has a large impact on the result in the end. For instance, if for Use Case 4 we have a lot of customer demand and our initial fleet composition does not contain any vehicles of Vehicle Type 3, the capacity of the initial fleet composition is not enough to serve all demand. However, we cannot make use of the vehicles of Vehicle Type 3 anymore, because our strategy is static. If the initial fleet composition in such cases is not tuned properly according to the customer demand that comes in, the performance in the end will be worse compared to when we have a dynamic strategy. The reason for this is that a dynamic strategy can correct the bad decisions regarding the initial fleet composition. In Chapter 4 we look further into this concept of the initial fleet composition. Our third hypothesis is the following:

> *In applications of Use Case 4, we achieve a better performance when we use a smart dynamic strategy with an initial fleet composition that uses vehicles that are as cheap as possible, compared to a static strategy that starts with a composition that has a total fleet capacity that is as large as possible and is therefore costlier.*

Recall that in this research we focus on cases in which on average more customer orders arrive than can be served with the maximum fleet capacity. The question may then rise why there is a need for a dynamic strategy, that makes use of complex algorithms which are difficult to implement. This question is the reason that we need to validate our third hypothesis.

We believe that a smart dynamic strategy with an as cheap as possible initial fleet composition can outperform a static strategy with maximized fleet capacity, even though the dynamic strategy is likely to end up with a composition with maximum capacity as well. The main challenge for an e-retailer lays in the fact that the demand is not known upfront and may have a large fluctuation. On days where demand is not sufficient to require the maximum fleet capacity for delivery, the dynamic strategy recognizes this and will not use more large and expensive vehicles than required. Using a static strategy with maximized fleet capacity for such days would result in higher delivery costs. The reason for this is that the static strategy would blindly make available as many large and expensive vehicles as possible.

Besides this, our smart dynamic strategy also implements smart decision mechanisms regarding whether to accept a customer or not. This may result in a better performance as well compared to a static strategy that may not reject customers based on whether they are unattractive. So, we may ask ourselves whether it is worthwhile to implement such a strategy. By analyzing this hypothesis, we aim to confirm our expectation that there actually is a benefit in implementing a smart dynamic strategy instead of a using static strategy with maximized fleet capacity.

3.4.4. Hypothesis 4

The last hypothesis that we put to a test is in a different realm compared to the other three. We introduce the impact of different distributions of customer arrivals over the booking period. We want to investigate if the smart dynamic strategy we design is able to show a stable performance, regardless of how the distribution of the customer arrivals over the booking period. Besides that, we want to obtain insights into the effect of forecast errors regarding this distribution. Our expectation is that these errors have a negative impact on the performance. Therefore, we formulate the following hypothesis:

> *In applications of Use Case 4, a smart dynamic strategy achieves a stable performance for different distribution patterns of customer arrivals over the booking period, as long as we have accurate forecasts of this spread. However, when forecasts are not accurate, the performance of a smart dynamic strategy may deteriorate significantly.*

The main purpose of analyzing this hypothesis is to show whether we are able to design a smart dynamic strategy that shows a stable performance even when the distributions of the arrival spread are different. Besides that, by validating this hypothesis we aim to show the importance of having accurate forecasts for the distribution of customer arrivals over the booking period. We distinguish between consistently over-forecasting or under-forecasting the percentage of customers that has arrived at a certain point of time during the booking period, and having a fluctuating forecast error. In some part of the booking period we may then be over-forecasting, and other parts we may be under-forecasting.

By putting this hypothesis to a test, we aim to provide insights in these issues and study the robustness of the smart dynamic strategy we design. In Chapter 5 we provide more details regarding the distributions we consider and the experiments we perform with regard to the forecast errors.

3.5. Conclusion

In this chapter we defined several practical use cases of the e-retailer case (Section 3.1) and we formally defined the problem that we are tackling (Section 3.3). We also gave an overview of what ORTEC's software solutions can already do in the realm of our problem context (Section 3.2). Finally, we stated several hypotheses which we investigate in the remainder of this research (Section 3.4). After this, we can now answer our second set of research questions:

2) *How can we model different use cases of the e-retailer case, to test the performance of the different strategies that tell us how to deal with an unfixed fleet composition during the ordering process?*
   a. *Which use cases of the e-retailer case should we consider?*
   b. *How can we formally define the problem we are tackling?*
   c. *How do we measure the quality of a final solution?*
   d. *Which hypotheses do we put to a test in our simulations?*

We consider the e-retailer case with customer orders that may be categorized into two order types and delivery vehicles that can be categorized into three vehicle types. The first two vehicle types can only be used to deliver one order type, and they are small but cheap. The last vehicle type is large and can be used to deliver both order types but is expensive. Combining the vehicle types in such a way that we are always able to deliver all order types results in four different use cases: in the first use case only the large and flexible vehicles are available, in the second use case only the small and dedicated vehicles are available, in the third use case vehicles of one dedicated vehicle type combined with the large and flexible vehicles are available and in the fourth use case vehicles of all types are available.

In our problem we need to deal with several decisions. First, we need to decide whether or not to accept a customer. If we accept a customer, we need to decide which time windows to offer to the customer. In case the customer selects a time window, we need to decide in which delivery route we plan the customer order. Finally, we need to decide whether we call the improvement algorithm for a quick optimization of the delivery routes after planning the customer order. The main focus of our solution strategies is on the first decision, the other decisions we take in a way based on how ORTEC currently takes them.

We model the e-retailer case by making use of four entities. The first entity is a *customer*, which has the following attributes: an id, an arrival time, a location, a list with time window preferences, an order quantity, an order type, a selected time window and an order duration. The second entity is a *driver*, which has an id, skills and a maximum number of working hours as attributes. The third entity is a *vehicle*, which has the following attributes: an id, a vehicle type, costs, capabilities, a list of customers, a list of route legs and a capacity. The last entity we consider is a *depot*, which has an id, a location and a list of all time windows that can possibly be offered to the customers as attributes. We deal with several restrictions that all somehow affect these entities.

Our first objective is to achieve an as high as possible customer satisfaction and our second objective is to achieve an as high as possible route efficiency. As a measure of customer satisfaction, we keep track of the percentage of customers that are served, which we try to maximize. As a measure of route efficiency, we keep track of the delivery costs per customer that is served, which we try to minimize.

We formulate four hypotheses that we put to a test in the remainder of this research. The first hypothesis is formulated to show the need for strategies that can dynamically change the fleet composition during the booking period. The second hypothesis is formulated to show that by introducing smart decision mechanisms, we can realize improvements in terms of the KPIs we consider. The third hypothesis is formulated to show why it is worthwhile to implement smart dynamic strategies in practice and the last hypothesis is formulated to analyze the robustness of the strategies we design.

4. Solution Approach and Strategy Design

In this chapter we describe our solution approach. We first describe the problems that we need to address in our strategies in Section 4.1. Section 4.2 presents the first of three strategies that we consider in this research. This first strategy (OBS) is a static strategy that ORTEC could implement without changing anything in OTS. Our two other strategies are dynamic and do change the fleet composition during the booking period, aiming to increase the number customers served in that way. Our first dynamic strategy is the myopic strategy (MYS), which we describe in Section 4.3. Section 4.4 describes our second dynamic strategy, which is the balanced strategy (BAS).

4.1. Decisions

In Chapter 3 we pointed out that our strategies focus on the decision whether to accept a customer or not and Figure 3.3 gives the general context of this decision. In this section we dig further into this decision. We focus on the decisions regarding the fleet composition that need to be considered before finally either accepting or rejecting a customer. We distinguish between two important types of decision moments for our strategies. First, a strategy needs to provide a method to determine the initial fleet composition before any customer order comes in (Section 4.1.1). Second, our strategy needs to provide a method that determines whether to change the fleet composition or not upon a customer order and whether or not the customer order is rejected (Section 4.1.2).

4.1.1. Initial Fleet Composition

During the booking period, a route plan is built up by updating preliminary routes each time a customer order comes in. Before the booking period, an e-retailer needs to determine an initial composition of the delivery fleet based on the number of drivers that are available. Based on this initial composition, time windows will be retrieved for incoming customer orders.

The main question when determining the initial fleet composition is how to assign the available drivers to the vehicles that an e-retailer owns. Recall that in our research context, the number of drivers is always smaller than the number of vehicles in the delivery fleet. Therefore, by definition, only a subset of all available vehicles can be used to deliver the customer orders. Ideally, we want the initial fleet composition to be tuned perfectly to the customer orders that will come in during the booking period. However, this may turn out to be quite a complicated task for an e-retailer.

In the context we consider, an e-retailer should consider things such as the expected ratio of the demand for each order type. Especially in cases where the initial fleet composition is fixed after the booking period starts, the impact of this composition becomes quite large. The strategies we design all provide a framework to determine the initial fleet composition at the start of the booking period, based on historical customer data.

To make sure that we can compare the performance of our strategies based on the same starting position, we determine the initial fleet composition in the same way for all our strategies. In all cases we try to make use of an as cheap as possible initial fleet composition. However, as required for Hypothesis 3, we also design a method to determine the initial fleet composition in such a way that we maximize the available fleet capacity. In Section 4.2 we look further into the methods that we use to determine the initial fleet composition for our first strategy. As the other two strategies make use of the same method, we refer to Section 4.2 when explaining how these strategies take their decisions.

It is important to note that the initial fleet composition consists of the vehicles that are made available for order delivery. This does not necessarily mean that all vehicles that are available in the initial fleet composition or even the final fleet composition, are actually used for the delivery of customer orders. For instance, suppose we have a fleet composition consisting of 10 vehicles. At a certain moment during the booking period, our routing plan is such that only 5 vehicles are required to deliver all customer orders confirmed so far. This means that 5 vehicles are available, but are not used. It also implies that only 5 of the 10 available drivers need to work. It is important to keep in mind that our strategies do not take the decision of which vehicles are actually used for delivery. This decision is taken by OTS and CVRS, as explained in Chapter 3. Our strategies only provide OTS and CVRS with the vehicles that are available, taking into consideration the restrictions with regard to the number of drivers employed by an e-retailer in our context.

Having said this, we must also be aware of the fact that in case we have an initial fleet composition with maximum capacity (Hypothesis 3), OTS and CVRS try to make use of as few vehicles as possible when constructing the routing plan. This may solve the problem of ending up with too many vehicles that have a low utilization in case we use a static strategy and demand turns out to be less than expected. However, it does not solve the problem in case the division of the capacity for each order type in the initial fleet composition is an inaccurate representation for the ratio of the actual demand per order type. In the latter case, the result will be a poor performance for a static strategy, because customers will have to be rejected due to a lack of capacity for their order type. A dynamic strategy does not have this problem, because in this case and in general cases it is able to “repair” the initial fleet composition by re-assigning drivers to other vehicles according to observations during the booking period. In the following section about the customer acceptance we look further into the difference between a static and a dynamic strategy.

4.1.2. Customer Acceptance

The customer acceptance decision is a more complex decision. Currently, OTS is configured in such a way that as soon as a feasible time window can be offered, the customer is accepted. Offering a feasible time window means that the customer order can be inserted into at least one delivery route, within at least one time window, without violating any restrictions. Examples of such restrictions can be the capacity of the vehicles or selected time windows of customer orders that are already confirmed. Note that accepting a customer does not necessarily mean that the customer also confirms the order by selecting a time window. It may occur that we accept a customer, but that the customer is not satisfied with the time windows that are offered and decides to not confirm the order. In Chapter 5 we explain in more detail how we model the customer choice behavior in our experiments.

Within our research context we introduce two new concepts with regard to the customer acceptance decision. First, we extend the decision by considering to reject unattractive customer orders. This consideration is relevant in our context, because we expect that on average more customer orders arrive than an e-retailer can deliver even if the largest possible vehicles are used. It may turn out to be beneficial to reject some unattractive customer orders, so in the end more customers can be served. Currently OTS is configured to not consider rejecting unattractive customer orders, customers are only rejected when no feasible time windows can be offered.

The second extension of the customer acceptance decision is that we consider to change the fleet composition. We need to consider several aspects when we take this decision. We may for instance consider trying to change the fleet composition upon each customer order, or only try changing the fleet composition if a customer order cannot be accepted. We categorize both these options as dynamic strategies. We may also choose to not try changing the fleet composition at all, which we categorize as a static strategy.

Another aspect we need to consider is how to change the fleet composition. We may choose to swap one vehicle at a time, but we may also swap multiple vehicles at once. Note that a swap of a vehicle is equivalent to re-assigning a driver to another vehicle. For that reason, we speak about swapping vehicles and not about adding vehicles. Recall that OTS and CVRS determine whether vehicles are used. Our fleet composition only tells the software which vehicles can be used. Therefore, our fleet composition always consists of the same number of vehicles as the number of drivers that an e-retailer employs, and we swap vehicles instead of only adding vehicles to or removing vehicles from the fleet composition.

A final aspect that we need to take into consideration is that in case we change the fleet composition, we must be able to re-plan all customer orders that are already confirmed so far in the new vehicles.

4.2. ORTEC Base Strategy (OBS)

In this section we describe we describe our static base strategy called *ORTEC Base Strategy* (OBS). As mentioned before, OBS can already be implemented in ORTEC's software right now and for that reason it is called a base strategy. The strategy determines an initial fleet composition, after which customer orders can come in. The fleet composition remains unchanged during the entire booking period, so if a customer cannot be offered any feasible time window, the customer order is simply rejected. This makes the quality of the initial fleet composition important for the performance of this strategy. We explain here in detail how we determine the initial fleet composition for OBS.

To minimize the delivery costs, we prefer to use vehicles that are as cheap as possible. Unfortunately, the cheap vehicles are the ones with the smallest capacity in our context. As we explained in Chapter 3 for Hypothesis 3, this would not be very smart when we make use of a static strategy, given that on average we have more customer demand than we can serve. Therefore, we distinguish between two methods to determine the initial fleet composition for OBS. The first method constructs an initial fleet composition that is as cheap as possible. This method is used in OBS mainly for benchmarking purposes, so we can filter out the effect of different initial fleet compositions when we compare the performance of the different strategies. As mentioned in Section 4.1, this first method is used for all strategies to have a benchmark with the same starting point for all strategies.

The second method used to determine the initial fleet composition for OBS starts with an initial fleet composition that has an as large as possible total capacity. This is a more realistic method from a practical point of view, as we expect that on average we are not able to serve all customer demand.

Both methods rely on historical data of customer orders. We use historical data to estimate the expected ratio of the total order quantity for Order Type A and the total order quantity for Order Type B. We take the order quantity here to determine the ratio instead of the number of orders, because the orders may vary a lot in order quantity. Let C_h^{A} be the set containing all historical customers (i) that have ordered something of an e-retailer in the past with Order Type A, and C_h^{B} the same set for Order Type B. Recall from Chapter 3 that q_i stands for the order quantity of customer i. The ratio of the total order quantity for Order Type A and Order Type B is then given by Equation (4.1) and Equation (4.2).

$$\%\ \mathit{Order\ Type}\ \mathrm{A} = \frac{\sum_{i \in C_h^{\mathrm{A}}}(q_i)}{\sum_{i \in C_h^{\mathrm{A}}}(q_i) + \sum_{i \in C_h^{\mathrm{B}}}(q_i)} \times 100\% \tag{4.1}$$

$$\%\ \mathit{Order\ Type}\ \mathrm{B} = \frac{\sum_{i \in C_h^{\mathrm{B}}}(q_i)}{\sum_{i \in C_h^{\mathrm{A}}}(q_i) + \sum_{i \in C_h^{\mathrm{B}}}(q_i)} \times 100\% \tag{4.2}$$

Based on the calculated ratio, we divide the number of drivers over the order types. Consider for instance a case in which we have ten drivers, and a ratio of 56%:44%. Six drivers are then assigned to drive vehicles that can deliver Order Type A, and four drivers are assigned to drive vehicles that can deliver Order Type B. In case the ratio results in non-integer values for the number of drivers, we round the number of drivers to the nearest integer, as also done in the example above. If this results in using one driver more than available, we reduce the number drivers assigned to the order type with the smallest percentage with one driver. This may for example be the case when the ratio is exactly equal to 45%:55%. The difference between the two methods that we use lies in the vehicles that are selected to assign the drivers to.

The first method selects vehicles that are as cheap as possible. So, first we make a list for each order type with all vehicles of vehicle types that can deliver that order type. We then sort this list in such a way that it starts with the cheapest vehicle (in terms of setup costs) and ends with the most expensive one. As explained in Chapter 3, in our context we always have a number of vehicles equal to the number of drivers for the small and cheap vehicle types. In case we have ten drivers, this means that on the list for Order Type A the first ten vehicles are of Vehicle Type 1, and on the list for Order Type B the first 10 vehicles are of Vehicle Type 2. Therefore, for both Use Case 2 and Use Case 4 our initial fleet composition always only consists of small and cheap vehicles when using our first method. Recall from Chapter 3 that m stands for the total number of drivers that an e-retailer employs. Let $k \in \{A, B\}$ indicate the considered order type, and m_k the number of drivers assigned to order type k. Figure 4.1 gives an overview of how the first method works.

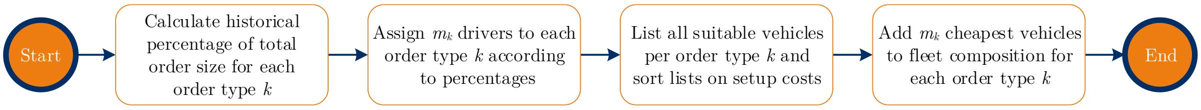

Start
Calculate historical percentage of total order size for each order type k
Assign m_k drivers to each order type k according to percentages
List all suitable vehicles per order type k and sort lists on setup costs
Add m_k cheapest vehicles to fleet composition for each order type k
End

Figure 4.1. *The first method to determine an initial fleet composition, used by all strategies*

The second method starts with adding as many large vehicles to the initial composition as possible. If the number of large and flexible vehicles (Vehicle Type 3) that an e-retailer owns is smaller than the number of drivers available, the remaining drivers are assigned to an order type according to the ratio as calculated with Equation (4.1) and Equation (4.2). Note that for Use Case 2 both methods result in the same initial fleet composition, because an e-retailer does not own any large and flexible vehicles in this use case. Figure 4.2 gives an overview of how the second method works. The symbols m, k, and m_k have the same meaning as described above.

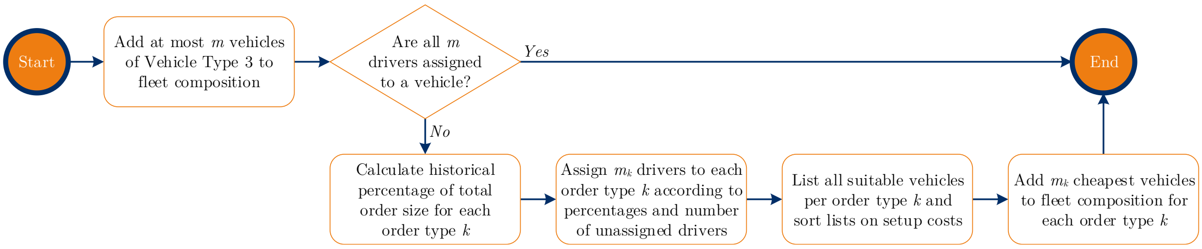

Start
Add at most m vehicles of Vehicle Type 3 to fleet composition
Are all m drivers assigned to a vehicle?
Yes
No
End
Calculate historical percentage of total order size for each order type k
Assign m_k drivers to each order type k according to percentages and number of unassigned drivers
List all suitable vehicles per order type k and sort lists on setup costs
Add m_k cheapest vehicles to fleet composition for each order type k

Figure 4.2. *The second method to determine an initial fleet composition, used by OBS (Large Initial Fleet)*

4.3. Myopic Strategy (MYS)

In this section we describe our second strategy which we call *Myopic Strategy* (MYS). MYS makes use of the first method to determine the initial fleet composition, as presented in Figure 4.1. MYS differs from OBS when it comes to the decisions regarding customer acceptance. Where OBS rejects customer orders whenever no time windows are available to be offered to the customer, MYS does not do this immediately. Instead, MYS tries to accept every customer order by attempting to change the fleet composition in case no time windows can be offered given the current composition.

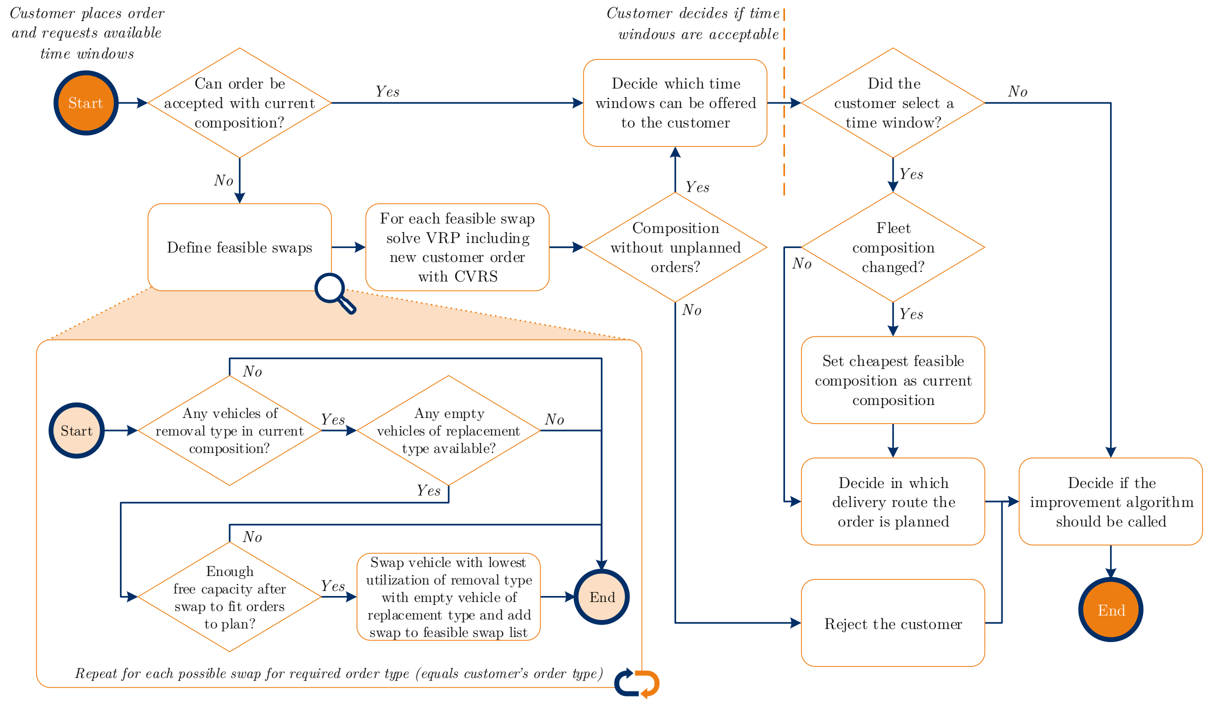
MYS only rejects a customer in case that even after trying to change the fleet composition still no time windows are available to be offered to the customer. As we may try to change the fleet composition at any time during the booking period, we make the choice to swap only one vehicle at a time. When MYS tries to change the fleet composition, this means that we require more capacity for the order type of the customer order that would be rejected if we do not change the fleet composition. Table 4.1 presents all possible swaps for MYS, categorized per use case and required order type.

Table 4.1. *Possible swaps of vehicle types per use case and required order type for MYS*

Required Order Type	Use Case 2	Use Case 4			
A	$2 \to 1$	$2 \to 1$	$2 \to 3$	$1 \to 3$	$3 \to 1$
B	$1 \to 2$	$1 \to 2$	$1 \to 3$	$2 \to 3$	$3 \to 2$

MYS always tries all feasible swaps when changing the fleet composition. A swap is considered to be feasible if the current fleet composition contains any vehicles of the vehicle type to remove and if there are any vehicles without drivers assigned to them of the vehicle type to add. If these conditions are both fulfilled, MYS always selects the vehicle v with the lowest utilization) to remove, to maximize the chance that the orders planned in the removed vehicle can be planned in other vehicles. The utilization is calculated based on the occupied capacity of a vehicle: $\sum_{i \in C_v}\left(q_i/q^v\right)$. Recall from Chapter 3 that q_i represents the order quantity of customer i and q^v the capacity of vehicle v, both in kilograms. C_v is the set of customers whose orders are planned in vehicle v.

After defining all possible new compositions, MYS makes use of CVRS to solve the VRP for each new composition. The orders considered in such a VRP are all the orders that are already planned in a delivery vehicle and the order of the customer that cannot yet be accepted given the current fleet composition. CVRS is configured to maximize the number of planned customer orders, then minimize the delivery costs and finally minimize the total distance of the delivery routes. If multiple new compositions result in being able to offer one or more time windows to the customer (given that all orders that are already confirmed can be re-planned), MYS groups all available time windows for these compositions and offers them to the customer. If the customer then selects one of the offered time windows, MYS continues with the cheapest composition for which that time window is available. Figure 4.3, which is based on Figure 3.3, presents the steps that MYS follows upon a customer arrival.



Customer places order and requests available time windows
Start
Can order be accepted with current composition?
Yes
No
Decide which time windows can be offered to the customer
Customer decides if time windows are acceptable
Did the customer select a time window?
No
Yes
Define feasible swaps
For each feasible swap solve VRP including new customer order with CVRS
Composition without unplanned orders?
Yes
No
Fleet composition changed?
No
Yes
Set cheapest feasible composition as current composition
Decide in which delivery route the order is planned
Decide if the improvement algorithm should be called
End
Reject the customer
Start
Any vehicles of removal type in current composition?
No
Yes
Any empty vehicles of replacement type available?
No
Yes
Enough free capacity after swap to fit orders to plan?
No
Yes
Swap vehicle with lowest utilization of removal type with empty vehicle of replacement type and add swap to feasible swap list
End
Repeat for each possible swap for required order type (equals customer's order type)

Figure 4.3. *Decision steps followed by MYS upon a customer arrival*

We show in Figure 4.3 that when we define feasible swaps, we check whether there is enough free capacity to fit the orders that have been unplanned from the removed vehicle and the new customer order after swapping two vehicles of a certain type. This step consists of checking whether the total free capacity in the new composition (taking into account the orders that were not unplanned) for each order type is enough to fit the orders of that type that still need to be planned. This check is not required, but we implement this check to keep the response time as low as possible, because all calculations regarding the change of the fleet composition are performed while the customer is waiting for available time windows.

4.4. Balanced Strategy (BAS)

In this section we present our smart dynamic strategy, called *Balanced Strategy* (BAS). This strategy builds further on MYS with regard to changing the fleet composition. However, BAS implements several smart decision mechanisms to be able to serve more customers. Just like MYS, the initial fleet composition for BAS is determined as presented in Figure 4.1. Unlike MYS, BAS considers changing the fleet composition for each customer order, even before offering any time windows to a customer. Section 4.4.1 explains how BAS does that. Besides that, BAS considers for each customer order whether the order is unattractive and should be rejected. Section 4.4.2 describes the way in which BAS determines whether a customer is attractive or not. Finally, Section 4.4.3 presents a schematic overview of all decision steps that BAS follows upon a customer arrival.

4.4.1. Changing the Fleet Composition

We first discuss the way in which BAS determines whether to try changing the fleet composition. We consider three questions with regard to possible changes of the fleet composition:

1) Do we want to increase the fleet capacity of the current fleet composition?
2) Do we want to decrease the fleet capacity of the current fleet composition?
3) Do we want to change the ratio of the available fleet capacity for the different order types?

Changing the ratio may be combined with either increasing or decreasing the capacity if desired. When increasing the fleet capacity the ratio is changed anyhow, whether this is required or not, because a small dedicated vehicle is swapped with a larger non-dedicated one. This decreases the percentage of the fleet capacity available for the order type to which the small vehicle was dedicated. Recall that we explained in Section 4.1 that when we change the fleet composition, we always swap a vehicle by re-assigning a driver to a new vehicle.

The same holds for decreasing the fleet capacity, because then a large non-dedicated vehicle is swapped with a small dedicated one. This increases the percentage of the fleet capacity available for the order type to which the new small vehicle is dedicated. Basically, to determine which swap to apply, we first need to know whether we want to increase or decrease the fleet capacity. Second, we need to know for which order type we want to increase the percentage of the fleet capacity that is available for that type. Table 4.2 presents all possible swaps for the different types of changes in the fleet composition, per order type for which we want to increase the share in the fleet capacity.

Table 4.2. *Possible swaps of vehicle types per change type, required order type and use case for BAS*

	Order Type A required		**Order Type B required**	
Change type	*Use Case 2*	*Use Case 4*	*Use Case 2*	*Use Case 4*
Increase capacity	–	2 → 3	–	1 → 3
Decrease capacity	–	3 → 1	–	3 → 2
Change ratio (no change in capacity)	2 → 1	2 → 1	1 → 2	1 → 2

BAS only tries to change the fleet composition if at least one of the three questions is responded with a positive answer. Note that the first two questions will always be responded negatively for Use Case 2. Just like MYS, BAS also checks whether a new composition has enough free capacity to fit all unplanned orders, before sending a VRP to CVRS to check whether a new composition is feasible or not.

To respond the three questions we consider, BAS makes use of forecasts. In our context, we assume that customer orders arrive according to a certain distribution. We have a distribution for the number of customers that arrive during the booking period, as well as a distribution for the spread of the customer arrivals over the booking period. Based on the latter, we can calculate a forecast for the cumulative percentage of customers that should have arrived at any moment during the booking period, according to our expectations. With this forecast, we can for instance estimate the number of customers that should have arrived by a certain point in time, and compare this value to the observed number of customers that have arrived by that time.

BAS makes use of this forecast to recognize for instance when we have a booking period in which the demand for a certain day deviates from the regular demand. Based on this observation BAS takes action with regard to the fleet composition, to improve the performance in terms of customer satisfaction as well as route efficiency. Besides the variables and sets that we defined earlier in Chapter 3, we consider the following new variables and sets when we determine an answer to the three questions BAS considers:

$\mathrm{E}[N]$	The expected number of customers for the entire booking period.
t	The current time expressed as a percentage of the length of the booking period.

F_t The expected cumulative percentage of customers that have arrived by time t, expressed as a decimal value. This percentage is determined based on the expected distribution of customer arrival times. In Chapter 5 we specify this distribution for each scenario that we consider in our computational experiments.

C_t^k The set of all customers $i \in \{1, ..., n\}$ that arrived during the booking period with Order Type $k \in \{A, B\}$ by time t.

α_t The absolute weighted difference between the percentage of the fleet capacity available for Order Type A and the percentage of the orders that arrive with Order Type A at time t.

Besides these variables and sets we define a set of parameters for BAS, that can be configured according to different scenarios. In the remainder of this chapter we explain the meaning of the parameters and how they are used, here we only define them. The parameters that we define can be tuned in such a way that they result in a good performance of the strategy in different cases of the e-retailer case in practice. The process of tuning these parameters for a specific client of ORTEC should ideally be incorporated into the implementation process for OTS. The parameters that we define for BAS are the following:

p_1 Parameter that indicates the expected cumulative percentage of customers that needs to arrive, before BAS considers changing the fleet composition.

p_2 Parameter that indicates the maximum percentage with which the observed number of customers arrived may exceed the expected number. If this parameter is exceeded, BAS tries to change the fleet composition in such a way that the fleet capacity increases.

p_3 Parameter that indicates the maximum absolute percentage that the expected cumulative percentage of customers arrived may be below the average utilization of the current fleet composition. If this threshold is exceeded, BAS tries to increase the fleet capacity by changing the fleet composition.

p_4 Parameter that indicates the maximum number of customers that may be rejected even though it is possible to increase the fleet capacity. When more customers are rejected, BAS changes the fleet composition in such a way that the fleet capacity increases.

p_5 Parameter that indicates the expected cumulative percentage of customers that needs to arrive, before BAS tries decreasing the fleet capacity by changing the fleet composition.

p_6 Parameter that indicates a maximum value for the average utilization of the fleet composition. If this value is not exceeded and the expected cumulative percentage of customers arrived exceeds p_5 BAS tries decreasing the fleet capacity by changing the fleet composition.

p_7 Parameter that indicates a maximum absolute weighted difference between the percentage of the fleet capacity available for Order Type A and the percentage of orders that arrive with Order Type A. When this threshold is exceeded, BAS tries to change the ratio of the available fleet capacity for the order types according to what the ratio should be.

p Parameter that indicates a threshold for the average utilization of the current fleet composition. BAS may only consider rejecting an unattractive customer when this threshold is exceeded.

p Parameter that indicates the minimum percentage point difference between the average utilization of the fleet composition at time t and the expected cumulative percentage of customers that have arrived by time t. Only if the cumulative percentage of customers arrived is deducted from the average utilization and the remainder is exceeds this parameter, BAS may consider rejecting unattractive customers.

p_{10} Parameter between 0 and 1 that indicates the maximum score of unattractiveness that a customer may have. If a customer has a higher score of unattractiveness, BAS rejects the customer.

Note that for all parameters that indicate percentages, we express them as a decimal value.

For all three questions, BAS only considers them after at least p_1 of the expected number of customers have arrived (i.e., $F_t > p_1$). This parameter is introduced to prevent taking bad decisions based on only a few data observations. Also, in the beginning of the booking period typically most customers can be accepted because delivery routes usually do not yet contain many customer orders. The threshold that is used can be configured according to the needs of an e-retailer in practice.

To answer the first question, BAS considers three conditions. If any of them is met, BAS decides to try scaling up the fleet capacity. The first condition regards the number of customers that have arrived by time t. If this amount exceeds the number of customers that are expected to have arrived by time t with a certain percentage, defined by p_2, BAS will take action. This condition indicates that we might be underestimating the total number of orders that will arrive during the booking period and therefore we correct this by increasing the fleet capacity if possible. If Equation (4.3) is satisfied, this means that the first condition is met:

$$\frac{\left|C_t^{\mathrm{A}}\right| + \left|C_t^{\mathrm{B}}\right|}{F_t \times \mathrm{E}[N]} > 1 + p_2 \tag{4.3}$$

Note that the number of customers that have arrived by time t consists of the customers that were accepted and have confirmed their orders, as well as the customers that have been rejected or did not confirm their orders.

The second condition for the first question is that the expected cumulative percentage of customers that have arrived by time t is at least p_3 smaller (in percentage points) than the average utilization at time t of the vehicles in the current fleet composition. This is yet another indication for underestimation, but with this condition we focus on the fleet capacity instead of the number of customers that have arrived. Suppose the expected cumulative percentage of customers that have arrived by time t is smaller than the average utilization at that time. This indicates that, if we expect that the customers that will still arrive on average have the same order quantity as the customers that arrived now, the current fleet capacity will not be sufficient to serve them. In that case it is worthwhile to increase the fleet capacity, as far as that is possible. We quantify this condition with Equation (4.4):

$$F_t + p_3 < \frac{\sum_{j \in \{1,2,3\}} \left(\sum_{v \in V_j} \left(\sum_{i \in C_v} \left(q_i / q^v \right) \right) \right)}{|V_1| + |V_2| + |V_3|} \tag{4.4}$$

The third condition for the first question is that more than p_4 customers were not accepted (either rejected of did not select any of the time windows offered) since the last time the fleet composition was changed, although the fleet capacity can still be increased. This condition is mainly meant to correct for decisions based on bad forecasting. For instance, suppose we are over-forecasting the cumulative percentage of customers that have arrived by time t. Both Equation (4.3) and Equation (4.4) may then not be satisfied, although we actually need to increase the fleet capacity to be able to serve the customer demand still to come. When customers are not accepted, even though we still have the option to increase the capacity of our current fleet composition, we are losing customers unnecessarily because of bad forecasting. Therefore, we correct our bad forecasts by trying to increase the fleet composition if this condition is satisfied.

Given our context, in which on average more customer orders arrive than can be delivered, the answer to our second question will be mostly negative. Also, whenever the answer to our first question is positive, this implies a negative answer to the second question. The answer to our second question is positive if *both* of the following two conditions are met:

- The expected cumulative percentage of customers that have arrived by time t is larger than p_5 (i.e., $F_t > p_5$).
- The average utilization at time t of the vehicles used in the current fleet composition is smaller than p_6 (i.e., right-hand side of Equation (4.4) $< p_6$).

The idea behind these conditions is that we can use p_5 and p_6 to configure BAS in such a way, that BAS for instance can prevent the software from using an expensive additional vehicle for only a few customer orders. It may in such cases be more desirable to reject those few customers by not making an additional vehicle available. If in practical situations it is undesirable to reject customer orders in this way, the parameters can be configured such that BAS never decreases the fleet capacity, by setting $p_6 = 0$.

To answer our third question, we calculate the percentage of vehicles in the fleet that can be used to deliver each order type. For vehicles that can deliver multiple order types, we divide the vehicles equally over each order type that the vehicles can deliver. As we only consider two order types in our context, it is sufficient to calculate the percentage only for Order Type A (denoted by f_{A}) with Equation (4.5):

$$f_{\mathrm{A}} = \frac{|V_1| + 0.5 \times |V_3|}{|V_1| + |V_2| + |V_3|} \times 100\% \tag{4.5}$$

Now that we have the percentage of the fleet that is available for Order Type A, we can compare this percentage with the percentage of customers with Order Type A that we expect according to historical data (denoted by h_{A}, see Equation (4.6)), as well as the percentage of customers that arrived with Order Type A during the booking period by the current time t (denoted by o_{A}, see Equation (4.7)). Recall from Section 4.2 that C_h^{A} is the set of all historical customers (i) with Order Type A, and C_h^{B} the same set for Order Type B.

$$h_{\mathrm{A}} = \frac{|C_h^{\mathrm{A}}|}{|C_h^{\mathrm{A}}| + |C_h^{\mathrm{B}}|} \tag{4.6}$$

$$o_{\mathrm{A}} = \frac{|C_t^{\mathrm{A}}|}{|C_t^{\mathrm{A}}| + |C_t^{\mathrm{B}}|} \tag{4.7}$$

At the beginning of the booking period, when only few customer orders have come in, we typically rely more on our expectations according to historical data. The reason for this is that we do not have much information about the realization for this booking period yet. However, as more customer orders come in while the time elapses, we obtain more and more observations from reality. Therefore, the more customers arrive, the more we want to rely on our observations from reality.

To clarify the application of this idea with regard to the ratio of capacity that is reserved for each order type, we consider a numerical example. We consider two moments in the booking period. Suppose that we expect 100 customers to arrive during a certain booking period. Suppose further that at the first moment 7 customers have arrived with Order Type A, and 3 with Order Type B. Furthermore, between the first and the second moment 3 customers have arrived with Order Type A, and 7 customers with Order Type B. The moments are characterized by the following states (the state at the second moment depends on the decision taken at the first moment):

1) $F_t = 0.10,\ o_{\mathrm{A}} = 0.70,\ f_{\mathrm{A}} = h_{\mathrm{A}} = 0.50$
2) **(a)** $F_t = 0.20,\ o_{\mathrm{A}} = 0.50,\ f_{\mathrm{A}} = 0.60,\ h_{\mathrm{A}} = 0.50$ **(b)** $F_t = 0.20,\ o_{\mathrm{A}} = f_{\mathrm{A}} = h_{\mathrm{A}} = 0.50$

For the first moment, we have several options with regard to the way in which we take a decision. A first option is to focus merely on our observations from practice. Those tell us that the current percentage of the fleet capacity reserved for Order Type A is 20% (absolute difference) below the percentage of orders of Order Type A that we observe in practice. Based on only this information we would change the fleet composition and for instance end up in the state of moment 2(a). However, we see that at the second moment the percentage of orders of Order Type A that we observe in practice dropped again. This percentage now equals the percentage according to our expectations based on historical data. We would have been better off if we would not have changed the fleet composition in this case, thus ending up in the state of moment 2(b).

A second option is that we take into account that only a small percentage of the customers has arrived at the first moment. Therefore, we may believe that our forecasts based on historical data (that are reflected in the fleet composition at the first moment), are still accurate and that relatively more customers with Order Type B will arrive in the remainder of the booking period. This would have led to not changing the fleet composition, and ending up in the state of moment 2(b). In this numerical example the second option would have been the best one, and we believe that this holds for many cases. Some initial experiments do confirm this hypothesis as well.

To reflect this principle, we define the absolute weighted difference (α_t) between the percentage of the fleet capacity available for Order Type A and the percentage of the orders that arrive with Order Type A, at time t. The term *weighted* refers to the fact that when only a small percentage of the customers has arrived according to our expectations, we consider the historical data to be more reliable to determine the percentage of orders with Order Type A when calculating α_t. However, the further the booking period elapses and the higher the percentage of customers that has arrived according to our expectations, the more confidence we have in using our observations in reality to determine the percentage of orders with Order Type A when calculating α_t. This results in the calculation of α_t as shown in Equation (4.8):

$$\alpha_t = \left|(1 - F_t) \times \left(h_{\mathrm{A}} - f_{\mathrm{A}}\right) + F_t \times \left(o_{\mathrm{A}} - f_{\mathrm{A}}\right)\right| \tag{4.8}$$

If this absolute weighted difference is larger than a certain threshold p_7 (i.e., $\alpha_t > p_7$), we consider the deviation of the ratio of the order types large enough to change the ratio in the fleet composition and give a positive answer to our third question. If the observed percentage of Order Type A is larger than the percentage of the fleet available for Order Type A ($o_{\mathrm{A}} > f_{\mathrm{A}}$), we set Order Type A as the order type for which we want to increase the available percentage of the fleet capacity. Otherwise, we set Order Type B as the required order type.

In case $\alpha \leq p_7$, we give a negative response to our third question. However, when we either want to increase or decrease the fleet capacity (only Use Case 4), we still need to provide an order type for which we want to increase the available percentage of the fleet capacity. We then select the order type with the largest total order quantity of the customer orders of that type that have arrived so far.

Recall the three questions with regard to possible changes of the fleet composition that we defined earlier. We now defined for all three the way in which BAS determines the response. If we finally obtain a positive answer to any of the three questions, BAS determines a new fleet composition based on the swaps that are defined in Table 4.2. BAS then proceeds in the same way as MYS to check whether the new compositions are feasible. However, there is one difference: BAS does not add the order of the customer that triggered the decision sequence to the orders that need to be planned when sending a VRP to CVRS. Instead, BAS first tries to change the fleet composition if desired, and afterwards determines whether to accept or reject the customer order, as described in Section 4.4.2. Note that BAS does use the order details of the customer that triggered the decision sequence to respond the three questions regarding a change of the fleet composition.

4.4.2. Customer Rejection

After changing the fleet composition (or not, in case that turned out not to be desirable), BAS returns to the original customer order that triggered the decision sequence to consider whether this customer order can be accepted. In our context, on average the customer demand is more than the delivery capacity an e-retailer has available. This means that on average, an e-retailer will always reject a number of customers. BAS anticipates on this by considering whether or not a customer is attractive enough to serve. If not, BAS rejects this customer, aiming to accept multiple future customers (or one more profitable future customer) in its place. BAS takes into account a few conditions that must be met before actually considering to reject a customer:

- The fleet capacity cannot be increased anymore, because none of the drivers can be re-assigned to a vehicle that has a larger capacity than the vehicle a driver is currently assigned to. The fact that the current fleet composition is not a composition with maximized capacity, reflects the expectation that at this stage we do not expect that more customer demand will come in than we can serve. Therefore, it is no use to reject any customer at this stage. We only want to reject customers if we expect that we cannot serve all customer order that will come in.

- The average utilization of the current fleet composition must be larger than a certain threshold p . This parameter can be used to prevent that BAS starts rejecting customers in a very early stage of the booking period, when few customers have arrived yet and actually there is still a lot of capacity left in the delivery vehicles. Especially when we have few observations it is difficult to make reliable estimations with regard to how demand will develop in the remainder of the booking period, so we want to be careful to be too early with rejecting unattractive customers.
- The expected cumulative percentage of customers that have arrived by the customer's arrival time t is at least p smaller (in percentage points) than the average utilization, indicating that we expect at least $p \times 100\%$ more customers to arrive than the average number of customers that can be served, given the total capacity of the current fleet composition. This parameter is yet another parameter that can be used to prevent BAS from rejecting customers without having a strong enough indication that during the current booking period indeed more customer demand will arrive than can be served. By configuring this parameter we can play with how large we allow the risk of unnecessarily rejecting customers to be.

If all these conditions are met, BAS needs to determine whether to accept or reject the customer based on the customer's attractivity. There are numerous ways to quantify attractivity. We can for instance think of the additional distance to include a customer's order in the current routing plan, the order quantity of the customer's order or the profitability of the customer's order. We may even combine several factors into one weighted score to determine the attractivity of a customer. The best way in which we can determine the attractivity may differ for each practical application of the e-retailer case.

To have a common way to determine whether a customer is unattractive, we introduce an unattractiveness score for each customer. This is a score between 0 and 1, where 0 means that a customer is not unattractive, and 1 means that a customer is very unattractive. We define p_{10} to be the threshold of unattractiveness. If a customer's unattractiveness score exceeds p_{10}, BAS rejects the customer. As mentioned, the way in which this score is determined may vary for each practical application of the e-retailer case.

Having introduced the method that BAS employs to determine whether a customer should be rejected or not, we need to place some side remarks. We emphasized that it is important to take care that customers are not rejected too soon. In the end the forecasting measures that we use remain estimations that may or may not be correct. Therefore, we should be careful to prevent rejecting customers due to bad forecasting, while in the end it turns out that those customers could actually have been served.

On the other hand, if we are too careful with rejecting customers, this may lead to accepting unattractive customers which may not be beneficial at all. It is therefore important to tune the parameters that we use in such a way that an equilibrium is found between the two extremes of either rejecting too many customers or rejecting to few customers. Also, the parameter tuning should of course be in line with the performance metrics that are chosen as optimization criteria in a certain context. We could however dedicate a whole thesis on this topic of tuning the parameters in an appropriate way for different applications of the e-retailer case. This is out of scope for our research, therefore we only apply a basic tuning method for the parameters that BAS uses for our experiments. We provide more details on this process in Chapter 5.

4.4.3. Schematic Overview

In this section we present a schematic overview of the decision steps followed by BAS. The overview is displayed in Figure 4.4, which is an extended version of Figure 3.3. Note that BAS is actually a very simple strategy. We could have chosen to not only use simple expected values for our forecasts. Instead, we could for instance make use of more complex statistical techniques as sampling, possibly combined with dynamic programming techniques. However, we need to keep in mind that the time that we have to perform all required calculations is limited, because all calculations are performed while the customer is waiting for a response. Also, for the more complex techniques we may need a lot of information which not necessarily is available to an e-retailer (for instance the customers' time window preferences). This forces us then to make more assumptions, which would likely result in a worse performance of the strategy. Therefore, we decided to keep our strategies as simple as possible, while still trying to achieve a good performance.

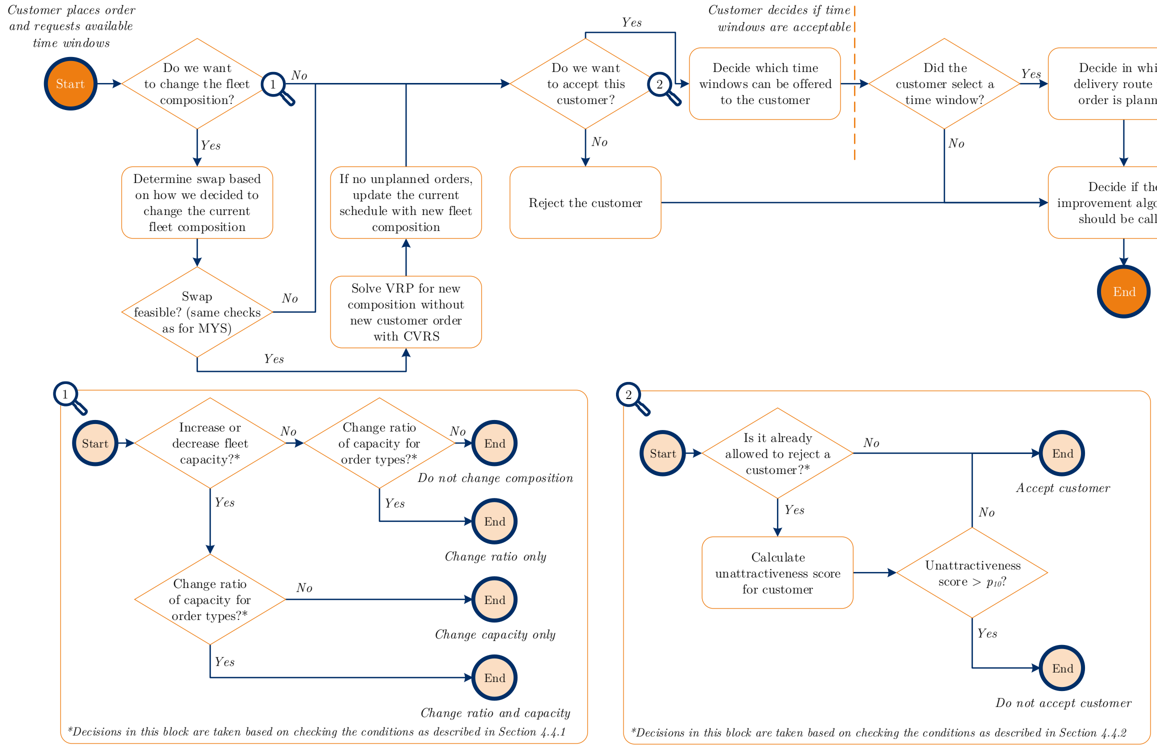

Customer places order and requests available time windows
Start
Do we want to change the fleet composition?
1
No
Yes
Determine swap based on how we decided to change the current fleet composition
Swap feasible? (same checks as for MYS)
No
Yes
Solve VRP for new composition without new customer order with CVRS
If no unplanned orders, update the current schedule with new fleet composition
Yes
Do we want to accept this customer?
2
No
Reject the customer
Customer decides if time windows are acceptable
Decide which time windows can be offered to the customer
Did the customer select a time window?
Yes
No
Decide in which delivery route the order is planned
Decide if the improvement algorithm should be called
End
1
Start
Increase or decrease fleet capacity?*
No
Change ratio of capacity for order types?*
No
End
Do not change composition
Yes
End
Change ratio only
Yes
Change ratio of capacity for order types?*
No
End
Change capacity only
Yes
End
Change ratio and capacity
*Decisions in this block are taken based on checking the conditions as described in Section 4.4.1
2
Start
Is it already allowed to reject a customer?*
No
End
Accept customer
Yes
Calculate unattractiveness score for customer
No
Unattractiveness score > p_{10}?
Yes
End
Do not accept customer
*Decisions in this block are taken based on checking the conditions as described in Section 4.4.2

Figure 4.4. *Schematic overview of decision steps followed by BAS upon a customer arrival*

4.5. Conclusion

In this chapter we defined our solution approach to solve the problem that we consider in the context of the e-retailer case. We described all decisions that our solution strategies should take in Section 4.1. We designed three strategies, that all deal with these decisions in different ways. In Section 4.2 we designed a static strategy (OBS) that can be implemented directly in ORTEC's software solutions. In Section 4.3 we designed a myopic dynamic strategy (MYS) that can change the fleet composition while customer orders arrive during the booking period. Finally, in Section 4.4 we designed a smart dynamic strategy (BAS) that implements several smart decision mechanisms to improve MYS. The insights obtained in this chapter enable us to answer our third set of research questions:

> *3) Which solution strategies do we design to deal with an unfixed fleet composition during the ordering process?*
>
> *a. What are the problems for which our solution strategies should come up with a decision?*
>
> *b. How can we deal with these problems in the solution strategies that we design?*

To find a solution to the problem we consider, our strategies must take decisions for two problems:

1) We need to determine an initial fleet composition before the start of the booking period.
2) For each customer order that arrives we must decide whether to accept or reject this order. Note that accepting an order does not necessarily mean that the customer confirms the order by selecting a time window.

OBS approaches the first decision in two different ways. The first way consists of calculating the historical percentage of the total order quantity for each order type that we consider. Then, according to these percentages m_k drivers are assigned to each Order Type k. The initial fleet composition is then determined by, for each order type, adding the m_k cheapest vehicles (in terms of setup costs) that can deliver Order Type k to the fleet composition. This first method is used to benchmark OBS against MYS and BAS.

As we expect more customer demand than can be served with maximum capacity (given the number of drivers available), in practice we would not add the cheapest and smallest vehicles to the initial fleet composition. Therefore, the second way in which OBS approaches the first decision consists of first adding as many vehicles of Vehicle Type 3 (large and non-dedicated) to the fleet composition. If any drivers are then not yet assigned to a vehicle, the first method is followed to assign those remaining drivers to a vehicle.

For the second decision, OBS accepts a customer if at least one time window is available to be offered to a customer. Otherwise, OBS rejects the customer. OBS does not try to change the fleet composition nor considers any other conditions to reject a customer, besides the one mentioned above.

MYS approaches the first decision in the same way as the first method for OBS. The second decision is approached in a different way. Just as OBS, MYS accepts a customer order if at least one time window is available to be offered to the customer. When this is not the case, MYS does not immediately reject the customer. Instead, MYS tries at any cost to change the fleet composition in such a way that at least one time window is available to be offered to the customer. If this is the case, MYS accepts the customer. If after trying to change the fleet composition still no time window is available to be offered to the customer, MYS rejects the customer order.

BAS approaches the first decision in the same way as MYS. For the second decision BAS makes use of a different approach. For each customer order BAS considers changing the fleet composition even before offering any time windows to the customer by answering three questions:

1) Do we want to increase the fleet capacity of the current fleet composition?
2) Do we want to decrease the fleet capacity of the current fleet composition?
3) Do we want to change the ratio of the available fleet capacity for the different order types?

If any of these questions can be responded with a positive answer, BAS tries to change the fleet composition, in a way that reflects the answers to the questions. To answer these questions, BAS calculates different value functions. With the resulting values BAS verifies whether several conditions are met. These conditions are used to check whether our fleet composition is suitable to serve the customers that are placing orders during the booking period. If that turns out not to be the case, BAS tries to change the fleet composition accordingly.

Besides that, BAS checks several conditions before a customer is accepted. Besides the check if time windows are available to be offered, BAS also checks whether the customer is attractive enough to accept based on certain conditions. BAS only considers rejecting an unattractive customer when several conditions are met. By rejecting unattractive customers BAS aims to serve more customers in the end, as we focus especially on cases where we know that on average we are not able to serve all customers that arrive during the booking period.

All conditions that BAS considers when taking its decisions can be configured by tuning the parameters that BAS uses. In this way BAS can be tuned according to different practical applications of the e-retailer case.

5. Simulation Approach

In this section we present our simulation approach for the ordering process for the e-retailer case. We first define the structure for our model in Section 5.1. After that, in Section 5.2 we describe the simulation tool that we develop to run our simulations. In Section 5.3 we present the scenarios of the e-retailer case for which we carry out computational experiments to put the hypotheses as defined in Chapter 3 to a test. Finally, in Section 5.4 we present an overview of the experiments that we carry out.

5.1. Simulation Model Structure

Before we develop a simulation model, it is important to know how the simulation model must be structured. In this section we design the structure of our simulation model in terms of the inputs and the outputs that are required (Section 5.1.1), as well as the way our simulation model processes the inputs to obtain the outputs (Section 5.1.2).

We chose to model the ordering process of the e-retailer case with a discrete-event simulation model, as described by (Law, 2015). As mentioned in Chapter 3, this saves us a lot of running time compared to real-time simulation. The discrete events that we consider are the sequential arrivals of the customers that place a request for available time windows during the booking period. In between those arrivals, the state of our system does not change which qualifies our context to be modeled with a discrete event simulation model. The current time in our model is represented by the arrival times of the customers, expressed as a percentage of the length of the booking period. We deal with a terminating simulation in which we start with an empty system (no customer orders arrived), and the cut-off time marks the end of a simulation run. Therefore, we do not need a warm-up period for our independent simulation runs.

5.1.1. Inputs and Outputs

In this section we define all required inputs for our simulation model, based on the formal problem definition from Chapter 3. Also, we define all the required outputs for our simulation that we need for the analysis of the results.

We denominate the set of all required inputs as a scenario. The main elements of a scenario are constructed with the four entities we defined in Chapter 3 as building blocks. We distinguish the following inputs for our simulation model:

- *Id* – A unique name to identify a scenario.
- *Use case* – The use case (as defined in Chapter 3) for which we consider the scenario.
- *Strategy* – This parameter indicates which strategy our simulation model should apply for the input scenario.
- *Customers* – A set that contains all historical customers and their attributes that we consider for a scenario. Based on the average number of customers and a Poisson distribution (explained further in Section 5.3), for each replication we randomly draw a number of customers that try to place an order (during the booking period for that replication) from this set.
- *Average number of customers* – A parameter that indicates how many customers on average want to place an order during the booking period.
- *Drivers* – A number of drivers that an e-retailer employs with their attributes.
- *Vehicles* – The delivery vehicles that an e-retailer owns. The vehicles are categorized based on their vehicle types.
- *Number of vehicles per type* – These parameters characterize the delivery fleet of an e-retailer in terms of how many vehicles an e-retailer owns of each vehicle type.
- *Depot* – A central depot where an e-retailer keeps its inventory with its attributes.
- *Number of customers until optimization* - This parameter indicates the number of customers after which we call an improvement algorithm from CVRS for intermediate optimization of the delivery routes formed so far.

- *Number of replications* – This parameter indicates how many replications of one booking period we simulate for the scenario. For an explanation of the concept of a replication (equivalent to a simulation run) we refer to Law (2015).
- *Distance matrix* – The distance matrix contains the distances in kilometers between all locations in the scenario, being the locations of all customers and the location of the central depot.
- *Driving time matrix* – Same as the distance matrix, but then for the driving time in seconds.

The outputs for our simulation model consist of two main elements. The first element is a list of customers that arrived during the booking period to place an order. The second element is a schedule that is being built up in the simulation during the booking period. Below we explain these two and other outputs of our simulation model:

- *List of customers* – The list of customers contains for each customer the response time in seconds to a request for available time windows, as well as the number of time windows offered to the customer. Besides that, the list contains for each confirmed customer order the selected time window and in which vehicle the order is planned.
- *Schedule* – The schedule consists of the vehicles used to form the final fleet composition and the sequence of customer orders that are planned in these vehicles. Based on the list of customers and the schedule we calculate the values for all the KPIs that we defined in Chapter 3.
- *Running time* – The time in seconds that it takes for our simulation model to run a replication of one booking period.
- *Schedule update time* – The time in seconds that it on average takes to update the current schedule in OTS. In Section 5.1.2 and Section 5.2 we explain when and how the communication with OTS and CVRS takes place.
- *Quick optimize time* – The time in seconds that it on average takes to perform an intermediate optimization of the schedule in CVRS.
- *Final optimize time* – The time in seconds that it takes to perform the final optimization of the schedule after the end of the booking period.
- *Change fleet time* – The time in seconds that we spend on average when we try to change the fleet composition. By definition, for scenarios in which the strategy is OBS this measure equals 0.

Finally, to keep track of the evolution of our KPIs over time and to facilitate the calculations for our strategies, we define the system state. This system state is updated every time a customer order arrives and consists of the following elements:

- *Customer* – All information about the customer order that triggered the update of the system state and the system state.
- *Total number of customers arrived* – The total number that arrived so far to place an order.
- *Percentage of customers arrived per order type* – For each order type the customers that have arrived to place an order of the corresponding type divided by the total number of customers arrived.
- *Total order quantity* – The total quantity of all orders that have been confirmed so far.
- *Percentage of total order quantity per order type* – The total order quantity of all orders of an order type divided by the total order quantity.
- *Current fleet composition* – The fleet composition that is currently used to serve customers.
- *Number of drivers used* – The total number of vehicles in which at least one customer order is planned.
- *Total fleet capacity* – The total capacity of all vehicles in the current fleet composition together.
- *Fleet capacity per order type* – The total fleet capacity available for an order type (based on Equation (4.5) for Order Type A, analogously for Order Type B).
- *Total delivery costs, total distance, total driving time, delivery costs per customer, percentage of customers served* – KPIs that are calculated based on the schedule built up so far.

5.1.2. Simulation Steps

In this section we describe in pseudo-code the steps that we follow in our simulation to convert the inputs to outputs. Algorithm 5.1 carries out the main simulation process, from the initialization of the simulation to the export of the outputs. This algorithm is carried out for every replication and simulates the whole ordering process of an e-retailer, from the before the start when the initial fleet composition is determined, until after the cut-off time when the delivery routes are optimized.

Algorithm 5.1. *Main simulation process*

```
read all input data of scenario into simulation model            ▷ Inputs as defined in Section 5.1.1
initialize all simulation parameters and schedule                ▷ Algorithm 5.2
for $c = 1$ to $n$ do                                            ▷ Let all customers arrive sequentially
    if strategy = OBS then                                       ▷ Check the strategy that should be applied
        send JSON request for available time windows to OTS      ▷ OTS runs cheapest insertion algorithm
        set $AT$ = list of available time windows                ▷ Retrieved from OTS response
        set $PL$ = sorted preference list of customer $c$
        if $AT$.count $> 0$ then
            if $PL$ contains any time windows from $AT$ then     ▷ If not, customer did not select time
                customer $c$ selects highest ranked time window from $PL$     window, order is not confirmed
                insert customer $c$ in cheapest way into schedule    ▷ Determined by OTS (cheapest insertion)
            end if
        else reject customer $c$
        end if
    else if strategy = MYS then                                  ▷ Customer is accepted when insertion is feasible
        execute specific simulation steps for MYS                ▷ Algorithm 5.3
    else if strategy = BAS then
        execute specific simulation steps for BAS                ▷ Algorithm 5.4
    end if
    set $x$ = number of customers between intermediate optimization calls
    if $c$ mod $x = 0$ then                                      ▷ After every $x$ customers, call
        send JSON request to CVRS for intermediate optimization      improvement algorithm
        convert JSON response CVRS into current schedule
    end if
    send JSON request to OTS with current schedule               ▷ Keep OTS up-to-date
next $c$
send JSON request to CVRS to perform final optimization of delivery routes
convert JSON response CVRS into final schedule
send JSON request to OTS to reset the central depot
export all output and KPIs                                       ▷ As defined in Section 5.1.1
```

Algorithm 5.1 makes use of different sub-algorithms, which we present below. Algorithm 5.2 shows the steps that are followed before the start of every replication, to ensure that the simulation can run properly.

Algorithm 5.2. *Initialization of simulation parameters*

```
send JSON request to initialize the depot in OTS
if strategy = BAS or MYS or OBS then
    initialize delivery fleet as cheap as possible               ▷ As presented in Figure 4.1
else if strategy = OBS (Large Initial Fleet) then
    initialize delivery fleet with maximum capacity              ▷ As presented in Figure 4.2
end if
send JSON request with initial empty schedule to OTS
randomly select the customers that arrive in this replication    ▷ Further explanation in Section 5.3
```

Algorithm 5.3 carries out all the steps upon a customer arrival as prescribed by MYS (Figure 4.3).

Algorithm 5.3. *Simulation steps upon a customer arrival for MYS*

```
send JSON request for available time windows to OTS        ▷ OTS runs cheapest insertion algorithm
set $AT$ = list of available time windows                    ▷ Retrieved from OTS response
set $PL$ = sorted preference list of customer $c$
if $AT$.count $> 0$ then
    if $PL$ contains any time windows from $AT$ then          ▷ If not, customer did not select time
        customer $c$ selects first time window in $PL$            window, order is not confirmed
        insert customer $c$ in cheapest way into schedule    ▷ Determined by OTS (cheapest insertion)
    end if
else
    set $FS$ = list of feasible swaps                         ▷ Swap defines a new fleet composition
    set $FC$ = list of feasible compositions                  ▷ Fleet composition is feasible when
    send JSON request to initialize a copy of the depot in OTS     we have no unplanned orders
    add feasible swaps (with customer's order type as required) to $FS$   ▷ See Figure 4.3 for steps to follow
    for each $swap$ in $FS$ do
        send JSON request to solve VRP for $swap$ with CVRS
        if no unplanned orders in response then                ▷ Composition is feasible, all
            add $swap$ composition to $FC$                         confirmed orders and order of
            convert CVRS response into temporary schedule            customer c could be planned
            send JSON request with temporary schedule to OTS
            send JSON request for available time windows to OTS   ▷ OTS runs cheapest insertion algorithm
            if any, add available time windows to $AT$            ▷ Retrieved from OTS response
            send JSON request to OTS to reset copy of the central depot
        end if
    next $swap$
    filter out any duplicate time windows in $AT$
    if $AT$.count $> 0$ then
        if $PL$ contains any time windows from $AT$ then      ▷ If not, customer did not select time
            customer $c$ selects highest ranked time window from $PL$   window, order is not confirmed
            set cheapest corresponding temporary schedule as current schedule and update schedule in OTS
            insert customer $c$ in cheapest way into schedule   ▷ Determined by OTS (cheapest insertion)
        end if
    else reject customer $c$
    end if
end if
```

Algorithm 5.4 carries out the steps of BAS upon a customer arrival, as discussed in Chapter 4 and presented in Figure 4.4. We use the notation of the parameters and variables as introduced in Chapter 4.

Algorithm 5.4. *Simulation steps upon a customer arrival for BAS*

```
set $IncreaseCapacity$ = false                    ▷ Initialize parameters to check if fleet composition
set $DecreaseCapacity$ = false                       should be changed, see Chapter 4 for details
set $ChangeRatio$ = false
set $ChangeComposition$ = false
set $ReqOrderType = k$
set $RC$ = list with customers that could not be served
if $F_t > p_1$ then                               ▷ First check the expected percentage of customers arrived
    if $\left(|C_t^A| + |C_t^B|\right)/(F_t \times \mathrm{E}[N]) > 1 + p_2$ or   ▷ Then check whether to increase capacity, making
     $F_t + p_3 <$ current average utilization or         use of Equation (4.3) and Equation (4.4)
     $RC$.count $> p_4$ then
        $IncreaseCapacity$ = true                 ▷ More fleet capacity is required to serve future customers
    else if $F_t > p_5$ and current average utilization $< p_6$ then   ▷ Check whether to decrease capacity
        $DecreaseCapacity$ = true      ▷ We expect to need less fleet capacity for serving future customers
    end if
    calculate $f_A$, $h_A$, $o_A$ and $\alpha_t$    ▷ Now check whether to change the ratio of the available fleet capacity,
    if $\alpha_t > p_7$ then    with Equation (4.5), Equation (4.6), Equation (4.7), Equation (4.8)
        $ChangeRatio$ = true
        if $o_A > f_A$ then    ▷ Determine order type for which we require more capacity
            $ReqOrderType$ = A
        else $ReqOrderType$ = B
        end if
    else if $IncreaseCapacity$ or $DecreaseCapacity$ then
        $ReqOrderType$ = order type with largest total order quantity    ▷ Based on customers served so far
    end if
    if $IncreaseCapacity$ or $DecreaseCapacity$ or $ChangeRatio$ then
        $ChangeComposition$ = true
        $RC$.clear    ▷ Reset the list with rejected customers
    end if
end if
if $ChangeComposition$ then    ▷ Now try changing the fleet composition
    define new composition based on $IncreaseCapacity$, $DecreaseCapacity$, $ChangeRatio$    ▷ Use Table 4.2
    if enough capacity available for unplanned orders then    ▷ Prevent high response time when we know
        send JSON request to CVRS to solve VRP for new composition    that new composition is not feasible
        if no unplanned orders in response then
            convert CVRS response into current schedule    ▷ Accept changed fleet composition
            send JSON request to OTS with current schedule    ▷ Keep OTS up-to-date
        end if
    end if
end if
set $RejectCustomer$ = false    ▷ Finally, consider if customer should be rejected
set $u$ = unattractiveness score for customer $c$    ▷ Further explanation of calculation in Section 5.3
if fleet capacity cannot be increased anymore and    ▷ Perform checks as described in Chapter 4
 current average utilization > $p$ and
 $F_t + p$ < current average utilization and $u > p_{10}$ then
    $RejectCustomer$ = true
end if
if not $RejectCustomer$ then    ▷ If customer is accepted, offer available time windows
    send JSON request for available time windows to OTS    ▷ OTS runs cheapest insertion algorithm
    set $AT$ = list of available time windows    ▷ Retrieved from OTS response
    set $PL$ = sorted preference list of customer $c$
    if $AT$.count > 0 then
        if $PL$ contains any time windows from $AT$ then    ▷ If not, customer did not select time
            customer $c$ selects highest ranked time window from $PL$    window, order is not confirmed
            insert customer $c$ in cheapest way into schedule    ▷ Determined by OTS (cheapest insertion)
        end if
    else reject customer $c$
    end if
else reject customer $c$
end if
```

All these algorithms together make sure that all inputs are converted to outputs, which we use to investigate whether the hypotheses as defined in Chapter 3 can be confirmed for our input scenarios.

5.2. Simulation Tool

In this section we describe the simulation tool we developed in C# to run our simulation experiments. C# is an object-oriented programming language that fits all the needs for running the algorithms we presented in Section 5.1. We present the way in which our simulation tool communicates with ORTEC's software solutions in Section 5.2.1. Subsequently, we briefly describe the different application modes that we developed in the tool in Section 5.2.2.

5.2.1. Communication with OTS and CVRS

As we already mentioned in the algorithms in Section 5.1, the communication with the cloud interfaces ORTEC uses for OTS and CVRS goes by means of JSON requests. JSON is frequently used in cloud applications to exchange data between two interfaces. It is a lightweight format and therefore suitable for applications where response time is an issue. ORTEC has specified a standard format for communication with its cloud services, that contain all the entities that are required to be able to respond the cloud request. We implemented these formats in our simulation tool to automate the communication with OTS and CVRS. So, our tool converts all entities (customer, driver, vehicle, depot) to entities that are defined in OTS and CVRS. After obtaining a response from OTS or CVRS the tool converts this response to the corresponding entities in our simulation tool.

5.2.2. Application Modes

Figure 5.1 shows the start screen of the application we developed to run our simulations and introduces the four application modes: a scenario generator mode, a demo mode, a simulation mode and a solution visualizer mode. These application modes can be used for different purposes, as we explain for each application mode in the remainder of this section.

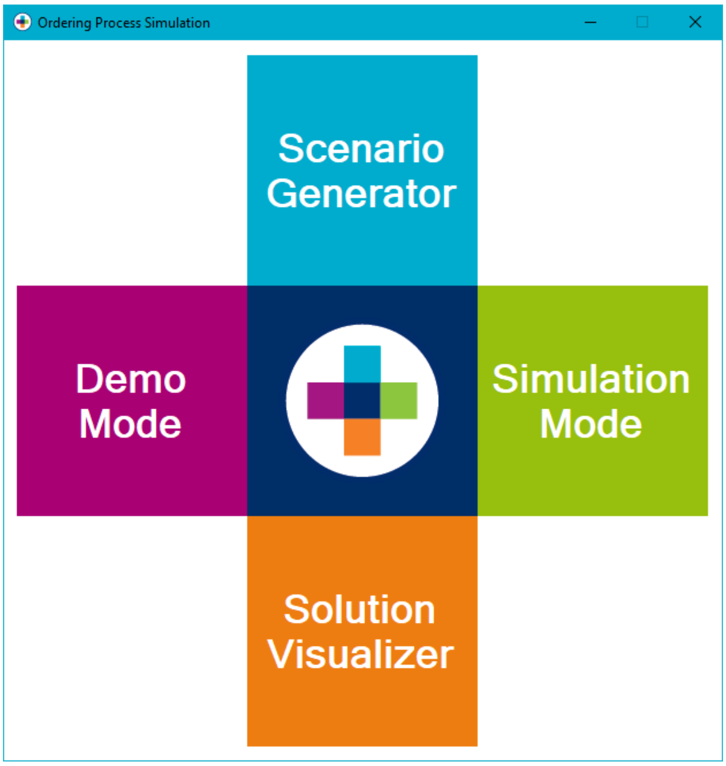
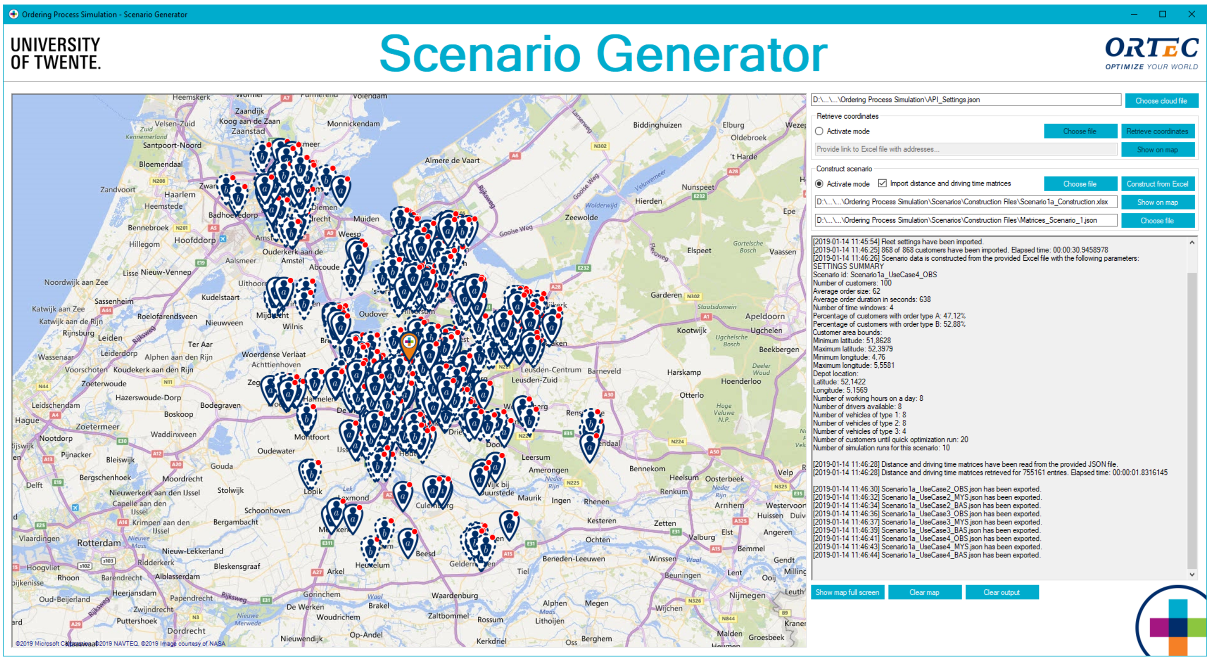


Figure 5.1. *Screenshot of the start screen of the simulation tool*

Figure 5.2 presents a visualization of the scenario generator mode of our simulation tool. This application mode can be used to construct input files for a scenario. We distinguish between two separate functions for this application mode: one to retrieve the coordinates for a set of given addresses and one to consolidate all inputs for a scenario from an Excel file into a JSON file that can be read by our simulation tool. To retrieve the coordinates linked to an address, the tool makes use of ORTEC's Maps Service. Whenever an address is not recognized, our tool gives the user the option to manually provide the coordinates. When constructing an input file for a scenario, the application also uses ORTEC's Maps Service, to retrieve the distance and driving time matrices. In case we already have the distance and driving time matrices available as a JSON file, the tool offers the option to import this JSON file. This can save much time, because for some scenarios we may have millions of entries for the distance and driving time matrices.



D:\...\...\Ordering Process Simulation\API_Settings.json

Retrieve coordinates

○ Activate mode

Provide link to Excel file with addresses...

Construct scenario

● Activate mode ☑ Import distance and driving time matrices

D:\...\...\Ordering Process Simulation\Scenarios\Construction Files\Scenario1a_Construction.xlsx

D:\...\...\Ordering Process Simulation\Scenarios\Construction Files\Matrices_Scenario_1.json

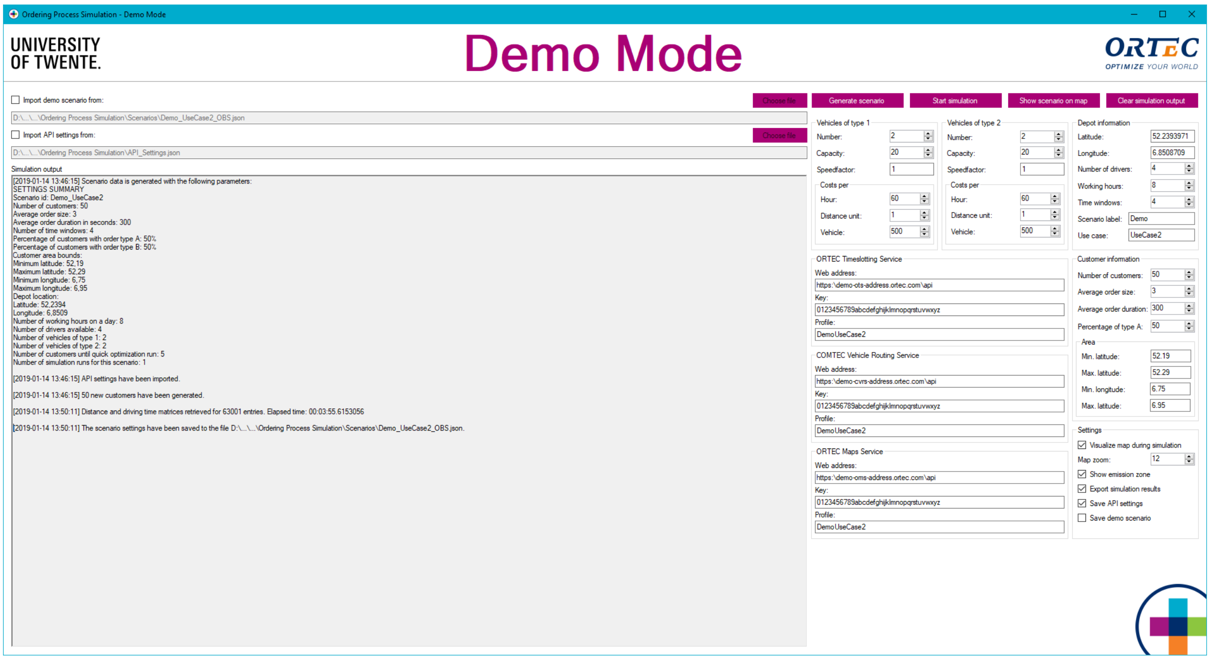
```
[2019-01-14 11:45:54] Fleet settings have been imported.
[2019-01-14 11:46:25] 868 of 868 customers have been imported. Elapsed time: 00:00:30.9458978
[2019-01-14 11:46:26] Scenario data is constructed from the provided Excel file with the following parameters:
SETTINGS SUMMARY
Scenario id: Scenario1a_UseCase4_OBS
Number of customers: 100
Average order size: 62
Average order duration in seconds: 638
Number of time windows: 4
Percentage of customers with order type A: 47,12%
Percentage of customers with order type B: 52,88%
Customer area bounds:
Minimum latitude: 51,8628
Maximum latitude: 52,3979
Minimum longitude: 4,76
Maximum longitude: 5,5581
Depot location:
Latitude: 52,1422
Longitude: 5,1569
Number of working hours on a day: 8
Number of drivers available: 8
Number of vehicles of type 1: 8
Number of vehicles of type 2: 8
Number of vehicles of type 3: 4
Number of customers until quick optimization run: 20
Number of simulation runs for this scenario: 10

[2019-01-14 11:46:28] Distance and driving time matrices have been read from the provided JSON file.
[2019-01-14 11:46:28] Distance and driving time matrices retrieved for 755161 entries. Elapsed time: 00:00:01.8316145

[2019-01-14 11:46:30] Scenario1a_UseCase2_OBS.json has been exported.
[2019-01-14 11:46:32] Scenario1a_UseCase2_MYS.json has been exported.
[2019-01-14 11:46:34] Scenario1a_UseCase2_BAS.json has been exported.
[2019-01-14 11:46:36] Scenario1a_UseCase3_OBS.json has been exported.
[2019-01-14 11:46:37] Scenario1a_UseCase3_MYS.json has been exported.
[2019-01-14 11:46:39] Scenario1a_UseCase3_BAS.json has been exported.
[2019-01-14 11:46:41] Scenario1a_UseCase4_OBS.json has been exported.
[2019-01-14 11:46:43] Scenario1a_UseCase4_MYS.json has been exported.
[2019-01-14 11:46:44] Scenario1a_UseCase4_BAS.json has been exported.
```

Figure 5.2. *Screenshot of the scenario generator mode*
Map rights: ©2019 Microsoft Corporation, ©2019 NAVTEQ, ©2019 Image courtesy of NASA, plugin by GMap.NET

Figure 5.3 shows a screenshot of the demo mode of our simulation tool. This mode, as the name indicates, is meant for demonstration purposes of the tool. The user can generate a basic random scenario for Use Case 2. Several parameters can be modified, which serve as input for the random distributions that the application uses to generate the input for the demo scenario. Another possibility is to import an earlier generated scenario. After either importing or generating a scenario, the user can perform a demo simulation run to show how the tool works. Instead of running a whole simulation, the user can also just display the scenario on a map, to give an idea of how the scenario looks like.



□ Import demo scenario from:

D:\...\...\Ordering Process Simulation\Scenarios\Demo_UseCase2_OBS.json

□ Import API settings from:

D:\...\...\Ordering Process Simulation\API_Settings.json

Simulation output

```
[2019-01-14 13:46:15] Scenario data is generated with the following parameters:
SETTINGS SUMMARY
Scenario id: Demo_UseCase2
Number of customers: 50
Average order size: 3
Average order duration in seconds: 300
Number of time windows: 4
Percentage of customers with order type A: 50%
Percentage of customers with order type B: 50%
Customer area bounds:
Minimum latitude: 52,19
Maximum latitude: 52,29
Minimum longitude: 6,75
Maximum longitude: 6,95
Depot location:
Latitude: 52,2394
Longitude: 6,8509
Number of working hours on a day: 8
Number of drivers available: 4
Number of vehicles of type 1: 2
Number of vehicles of type 2: 2
Number of customers until quick optimization run: 5
Number of simulation runs for this scenario: 1

[2019-01-14 13:46:15] API settings have been imported.

[2019-01-14 13:46:15] 50 new customers have been generated.

[2019-01-14 13:50:11] Distance and driving time matrices retrieved for 63001 entries. Elapsed time: 00:03:55.6153056

[2019-01-14 13:50:11] The scenario settings have been saved to the file D:\...\...\Ordering Process Simulation\Scenarios\Demo_UseCase2_OBS.json.
```

Vehicles of type 1
Number: 2
Capacity: 20
Speedfactor: 1
Costs per
Hour: 60
Distance unit: 1
Vehicle: 500

Vehicles of type 2
Number: 2
Capacity: 20
Speedfactor: 1
Costs per
Hour: 60
Distance unit: 1
Vehicle: 500

Depot information
Latitude: 52.2393971
Longitude: 6.8508709
Number of drivers: 4
Working hours: 8
Time windows: 4
Scenario label: Demo
Use case: UseCase2

ORTEC Timeslotting Service
Web address: https:\demo-ots-address.ortec.com\api
Key: 0123456789abcdefghijklmnopqrstuvwxyz
Profile: DemoUseCase2

COMTEC Vehicle Routing Service
Web address: https:\demo-cvrs-address.ortec.com\api
Key: 0123456789abcdefghijklmnopqrstuvwxyz
Profile: DemoUseCase2

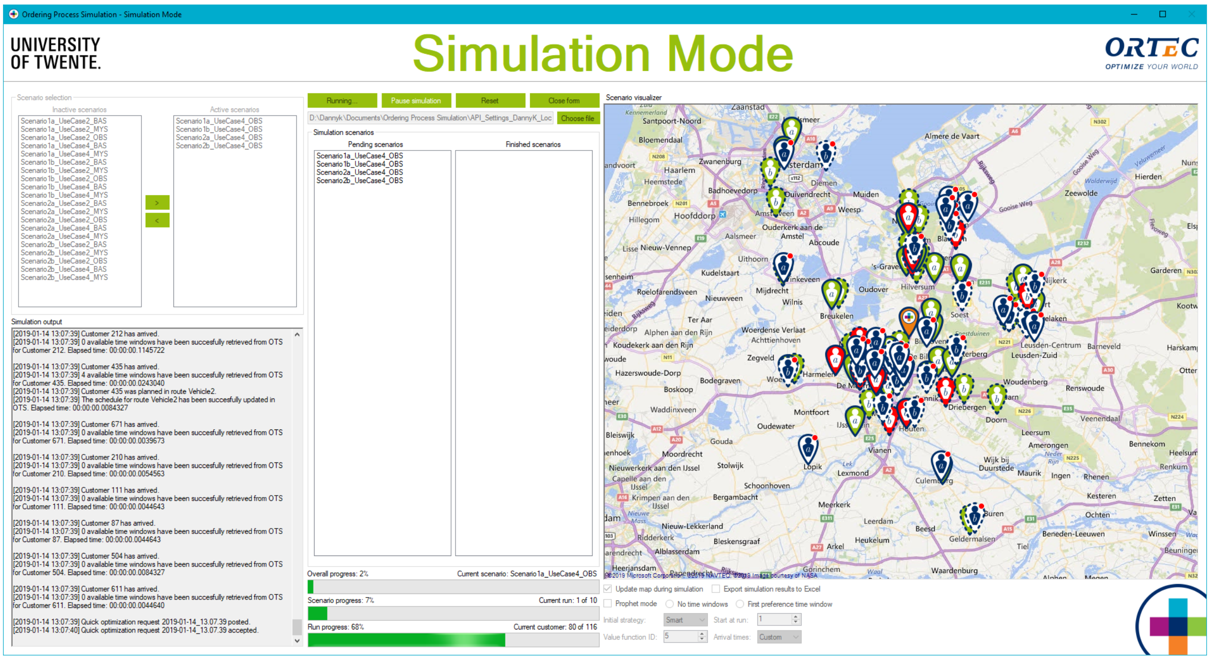
ORTEC Maps Service
Web address: https:\demo-oms-address.ortec.com\api
Key: 0123456789abcdefghijklmnopqrstuvwxyz
Profile: DemoUseCase2

Customer information
Number of customers: 50
Average order size: 3
Average order duration: 300
Percentage of type A: 50
Area
Min. latitude: 52.19
Max. latitude: 52.29
Min. longitude: 6.75
Max. latitude: 6.95

Settings
☑ Visualize map during simulation
Map zoom: 12
☑ Show emission zone
☑ Export simulation results
☑ Save API settings
□ Save demo scenario

Figure 5.3. *Screenshot of the demo mode*

Figure 5.4 visualizes the most important application mode of our simulation tool, the simulation mode. This application mode can be used to run multiple replications of different scenarios. In the section for scenario selection, under inactive scenarios, all available scenarios in the base application folder (a designated folder for the application located in the user's documents) are displayed. By selecting one or multiple inactive scenarios and then pressing the button with the arrow to the right, the user can activate the selected scenarios. When the user starts the simulation (by pressing the button that displays "Running..." in Figure 5.4, but before the start of the simulation displays "Start simulation"), these activated scenarios will be simulated. The bottom progress bar indicates how many customers have already arrived in the simulation of the current booking period. The middle progress bar indicates how many replications have been performed already. The top progress bar indicates the overall progress for all activated scenarios. Whenever all replications for a scenario have been performed, the scenario is moved to the section with finished scenarios. During the simulation the current scenario and replication is displayed on a map. A green marker indicates a confirmed customer order, a red marker a customer order that could not be served. The orange marker with the ORTEC plus symbol indicates the location of the central depot. The section with simulation output gives the user information about what happens in the simulation run and serves for instance for verification purposes. During a replication all algorithms as described in Section 5.1 are run and at the end of each replication the results are exported, so the user can analyze the results afterwards.



Inactive scenarios:
Scenario1a_UseCase2_BAS
Scenario1a_UseCase2_MYS
Scenario1a_UseCase2_OBS
Scenario1a_UseCase4_BAS
Scenario1a_UseCase4_MYS
Scenario1b_UseCase2_BAS
Scenario1b_UseCase2_MYS
Scenario1b_UseCase2_OBS
Scenario1b_UseCase4_BAS
Scenario1b_UseCase4_MYS
Scenario2a_UseCase2_BAS
Scenario2a_UseCase2_MYS
Scenario2a_UseCase2_OBS
Scenario2a_UseCase4_BAS
Scenario2a_UseCase4_MYS
Scenario2b_UseCase2_BAS
Scenario2b_UseCase2_MYS
Scenario2b_UseCase2_OBS
Scenario2b_UseCase4_BAS
Scenario2b_UseCase4_MYS

Active scenarios:
Scenario1a_UseCase4_OBS
Scenario1b_UseCase4_OBS
Scenario2a_UseCase4_OBS
Scenario2b_UseCase4_OBS

D:\Dannyk\Documents\Ordering Process Simulation\API_Settings_DannyK_Loc

Pending scenarios:
Scenario1a_UseCase4_OBS
Scenario1b_UseCase4_OBS
Scenario2a_UseCase4_OBS
Scenario2b_UseCase4_OBS

Simulation output

```
[2019-01-14 13:07:39] Customer 212 has arrived.
[2019-01-14 13:07:39] 0 available time windows have been succesfully retrieved from OTS
for Customer 212. Elapsed time: 00:00:00.1145722

[2019-01-14 13:07:39] Customer 435 has arrived.
[2019-01-14 13:07:39] 4 available time windows have been succesfully retrieved from OTS
for Customer 435. Elapsed time: 00:00:00.0243040
[2019-01-14 13:07:39] Customer 435 was planned in route Vehicle2.
[2019-01-14 13:07:39] The schedule for route Vehicle2 has been succesfully updated in
OTS. Elapsed time: 00:00:00.0084327

[2019-01-14 13:07:39] Customer 671 has arrived.
[2019-01-14 13:07:39] 0 available time windows have been succesfully retrieved from OTS
for Customer 671. Elapsed time: 00:00:00.0039673

[2019-01-14 13:07:39] Customer 210 has arrived.
[2019-01-14 13:07:39] 0 available time windows have been succesfully retrieved from OTS
for Customer 210. Elapsed time: 00:00:00.0054563

[2019-01-14 13:07:39] Customer 111 has arrived.
[2019-01-14 13:07:39] 0 available time windows have been succesfully retrieved from OTS
for Customer 111. Elapsed time: 00:00:00.0044643

[2019-01-14 13:07:39] Customer 87 has arrived.
[2019-01-14 13:07:39] 0 available time windows have been succesfully retrieved from OTS
for Customer 87. Elapsed time: 00:00:00.0044643

[2019-01-14 13:07:39] Customer 504 has arrived.
[2019-01-14 13:07:39] 0 available time windows have been succesfully retrieved from OTS
for Customer 504. Elapsed time: 00:00:00.0084327

[2019-01-14 13:07:39] Customer 611 has arrived.
[2019-01-14 13:07:39] 0 available time windows have been succesfully retrieved from OTS
for Customer 611. Elapsed time: 00:00:00.0044640

[2019-01-14 13:07:39] Quick optimization request 2019-01-14_13.07.39 posted.
[2019-01-14 13:07:40] Quick optimization request 2019-01-14_13.07.39 accepted.
```

Overall progress: 2% Current scenario: Scenario1a_UseCase4_OBS
Scenario progress: 7% Current run: 1 of 10
Run progress: 68% Current customer: 80 of 116

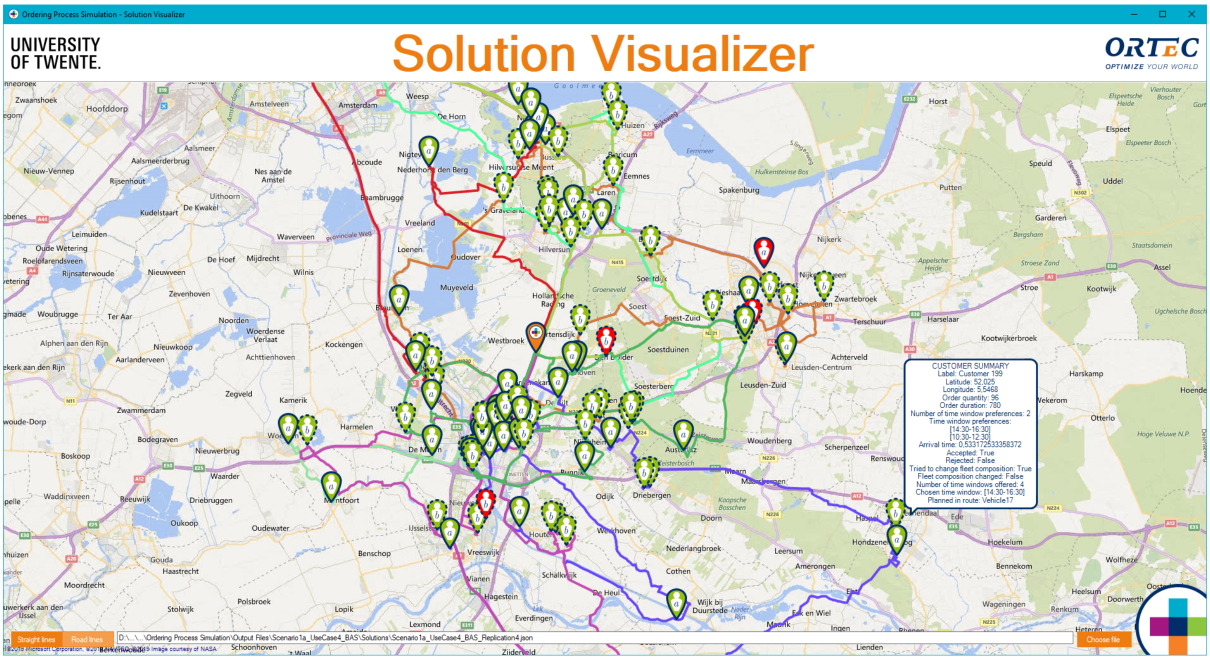
☑ Update map during simulation ☐ Export simulation results to Excel
☐ Prophet mode ○ No time windows ○ First preference time window
Initial strategy: Smart Start at run: 1
Value function ID: 5 Arrival times: Custom

©2019 Microsoft Corporation, ©2019 NAVTEQ, ©2019 Image courtesy of NASA

Figure 5.4. *Screenshot of the simulation mode*

Map rights: ©2019 Microsoft Corporation, ©2019 NAVTEQ, ©2019 Image courtesy of NASA, plugin by GMap.NET

Figure 5.5 displays the last mode, the solution visualizer. For each replication our simulation tool exports a JSON file that contains all information that is required to display the solution found for that replication on a map. We export for instance all the customers, with for each customer whether they are served or not and many other output metrics that were discussed in Section 5.1. We also export the final delivery routes that are formed, as well as other solution metrics. The solution visualizer mode enables us to import any solution file generated by our simulation tool and visualize the final routing plan that is formed. This application mode is quite useful, especially for verification purposes and gaining credibility for our simulation model.



Ordering Process Simulation - Solution Visualizer

Solution Visualizer

CUSTOMER SUMMARY
Label: Customer 199
Latitude: 52,025
Longitude: 5,5468
Order quantity: 96
Order duration: 780
Number of time window preferences: 2
Time window preferences:
[14:30-16:30]
[10:30-12:30]
Arrival time: 0,533172533358372
Accepted: True
Rejected: False
Tried to change fleet composition: True
Fleet composition changed: False
Number of time windows offered: 4
Chosen time window: [14:30-16:30]
Planned in route: Vehicle17

D:\...\...\Ordering Process Simulation\Output Files\Scenario1a_UseCase4_BAS\Solutions\Scenario1a_UseCase4_BAS_Replication4.json

Figure 5.5. *Screenshot of the solution visualizer mode*
Map rights: ©2019 Microsoft Corporation, ©2019 NAVTEQ, ©2019 Image courtesy of NASA, plugin by GMap.NET

5.3. Scenarios

In this section we describe the scenarios for which we do computational experiments to put our hypotheses to a test. We have two different scenarios that we consider. Section 5.3.1 describes the first scenario in terms of the inputs as defined in Section 5.1, Section 5.3.2 presents the second scenario. Section 5.3.3 gives a brief motivation for the number of replications we choose to perform. Subsequently, Section 5.3.4 describes the input distributions we use to determine the number of customers that arrive during the booking period for a replication and to determine the arrival times of the customers. Section 5.3.5 presents the way in which we model the customer choice behavior. Finally, Section 5.3.6 explains how the parameters for the strategy BAS are tuned in our experiments.

5.3.1. Scenario 1

The first scenario we consider is based on real data of an ORTEC client that resembles an e-retailer of our context. This makes it a perfect test-case to explore how our strategies perform. The data is adapted to fit our e-retailer case. For instance, the original client makes use of several depots. We replaced all depots by one central depot that is located somewhere around the center of all customer locations. Also, the original data lacked a classification of order types. We therefore randomly assigned the orders to either Order Type A or Order Type B. Finally, the customers' time window preferences are obviously not known. Section 5.3.5 explains the way in which we modeled this behavior both for Scenario 1 and Scenario 2. The central depot for this scenario is located somewhere in the province Utrecht of The Netherlands. The customer locations are spread around the depot, and they cover Dutch cities as Utrecht, Amsterdam, Amersfoort and Hilversum. Figure 5.6 presents a visualization of the customer and depot locations, generated by our simulation tool. The orange marker with the ORTEC plus symbol in the middle represents the location of the central depot for this scenario.

Table 5.1 presents a summary of the characteristics of the customers in this scenario. For Scenario 1 we distinguish between Scenario 1a and Scenario 1b. The only difference between the two scenarios is that for Scenario 1b the percentage of customer orders that arrive of Order Type A is 50% larger than for Scenario 1a. For Scenario 1a this percentage equals the percentage from historical data (the whole dataset of customers), as computed with Equation (4.1). We make this distinction between Scenario 1a and

Scenario 1b to study the effect of our strategies in cases where our expectations with regard to the ratio of customer orders of Order Type A and customer orders of Order Type B match reality, and cases where they do not match reality.

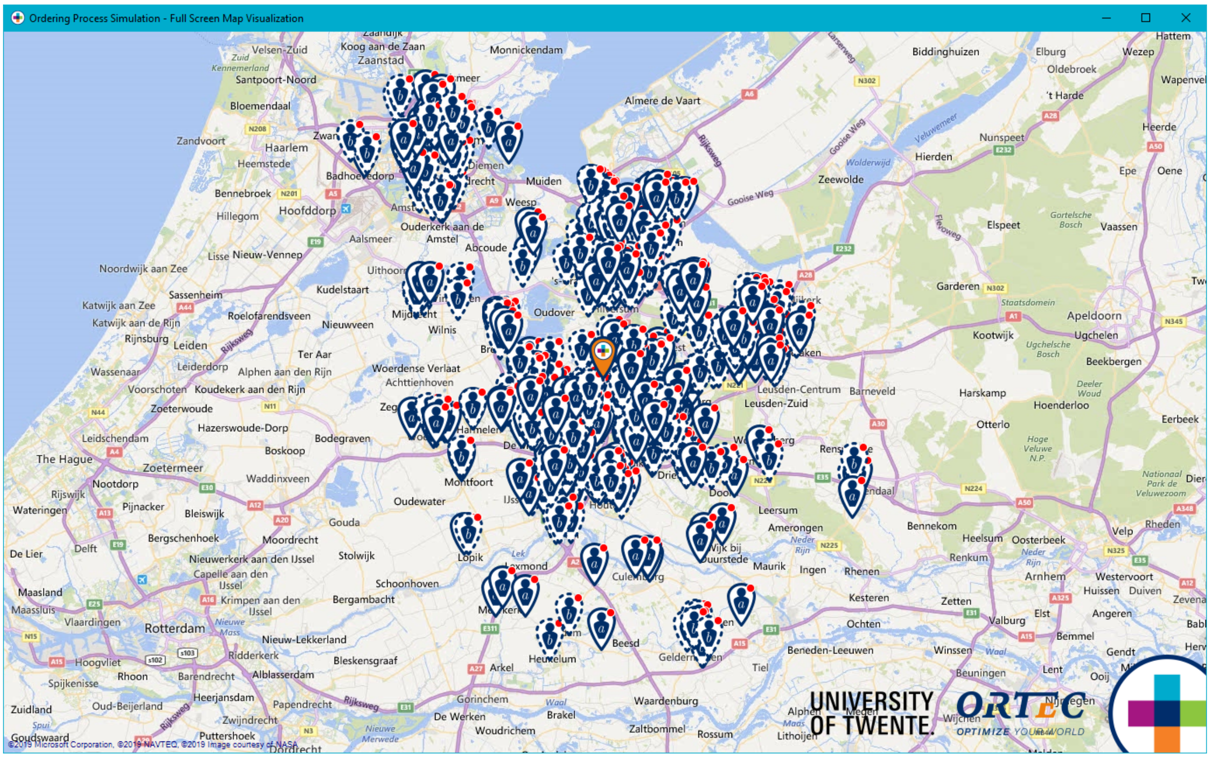

Ordering Process Simulation - Full Screen Map Visualization
UNIVERSITY OF TWENTE.
ORTEC
OPTIMIZE YOUR WORLD
©2019 Microsoft Corporation, ©2019 NAVTEQ, ©2019 Image courtesy of NASA

Figure 5.6. *Customer and depot locations for Scenario 1*
Map rights: ©2019 Microsoft Corporation, ©2019 NAVTEQ, ©2019 Image courtesy of NASA, plugin by GMap.NET

Due to technical restrictions on the cloud capacity for ORTECs cloud services that is available for simulation purposes, we limit the average number of customers that arrive during the booking period to 100. We discuss the input distribution for the number of customers that arrive in a replication and the input distribution used to determine the arrival times of the customers in Section 5.3.4.

Table 5.1. *Characteristics of the customers for Scenario 1*

Characteristic	**Scenario 1a**	**Scenario 1b**
Expected number of customers	100	100
Observed average % Order Type A	47.12%	70.68%
Observed average % Order Type B	52.88%	29.32%
Historical % Order Type A	47.12%	47.12%
Historical % Order Type B	52.88%	52.88%
Average order quantity	62 kilograms	62 kilograms
Average order duration	638 seconds	638 seconds

Table 5.2 presents the characteristics of the drivers for Scenario 1, and Table 5.3 the characteristics of the different vehicle types. Note that not all vehicle types are used in all use cases. Table 5.4 presents the characteristics of the central depot for Scenario 1 and finally Table 5.5 gives some general characteristics of the scenario.

Table 5.2. *Characteristics of the drivers for Scenario 1*

Characteristic	**Driver type 1**
Number of drivers	8
Driver skills	Vehicle Type 1, 2, 3
Maximum number of working hours	8

Table 5.3. *Characteristics of the vehicles for Scenario 1*

Characteristic	Vehicle Type 1	Vehicle Type 2	Vehicle Type 3
Number of vehicles	8	8	4
Costs per hour	€20.00	€20.00	€20.00
Costs per kilometer	€0.20	€0.20	€0.26
Setup costs	€ 5 .00	€ 5 .00	€16 .00
Capabilities	Order Type A	Order Type B	Order Type A, B
Capacity	500 kilograms	500 kilograms	1000 kilograms

Table 5.4. *Characteristics of the central depot for Scenario 1*

Characteristic	Central depot
Depot latitude	52.1422
Depot longitude	5.1568
Available time windows	[9:00-11:00], [11:00-13:00], [13:00-15:00], [15:00-17:00]

Table 5.5. *General characteristics for Scenario 1*

Characteristic	Scenario 1a	Scenario 1b
Id format	Scenario1a_[*Use case*]_[*Strategy*]	Scenario1b_[*Use case*]_[*Strategy*]
Use case	UseCase2, UseCase4	UseCase2, UseCase4
Strategy	OBS, MYS, BAS	OBS, MYS, BAS
Customers until optimization	20	20
Number of replications	10	10

For all experiments with Scenario 1 we use a length of the booking period of one weekday. So, customers in our context always order for the next day. Our methods do not require this, because we express the time as a percentage of the length of the booking period. However, the input distributions that we use make more sense in practice when we consider the length of one weekday.

5.3.2. Scenario 2

Figure 5.7 displays a visualization of Scenario 2 as generated by our simulation tool. Again, the orange marker with the ORTEC plus symbol represents the location of the central depot.

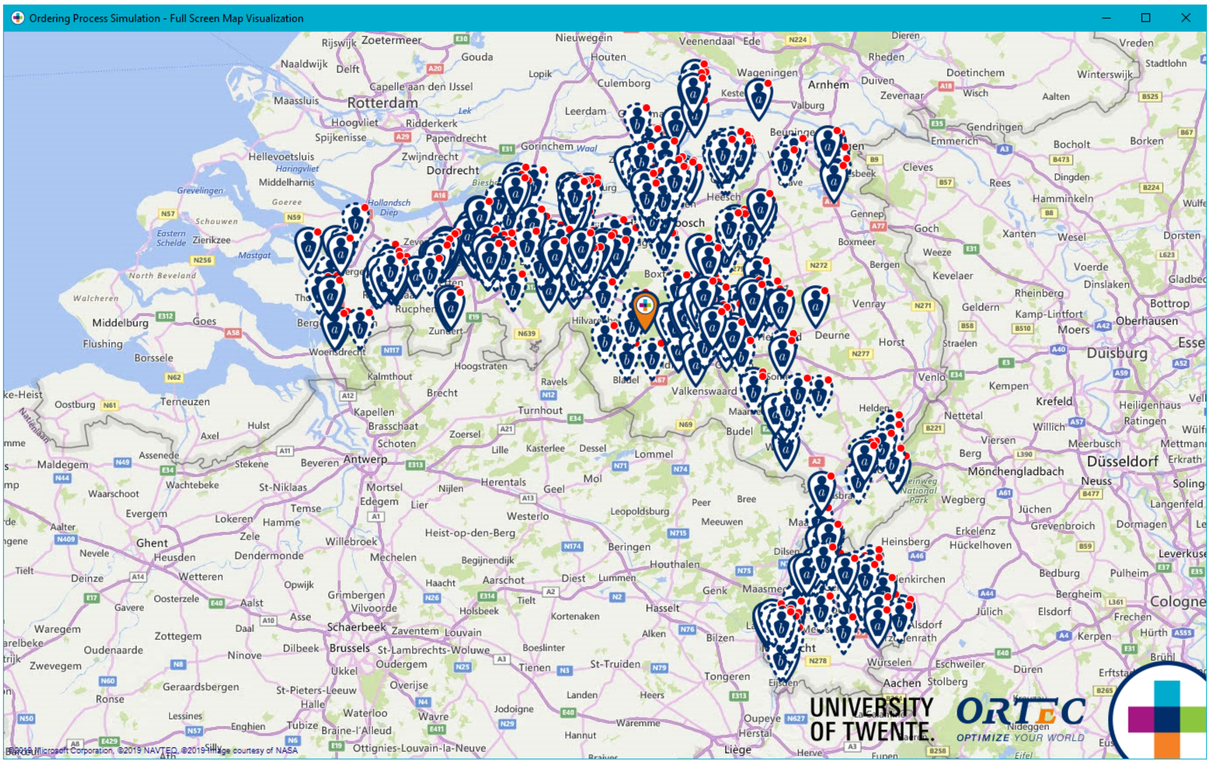


Figure 5.7. *Customer locations and scenario details for Scenario 2*
Map rights: ©2019 Microsoft Corporation, ©2019 NAVTEQ, ©2019 Image courtesy of NASA, plugin by GMap.NET

Just as our first scenario, the second scenario we consider is also based on real data of an ORTEC client that resembles an e-retailer of our context. We applied the same modifications to this dataset as to the dataset for Scenario 1. However, Scenario 2 considers different customers, with a different spread around the central depot. The central depot is now located in the province Noord-Brabant of the Netherlands. The customer locations are spread over the south of the Dutch provinces Utrecht and Gelderland, as well as over the whole provinces Noord-Brabant and Limburg. The locations are now further away from the central depot, the average distance from any customer location to the central depot is 47.63 kilometer (against 18.08 kilometer for Scenario 1).

Table 5.6 presents a summary of the characteristics of the customers in this scenario. For Scenario 2 we again distinguish between Scenario 2a and Scenario 2b, with the same difference as described before. Just as for Scenario 1, we limit the average number of customers that arrive during the booking period to 100 because of the technical restrictions on the cloud capacity for ORTECs cloud services that is available for simulation purposes. For Scenario 2 we also discuss the input distribution for the number of customers that arrive in a replication and the input distribution used to determine the arrival times of the customers in Section 5.3.4.

Table 5.6. *Characteristics of the customers for Scenario 2*

Characteristic	Scenario 2a	Scenario 2b
Expected number of customers	100	100
Observed average % Order Type A	51.55%	77.32%
Observed average % Order Type B	48.45%	22.68%
Historical % Order Type A	51.55%	51.55%
Historical % Order Type B	48.45%	48.45%
Average order quantity	59 kilograms	59 kilograms
Average order duration	559 seconds	559 seconds

Table 5.7 displays the characteristics of the drivers for Scenario 2, and Table 5.8 presents the characteristics of the different vehicle types. Again, note that not all vehicle types are used in all use cases. In Table 5.9 the characteristics of the central depot for Scenario 2 are shown and Table 5.10 presents some general characteristics of the scenario. Just as for the experiments with Scenario 1, we use a length of the booking period of one weekday for Scenario 2 as well.

Table 5.7. *Characteristics of the drivers for Scenario 2*

Characteristic	Driver type 1
Number of drivers	8
Driver skills	Vehicle Type 1, 2, 3
Maximum number of working hours	8

Table 5.8. *Characteristics of the vehicles for Scenario 2*

Characteristic	Vehicle Type 1	Vehicle Type 2	Vehicle Type 3
Number of vehicles	8	8	4
Costs per hour	€20.00	€20.00	€20.00
Costs per kilometer	€0.20	€0.20	€0.26
Setup costs	€92.00	€92.00	€16 .00
Capabilities	Order Type A	Order Type B	Order Type A, B
Capacity	500 kilograms	500 kilograms	900 kilograms

Table 5.9. *Characteristics of the central depot for Scenario 2*

Characteristic	Central depot
Depot latitude	51.4650
Depot longitude	5.2774
Available time windows	[9:00-11:00], [11:00-13:00], [13:00-15:00], [15:00-17:00]

Table 5.10. *General characteristics for Scenario 2*

Characteristic	Scenario 2a	Scenario 2b
Id format	Scenario2a_[*Use case*]_[*Strategy*]	Scenario2b_[*Use case*]_[*Strategy*]
Use case	UseCase2, UseCase4	UseCase2, UseCase4
Strategy	OBS, MYS, BAS	OBS, MYS, BAS
Customers until optimization	20	20
Number of replications	10	10

5.3.3. Number of Replications

As pointed out by Law (2015), there exist several methods to determine how many independent replications we should perform. We carried out the sequential procedure that he explains, to obtain a certain precision for the confidence interval of the mean values of our KPIs over the replications we perform. We focused on our two main KPIs, the percentage of customers that is served and the costs per customer that is served. However, when we use a common relative error of 5% for the real mean, (0.05/(1-0.05) for the sample mean) we find that after 20 replications we do still not have a relative error below this threshold. We can justify this phenomenon, because of the following: over the replications our maximum fleet capacity does not change, because we are bound by the number of drivers that is available. However, in every replication a different number of customer orders arrives, according to the distribution that we explain in Section 5.3.4. But the number of customers that we can serve with maximum fleet capacity does not vary that much. Therefore, the percentage of customers that is served is likely to vary a lot more over replications, especially when more customers arrive than can be served. This is indeed what we observe, the number of customers that are served shows a much lower variance than the percentage of customers that is served.

Preliminary simulation runs indicate that running one replication on average takes around 10 to 15 minutes. If we would need to perform 20 replications for experiment, we would probably need at least 200 hours of simulation running time with 60 experiments (Section 5.4). Given that typically we need to perform some runs again after changing some parameters for instance, these 200 hours would only be a very positive estimation of a lower bound for the real time that is required. We therefore decided to restrict the number of replications to 10 replications per experiment which seems plausible to us, given the time that we have available for simulation runs.

5.3.4. Input Distributions for Number of Customers and Customer Arrival Times

In our simulation model we make use of random distributions for both the number of customers that arrive during the booking period and the spread of the customer arrivals over the booking period, which we consider to be one day.

For both scenarios that we consider the average number of customers that arrive during the booking period equals 100 customers. In the different replications that we perform, we want to analyze the impact of having different amounts of customers arrive during the booking period. Therefore, we make use of a Poisson distribution to model the number of customers that arrive during the booking period. A Poisson distribution is commonly used to model the number of items in a batch of random size (Law, 2015), which describes our case quite well. In case we would use the Poisson(100) distribution, the spread of possible realizations would not be very large. For instance, with approximately 95% probability the number of customers drawn for a certain booking period lies between 80 and 120 for the Poisson(100) distribution. To ensure a somewhat larger spread for our experiments we therefore model the number of customers that arrive with a Poisson(50) distribution and we multiply the realization by 2. To get an idea of the impact, the probability that a realization is between 80 and 120 is around 85% for this distribution. Figure 5.8 presents the resulting probability distribution which we use to determine for each replication how many customers arrive during the booking period. In Appendix A.1 we validate the implementation of this distribution in our simulation tool and we show that the hypothesis that the realizations have the correct distribution cannot be rejected at a pre-defined confidence level.

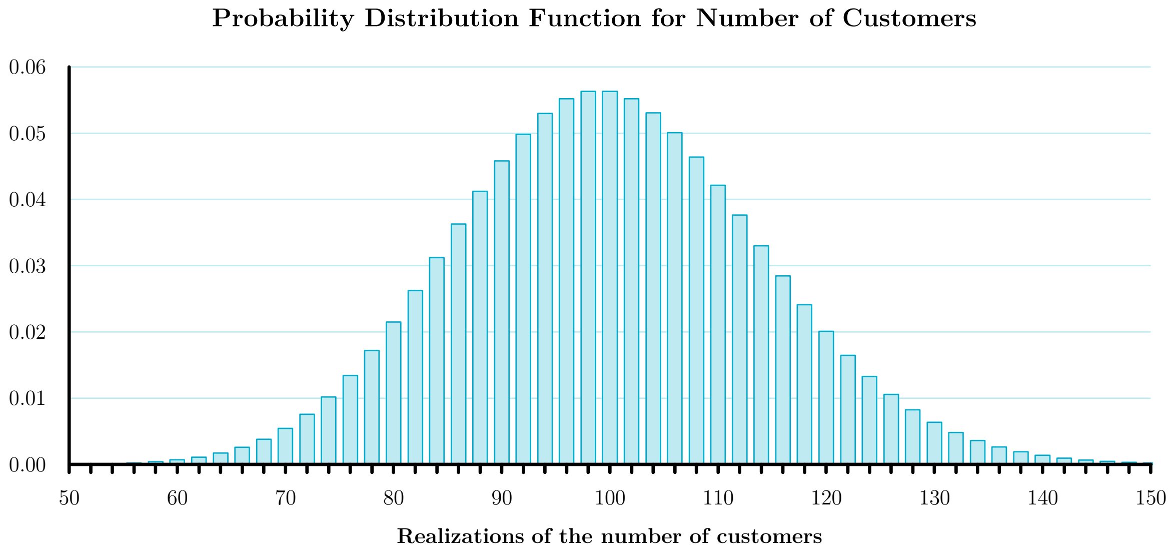


Figure 5.8. *Probability distribution used to determine how many customers arrive during a booking period*

For the spread of customer arrivals over the booking period of one day, we primarily make use of an empirical distribution that reflects common practices among customers of e-retailers during weekdays. Figure 5.9 displays the arrival frequency for each hour of the day in case of 100 customers that arrive during the booking period of one day.

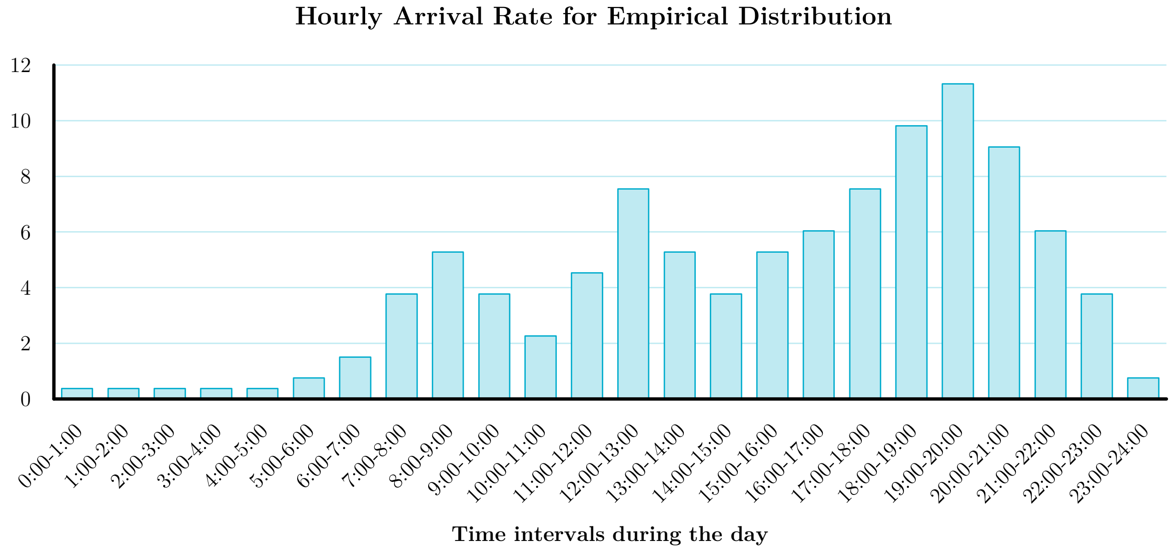


Figure 5.9. *Empirical pattern of average arrival rate per hour, expected number of customers = 100*

We see that from midnight to around 6AM hardly any customers arrive. From 6AM onward more customer orders start coming in, due to people that wake up early to go to work and want to place an order quickly before they leave. We see the first peek between 8AM and 9AM, when most people are awake and still find some time to do a bit of online shopping. The arrival rate then slowly reduces, because most potential customers have started working and are busy. The arrival rate then grows to a peek again between 12AM and 1PM, because people usually have their lunchbreaks during these hours. After the lunch peek the arrival rate drops a bit but starts increasing again after a few hours due to the potential customers that arrive home early and do their shopping as soon as they come home. The arrival rate reaches its highest peak just after dinner time, when most people have arrived at home and take some time to do their shopping after dinner is finished and the dishes are done. Then in the course of the evening the hourly arrival rate drops again, because most customers that wanted to place an order have already done so. In Appendix A.2 we validate the implementation in our simulation tool of this empirical distribution for the spread of the customer arrivals during the booking period. Besides that, we perform a check if the spread of arrival times realized by our simulation tool is statistically valid given the input distribution for a certain confidence level.

For most of the experiments that we carry out to put our hypothesis to a test we make use of the empirical distribution of customer arrival times. Only for our fourth hypothesis, we make use of two different distributions to analyze the impact of a changing arrival time spread. Figure 5.10 shows the first pattern that we consider, which is based on an increasing hourly arrival rate over the booking period and 100 customers that are expected to arrive during the entire booking period.

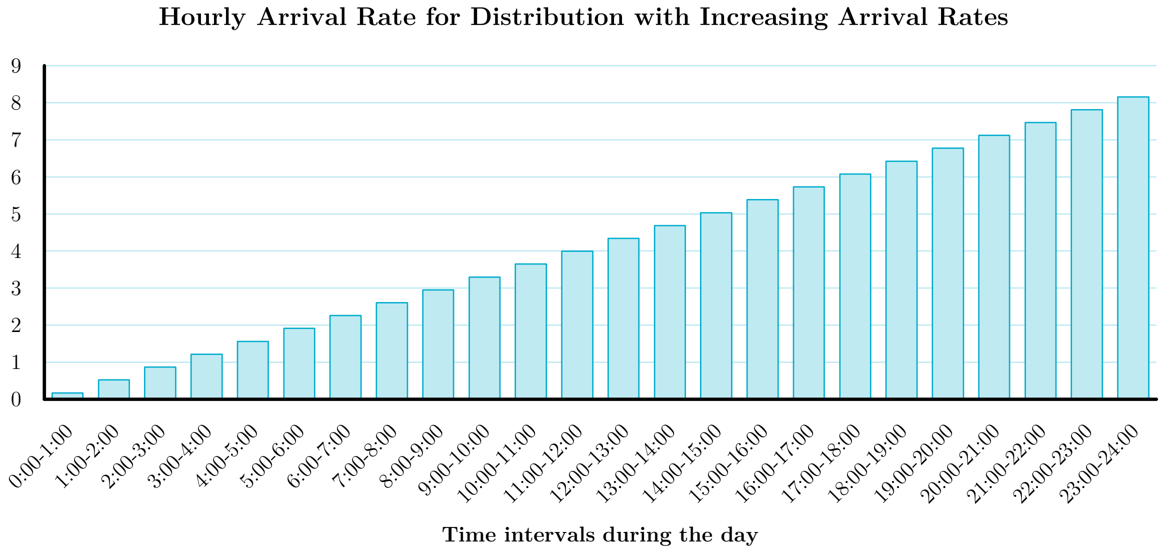


Figure 5.10. *Pattern for increasing average arrival rate per hour, expected number of customers = 100*

Although we display the spread for a booking period of one day because that is the length of the booking period in our experiments, this distribution of the customer arrival spread could very well represent a booking period of a longer period. For instance, if people can order their groceries for a certain day already 50 days in advance, we typically observe an increasing rate of customer orders that arrive as the booking period advances. Therefore, we do some experiments with this arrival time distribution for our last hypothesis. We validate the implementation of the distribution with increasing arrival rates in Appendix A.3.

The second alternative pattern that we consider is a pattern in which the hourly customer arrival rate remains constant over the booking period of one day. This uniform distribution of the customer arrival rates over the booking period is again used in experiments for our fourth hypothesis, to show the impact of inaccurate forecasting of the spread of customer arrival times. The hourly customer arrival rate when 100 customers are expected to arrive during the booking period of one day equals 4.1667 for each hour of the day. In Appendix A.4 we validate the way in which we implemented this distribution in our simulation tool.

5.3.5. Customer Choice Behavior

As we already pointed out in Chapter 2, there exist numerous ways to model customer choice behavior. Unfortunately, it is very difficult to obtain accurate data on customer choice behavior. We can analyze a lot of data of companies similar to an e-retailer in our context, but this only gives us information about the final selection of the customers whose orders were confirmed. The data do not give us any information about the customers' preferences, neither do they give us any information about the preferences of the customers that did not select a time window at all and did therefore not confirm their order.

Due to this lack of information on customer choice behavior, we decided not to distinguish between time windows in terms of customer preference. In other words, each time window is on average equally preferred by a random customer. In both Scenario 1 and Scenario 2 we consider 4 time windows. For each customer we ranked the 4 time windows according to a random score between 0 and 1 assigned to each time window. The higher the score, the higher the rank of the time window on the preference list of the customer. This method ensures that on average, each time window is equally attractive to customers.

After ranking the time windows for customers, we distinguish between the number of time windows on the preference list of a customer. Not all customers prefer the same number of time windows. Some customers may not care when their order is delivered during the day, resulting in 4 time windows on the preference list. However, other customers may for instance only be available during the morning part of the day, resulting in only 2 time windows on the preference list. This means that if we offer a time window in the afternoon to them, they will refuse to confirm their order and withdraw instead. We implemented this by for each customer drawing a random length between 1 and 4 time windows for the preference list, with an equal probability for each value between 1 and 4 (including both). In this way, time window preferences are generated for all customers that arrive during a booking period in our replications. Of course, models do exist that are much more advanced when it comes to modeling customer choice behavior. However, it is out of scope for our research to dig deeper into this topic.

5.3.6. Parameter Tuning for BAS

As mentioned in Chapter 4, we can dedicate a whole thesis to the process of obtaining the best parameter tuning for the application of BAS in certain cases. It is a topic that on its own has to do with so many aspects, that it is not feasible for us to focus on how to tune the parameters perfectly for our scenarios. Besides that, it is also not our core focus to obtain a good parameter tuning for two example cases of the e-retailer case. We are more interested in exploring what the benefits can be of the strategy that we design, instead of obtaining the best possible results for the scenarios that serve as test-cases.

However, in order to not just making up some random parameter values, we performed some preliminary simulation runs for some of the experiments that we define in Section 5.4. We performed these runs for different values for all parameters that BAS uses, thus exploring the spaces with possible values for the parameters. Our preliminary results indicated that the configuration as presented in Table 5.11 overall resulted in the best performance for our scenarios.

Of course, this is not a guarantee that this configuration is indeed the best possible one for all scenarios and all experiments. However, as it would consume a lot of time (which we would still need, to run all experiments with 10 replications) to investigate if we could achieve a better performance with different parameter configurations, we decided to stick to the configuration from Table 5.11.

Table 5.11. *Parameter values used for BAS in our experiments*

Parameter	Value	Parameter	Value
p_1	0.10	p_6	0.70
p_2	0.10	p_7	0.10
p_3	0.10	p	0.50
p_4	2	p	0.05
p_5	0.90	p_{10}	0.80

To calculate the unattractiveness score for a customer, to compare with p_{10}, we need to define a measure for this score. In our context, we decided to consider the order quantity of the customer as a measure of unattractiveness. As we deal with a fixed maximum fleet capacity given the number of drivers that are available, and on average more demand comes in than can be served with this capacity, we can on average not serve all customers. It then becomes unattractive to accept large customer orders, because instead of those large orders we may accept multiple smaller orders for instance. This results in a better performance in terms of the average percentage of customers that can be served, which is the primary objective that we maximize. Therefore, we define the unattractiveness score of a customer as the percentage of historical customer orders with a smaller order quantity than this customer. In other words, if more than 80% of the historical customer orders has a smaller order quantity, this customer is considered as unattractive.

5.4. Experiments

This section presents an overview of the experiments that are performed to validate our hypotheses. We characterize experiments based on the scenario, the use case, the strategy and the distribution used for the customer arrival times. Section 5.4.1 presents experiments that are linked to analyzing our hypotheses. Section 5.4.2 presents some additional experiments, that are mainly intended for benchmarking purposes.

5.4.1. Hypotheses

For our first hypothesis we carry out experiments only for Use Case 2 and our strategies MYS and OBS. Table 5.12 presents the characteristics of all 8 experiments for this hypothesis.

Table 5.12. *Experiments carried out for Hypothesis 1*

Id	Scenario	Use Case	Strategy	Distribution Arrival Times
Scenario1a_UseCase2_OBS	Scenario 1a	Use Case 2	OBS	Empirical
Scenario1a_UseCase2_MYS	Scenario 1a	Use Case 2	MYS	Empirical
Scenario1b_UseCase2_OBS	Scenario 1b	Use Case 2	OBS	Empirical
Scenario1b_UseCase2_MYS	Scenario 1b	Use Case 2	MYS	Empirical
Scenario2a_UseCase2_OBS	Scenario 2a	Use Case 2	OBS	Empirical
Scenario2a_UseCase2_MYS	Scenario 2a	Use Case 2	MYS	Empirical
Scenario2b_UseCase2_OBS	Scenario 2b	Use Case 2	OBS	Empirical
Scenario2b_UseCase2_MYS	Scenario 2b	Use Case 2	MYS	Empirical

For our second hypothesis, we compare the performance of BAS and MYS. However, we also carry out experiments for OBS, because the performance of OBS is used as a benchmark to compare both BAS and MYS with. We now do not only consider Use Case 2 but also Use Case 4, resulting in the 24 experiments as displayed in Table 5.13. The results of 8 experiments (dimmed in the table) are already available after doing the experiments for Hypothesis 1, so we need to run 16 new experiments for Hypothesis 2.

Table 5.13. *Experiments carried out for Hypothesis 2*

Id	Scenario	Use Case	Strategy	Distribution Arrival Times
Scenario1a_UseCase2_OBS	Scenario 1a	Use Case 2	OBS	Empirical
Scenario1a_UseCase2_MYS	Scenario 1a	Use Case 2	MYS	Empirical
Scenario1a_UseCase2_BAS	Scenario 1a	Use Case 2	BAS	Empirical
Scenario1b_UseCase2_OBS	Scenario 1b	Use Case 2	OBS	Empirical
Scenario1b_UseCase2_MYS	Scenario 1b	Use Case 2	MYS	Empirical
Scenario1b_UseCase2_BAS	Scenario 1b	Use Case 2	BAS	Empirical
Scenario2a_UseCase2_OBS	Scenario 2a	Use Case 2	OBS	Empirical
Scenario2a_UseCase2_MYS	Scenario 2a	Use Case 2	MYS	Empirical
Scenario2a_UseCase2_BAS	Scenario 2a	Use Case 2	BAS	Empirical
Scenario2b_UseCase2_OBS	Scenario 2b	Use Case 2	OBS	Empirical
Scenario2b_UseCase2_MYS	Scenario 2b	Use Case 2	MYS	Empirical
Scenario2b_UseCase2_BAS	Scenario 2b	Use Case 2	BAS	Empirical
Scenario1a_UseCase4_OBS	Scenario 1a	Use Case 4	OBS	Empirical
Scenario1a_UseCase4_MYS	Scenario 1a	Use Case 4	MYS	Empirical
Scenario1a_UseCase4_BAS	Scenario 1a	Use Case 4	BAS	Empirical
Scenario1b_UseCase4_OBS	Scenario 1b	Use Case 4	OBS	Empirical
Scenario1b_UseCase4_MYS	Scenario 1b	Use Case 4	MYS	Empirical
Scenario1b_UseCase4_BAS	Scenario 1b	Use Case 4	BAS	Empirical
Scenario2a_UseCase4_OBS	Scenario 2a	Use Case 4	OBS	Empirical
Scenario2a_UseCase4_MYS	Scenario 2a	Use Case 4	MYS	Empirical
Scenario2a_UseCase4_BAS	Scenario 2a	Use Case 4	BAS	Empirical
Scenario2b_UseCase4_OBS	Scenario 2b	Use Case 4	OBS	Empirical
Scenario2b_UseCase4_MYS	Scenario 2b	Use Case 4	MYS	Empirical
Scenario2b_UseCase4_BAS	Scenario 2b	Use Case 4	BAS	Empirical

For Hypothesis 3 we consider the difference between the performance of BAS and OBS (Large Initial Fleet). To that end, we need the results of the 12 experiments presented in Table 5.14. We only consider Use Case 4 in the experiments for this hypothesis. As we already obtain quite some relevant results from the experiments for Hypothesis 1 and Hypothesis 2, we only need to carry out 4 new experiments for Hypothesis 3. The experiments that can be re-used are dimmed in the table.

Table 5.14. *Experiments carried out for Hypothesis 3*

Id	Scenario	Use Case	Strategy	Distribution Arrival Times
Scenario1a_UseCase4_OBS	Scenario 1a	Use Case 4	OBS	Empirical
Scenario1a_UseCase4_OBS(Large)	Scenario 1a	Use Case 4	OBS (Large Initial Fleet)	Empirical
Scenario1a_UseCase4_BAS	Scenario 1a	Use Case 4	BAS	Empirical
Scenario1b_UseCase4_OBS	Scenario 1b	Use Case 4	OBS	Empirical
Scenario1b_UseCase4_OBS(Large)	Scenario 1b	Use Case 4	OBS (Large Initial Fleet)	Empirical
Scenario1b_UseCase4_BAS	Scenario 1b	Use Case 4	BAS	Empirical
Scenario2a_UseCase4_OBS	Scenario 2a	Use Case 4	OBS	Empirical
Scenario2a_UseCase4_OBS(Large)	Scenario 2a	Use Case 4	OBS (Large Initial Fleet)	Empirical
Scenario2a_UseCase4_BAS	Scenario 2a	Use Case 4	BAS	Empirical
Scenario2b_UseCase4_OBS	Scenario 2b	Use Case 4	OBS	Empirical
Scenario2b_UseCase4_OBS(Large)	Scenario 2b	Use Case 4	OBS (Large Initial Fleet)	Empirical
Scenario2b_UseCase4_BAS	Scenario 2b	Use Case 4	BAS	Empirical

For Hypothesis 4 we carry out some different experiments. First, we study the impact on the performance of BAS of a different distribution for the spread of customer arrivals over the booking period. Just as for all experiments as defined until now, we still consider that we can accurately forecast the average spread of the customer arrivals. Note that this does not mean that for every replication we know exactly when the customers arrive, as for each replication we have random realizations of the theoretical distribution of customer arrival times. To study the effect on the performance of BAS when we do not have accurate forecasts for the spread of customer arrivals over the booking period, we perform experiments to study two types of forecast errors: fixed forecast errors and fluctuating forecast errors. To keep our scope limited, we study the effect of forecast errors only for Scenario 1b, which is most relevant for ORTEC's practice.

When we consider the fixed forecast errors, we test the performance of BAS for forecasts that were 10%, 20% or 30% higher and lower than the correct forecasts according to the theoretical distribution of arrival times. Figure 5.11 shows the forecasted value of the cumulative percentage of customers that has arrived as a function of the time, for a booking period of one day and for all forecast errors we consider.

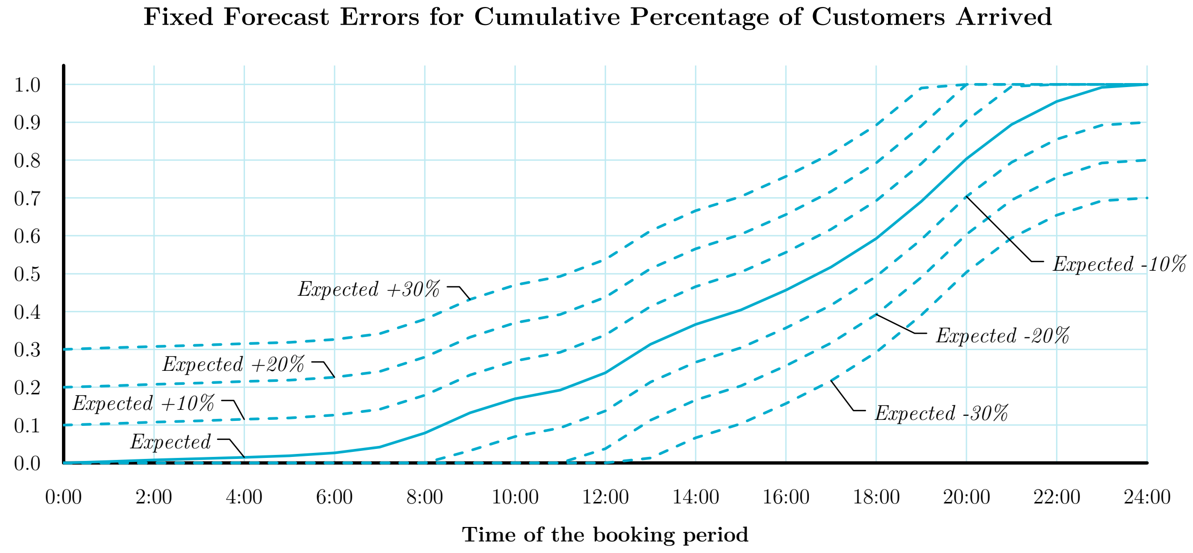


Figure 5.11. *Expected cumulative percentage of customers arrived at a given time with fixed forecast errors*

To clarify, suppose we are forecasting 20% higher than the correct forecast, and we expect (according to the theoretical distribution) that after 50% of the booking period 25% of the customers has arrived. We then deliberately use a forecast of 45% for the percentage of customers that has already arrived.

When we consider fluctuating forecast errors, we compare the performance of BAS without forecast errors to two different cases. For the first case we consider customer arrivals spread according to the empirical distribution (Section 5.3), but in our forecasts we assume the uniform distribution. The effect of this assumption in terms of the forecast for the cumulative percentage of customers that has arrived at a given point in time is displayed in Figure 5.12. For the second case we invert the two distributions. The uniform distribution is now the actual distribution, whereas the empirical distribution is the assumed one.

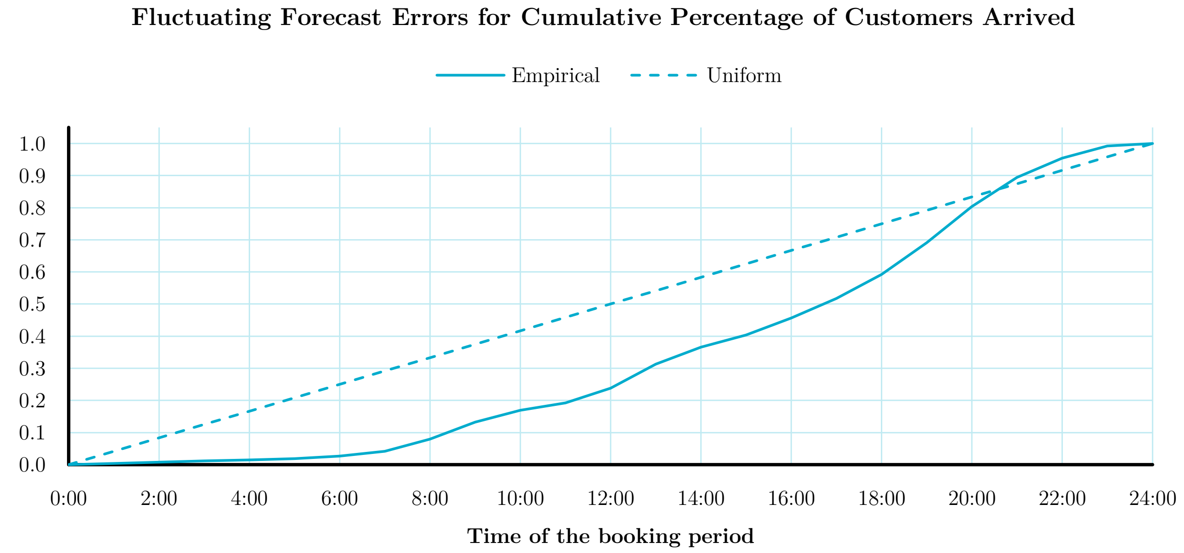


Figure 5.12. *Expected cumulative percentage of customers arrived at a given time with fluctuating forecast errors*

The resulting 20 experiments for Hypothesis 4 are displayed in Table 5.15. Of those 20 experiments 8 experiments (dimmed in the table) are carried out for an earlier hypothesis, leaving us with 12 new experiments to carry out. Counting up all the experiments we need to perform so far, we conclude that we need to carry out a total of 40 experiments to put our hypotheses to a test.

Table 5.15. Experiments carried out for Hypothesis 4

Id	Scenario	Use Case	Strategy	Dist. Arrival Times	Forecast Error
Scenario1a_UseCase4_OBS	Scenario 1a	Use Case 4	OBS	Empirical	None
Scenario1a_UseCase4_BAS	Scenario 1a	Use Case 4	BAS	Empirical	None
Scenario1a_UseCase4_BAS_Increasing	Scenario 1a	Use Case 4	BAS	Increasing	None
Scenario1b_UseCase4_OBS	Scenario 1b	Use Case 4	OBS	Empirical	None
Scenario1b_UseCase4_BAS	Scenario 1b	Use Case 4	BAS	Empirical	None
Scenario1b_UseCase4_BAS_Increasing	Scenario 1b	Use Case 4	BAS	Increasing	None
Scenario2a_UseCase4_OBS	Scenario 2a	Use Case 4	OBS	Empirical	None
Scenario2a_UseCase4_BAS	Scenario 2a	Use Case 4	BAS	Empirical	None
Scenario2a_UseCase4_BAS_Increasing	Scenario 2a	Use Case 4	BAS	Increasing	None
Scenario2b_UseCase4_OBS	Scenario 2b	Use Case 4	OBS	Empirical	None
Scenario2b_UseCase4_BAS	Scenario 2b	Use Case 4	BAS	Empirical	None
Scenario2b_UseCase4_BAS_Increasing	Scenario 2b	Use Case 4	BAS	Increasing	None
Scenario1b_UseCase4_BAS_-30%	Scenario 1b	Use Case 4	BAS	Empirical	Fixed, -30%
Scenario1b_UseCase4_BAS_-20%	Scenario 1b	Use Case 4	BAS	Empirical	Fixed, -20%
Scenario1b_UseCase4_BAS_-10%	Scenario 1b	Use Case 4	BAS	Empirical	Fixed, -10%
Scenario1b_UseCase4_BAS_+10%	Scenario 1b	Use Case 4	BAS	Empirical	Fixed, +10%
Scenario1b_UseCase4_BAS_+20%	Scenario 1b	Use Case 4	BAS	Empirical	Fixed, +20%
Scenario1b_UseCase4_BAS_+30%	Scenario 1b	Use Case 4	BAS	Empirical	Fixed, +30%
Scenario1b_UseCase4_BAS_Empirical	Scenario 1b	Use Case 4	BAS	Empirical	Fluctuating, Uniform
Scenario1b_UseCase4_BAS_Uniform	Scenario 1b	Use Case 4	BAS	Uniform	Fluctuating, Empirical

5.4.2. Additional Experiments

For benchmarking purposes, we perform a few additional computational experiments. In these experiments we evaluate the performance of two so-called prophet strategies. The strategies are characterized as prophet strategies, because they consider information regarding customer orders that will arrive in the future and take that information into account when constructing a solution for our problem. Of course, in real life this information is not available to an e-retailer. Therefore, the performance of these prophet strategies serves as benchmark for the performance of the strategies we designed in this research. The aim for both prophet strategies is to accept as many customer orders as possible and then minimizing the total delivery costs spent to deliver the customer orders.

The first prophet strategy does not consider any time window preferences of the customers and assumes that each customer would select any time window offered to him or her. This strategy knows upfront all customer orders that will arrive during the booking period and the sequence in which they come in. Therefore, we basically need to solve the HFVRPTW with all these customer orders, for all possible fleet compositions, to find out what the best performance is that we can achieve in terms of the percentage of customers served and the costs per customer that is served. We solve the problem with CVRS, configured in the same way as the final optimization step for our other strategies at the end of the booking period. We cannot guarantee that we obtain the optimal solution for all instances, because CVRS makes use of heuristics. However, we believe that it is a good approximation to obtain an upper bound for the performance of our strategies. Of course, in real life we can never achieve the performance of this first prophet strategy, because in real life the customers do have time window preferences and do not select every time window. Also, in practice we never know in which sequence the customers arrive, so we cannot reject customer orders based on information regarding the future, of which the prophet strategy does make use.

Our second prophet strategy is similar to the first one, with one main difference: instead of assuming that each customer would select any time window offered to him or her, we assume that a customer would only choose the first time window of his or her preference list. This strategy gives us an idea of how good we could perform in case we offer the highest possible customer satisfaction in terms of time window preferences.

We study the performance of our prophet strategies for Use Case 2 and Use Case 4, which are the use cases that we study in our hypotheses. This results in the 16 new experiments that are displayed in Table 5.16.

Table 5.16. *Experiments carried out for prophet strategies*

Id	Scenario	Use Case	Strategy
Scenario1a_UseCase2_Prophet(NTW)	Scenario 1a	Use Case 2	Prophet (No Time Windows)
Scenario1a_UseCase2_Prophet(FCTW)	Scenario 1a	Use Case 2	Prophet (First Choice Time Window)
Scenario1b_UseCase2_Prophet(NTW)	Scenario 1b	Use Case 2	Prophet (No Time Windows)
Scenario1b_UseCase2_Prophet(FCTW)	Scenario 1b	Use Case 2	Prophet (First Choice Time Window)
Scenario2a_UseCase2_Prophet(NTW)	Scenario 2a	Use Case 2	Prophet (No Time Windows)
Scenario2a_UseCase2_Prophet(FCTW)	Scenario 2a	Use Case 2	Prophet (First Choice Time Window)
Scenario2b_UseCase2_Prophet(NTW)	Scenario 2b	Use Case 2	Prophet (No Time Windows)
Scenario2b_UseCase2_Prophet(FCTW)	Scenario 2b	Use Case 2	Prophet (First Choice Time Window)
Scenario1a_UseCase4_Prophet(NTW)	Scenario 1a	Use Case 4	Prophet (No Time Windows)
Scenario1a_UseCase4_Prophet(FCTW)	Scenario 1a	Use Case 4	Prophet (First Choice Time Window)
Scenario1b_UseCase4_Prophet(NTW)	Scenario 1b	Use Case 4	Prophet (No Time Windows)
Scenario1b_UseCase4_Prophet(FCTW)	Scenario 1b	Use Case 4	Prophet (First Choice Time Window)
Scenario2a_UseCase4_Prophet(NTW)	Scenario 2a	Use Case 4	Prophet (No Time Windows)
Scenario2a_UseCase4_Prophet(FCTW)	Scenario 2a	Use Case 4	Prophet (First Choice Time Window)
Scenario2b_UseCase4_Prophet(NTW)	Scenario 2b	Use Case 4	Prophet (No Time Windows)
Scenario2b_UseCase4_Prophet(FCTW)	Scenario 2b	Use Case 4	Prophet (First Choice Time Window)

Finally, we perform some experiments to study the impact of intermediate optimization calls. The intermediate optimization calls consume a lot of time computational time and require a lot of effort in practice in terms of technical implementation. The reason for this is that we must ensure that while the optimization calls did not yet return a response, new customers can still be offered time windows. The question may therefore rise whether the impact of the intermediate optimization calls on the performance justifies these efforts. We test this for Use Case 2 and OBS, resulting in 8 required experiments, as shown in Table 5.17. The experiments that can be re-used from earlier hypotheses are dimmed in the table. We need to carry out 4 new experiments, bringing our grand total to 60 experiments.

Table 5.17. *Experiments carried out to study the impact of intermediate optimization calls*

Id	Scenario	Use Case	Strategy	Dist. Arrival Times
Scenario1a_UseCase2_OBS	Scenario 1a	Use Case 2	OBS	Empirical
Scenario1a_UseCase2_OBS(NoOpt)	Scenario 1a	Use Case 2	OBS (No Intermediate Optimization)	Empirical
Scenario1b_UseCase2_OBS	Scenario 1b	Use Case 2	OBS	Empirical
Scenario1b_UseCase2_OBS(NoOpt)	Scenario 1b	Use Case 2	OBS (No Intermediate Optimization)	Empirical
Scenario2a_UseCase2_OBS	Scenario 2a	Use Case 2	OBS	Empirical
Scenario2a_UseCase2_OBS(NoOpt)	Scenario 2a	Use Case 2	OBS (No Intermediate Optimization)	Empirical
Scenario2b_UseCase2_OBS	Scenario 2b	Use Case 2	OBS	Empirical
Scenario2b_UseCase2_OBS(NoOpt)	Scenario 2b	Use Case 2	OBS (No Intermediate Optimization)	Empirical

5.5. Conclusion

In this chapter we discussed our simulation approach. We defined the structure of our simulation model in Section 5.1 and we presented the simulation tool that we developed to run our simulations in Section 5.2. Furthermore, in Section 5.3 we defined the scenarios that we consider in our experiments, which we presented in Section 5.4. With the insights we obtained in this chapter we can now respond our fourth set of research questions:

4) How can we simulate the ordering process from the e-retailer case?

- *a. What are the inputs that we need for our simulations and how do we process them?*
- *b. How can we run our simulations?*
- *c. Which scenarios are we going to use as input data for our simulations?*
- *d. Which experiments do we define to validate our hypotheses?*

We model the ordering process of the e-retailer case with a discrete-event simulation model, in which the customer arrivals are considered as the events that trigger a change in the system state. The inputs of our simulations are mainly based on the entities that we defined in Chapter 3: customers, drivers, vehicles and a central depot. Besides that, we need several parameters that define the settings for our simulation runs. The set of all required inputs is called a scenario. The outputs of our simulations are defined in such a way that we can use them to calculate all KPIs from Chapter 3. The inputs are processed by several algorithms that carry out the simulation runs in such a way that the inputs are converted to the required outputs. Each simulation run, or replication, consists of one booking period and the final optimization by the end of the booking period.

To be able to carry out the simulation runs, we developed a simulation tool in C#. All algorithms that have been designed to carry out the simulations are implemented in this tool, that consists of four application modes: a scenario generator mode, a demo mode, a simulation mode and a solution visualizer mode. The tool communicates with ORTEC's software solutions via JSON requests whenever that is required during the simulations.

We consider two scenarios for which we perform computational experiments. Both scenarios are based on real data of an ORTEC client that resembles an e-retailer of our context. For both scenarios we consider two cases. The first one (a) represents a situation in which the observed percentage of customers for each order type is on average equal to the percentage of customers with that order type according to historical data. The second case (b) is a situation in which the observed ratio of customers of each order type does not equal the historical ratio. The scenarios differ further in the characteristics of the customers, the delivery fleet and the location of the central depot. For all scenarios we also define the distribution of the number of customers that arrive during the booking period and several possible distributions of the customer arrival times. Besides that, we also defined the way in which we model customer choice behavior for the two scenarios we consider. Finally, we tuned the parameters that we use for our strategy BAS in our experiments based on some preliminary results.

To conclude, we defined all the 40 experiments that are required to validate our hypotheses. Besides these 40 experiments, we defined a few additional experiments that allow us to obtain some interesting general insights that cannot be directly linked to one of our hypotheses. This results in a grand total of 60 computational experiments that we need to carry out.

6. Analysis of Computational Results

In this chapter we analyze the results which we obtain from our computational experiments. The structure of this chapter is based on the hypotheses we defined in Chapter 3. In Section 6.1, 6.2, 6.3 and 6.4 we analyze the computational results of the experiments we performed when investigating the hypotheses. In Section 6.5 we report some general results that are not necessarily linked to our hypotheses, but may provide valuable insights for practice. More detailed results for all sections are presented in Appendix B.

6.1. Hypothesis 1

> *In applications of Use Case 2 where forecasts (based on historical data) for the ratio of customer orders of Order Type A and Order Type B are not accurate, using a myopic dynamic strategy leads to a better performance compared to a static strategy. However, when the forecasts are accurate, using a myopic dynamic strategy does not necessarily improve the performance.*

The goal of investigating our first hypothesis is to show the benefits of using a dynamic strategy that can change the fleet composition during the booking period. Recall from Chapter 5 that Scenario 1a and Scenario 2a are scenarios in which we have accurate forecasts based on historical data (with regard to the ratio of order types of arriving customers). Scenario 1b and Scenario 2b are scenarios in which those forecasts are inaccurate. To give an idea of the difference in performance for OBS (the static strategy) and MYS (the dynamic strategy), we compare the average percentage of customers that is served (Figure 6.1) and the average costs per customer that is served (Figure 6.2). In all graphs where the title contains that OBS equals 0%, the performance of the strategies in the graph is relative to OBS. For instance, if OBS served 80% of the customers and MYS 88%, the graph displays a score of 10% for MYS for this KPI. Or if the average costs per customer are €100 for OBS and € 5 for MYS, the graph displays a score of -5% for MYS. Note that for the average percentage of customers that is served, a strategy outperforms OBS whenever the scaled average percentage of customers served for that strategy is positive, because we try to maximize this KPI. For the average costs per customer that is served, a strategy outperforms OBS whenever the scaled average costs per customer served are negative, as we want to minimize this KPI.

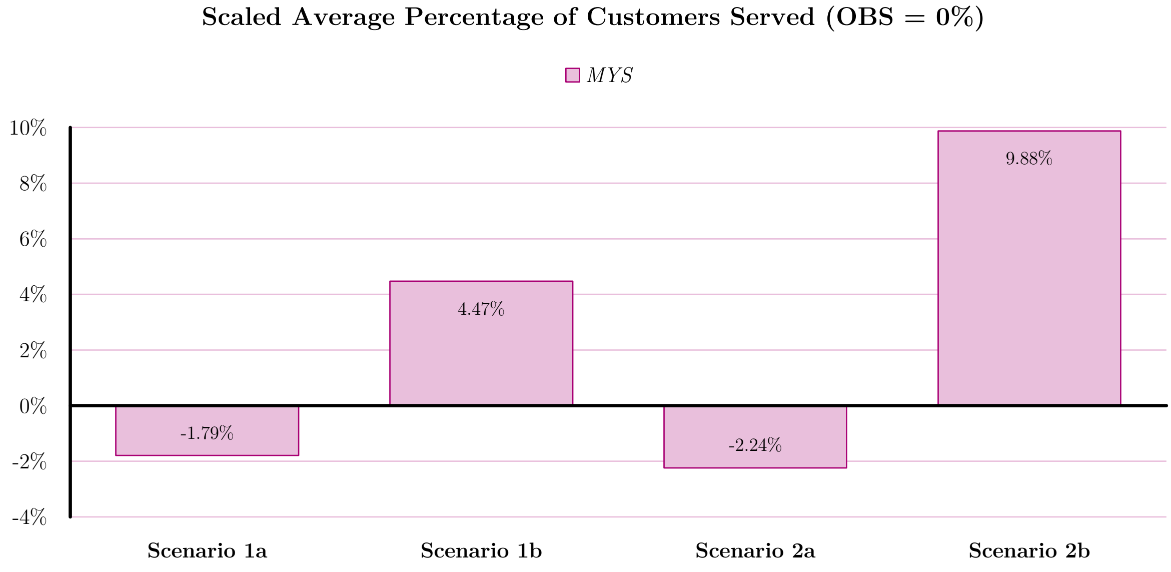

Scaled Average Percentage of Customers Served (OBS = 0%)

Figure 6.1. *MYS compared to OBS for the percentage of customers served*

What calls attention is that, in the first place, we see our hypothesis confirmed in the scenarios we tested it on, looking at the average performance over the replications. When we do not have accurate forecasts (Scenario 1b & Scenario 2b), we see that MYS on average demonstrates a better performance compared to OBS. What is also interesting, is the fact that when we do have accurate historical forecasts (Scenario 1a & Scenario 2a), OBS on average outperforms MYS in terms of the percentage of customers that is served. The margin however is smaller compared to the margin MYS has when it outperforms OBS. There may be several reasons why OBS outperforms MYS. The most likely reason is that MYS tries to accept every customer at any cost. Therefore, it may occur that MYS changes the fleet composition to accept a customer order that is actually not very profitable to accept. OBS cannot change the fleet composition and would reject this customer order. Instead, OBS may be able to accept other customer orders that turn out to be more profitable in the end, where MYS may not be able to accept those orders that are more profitable anymore.

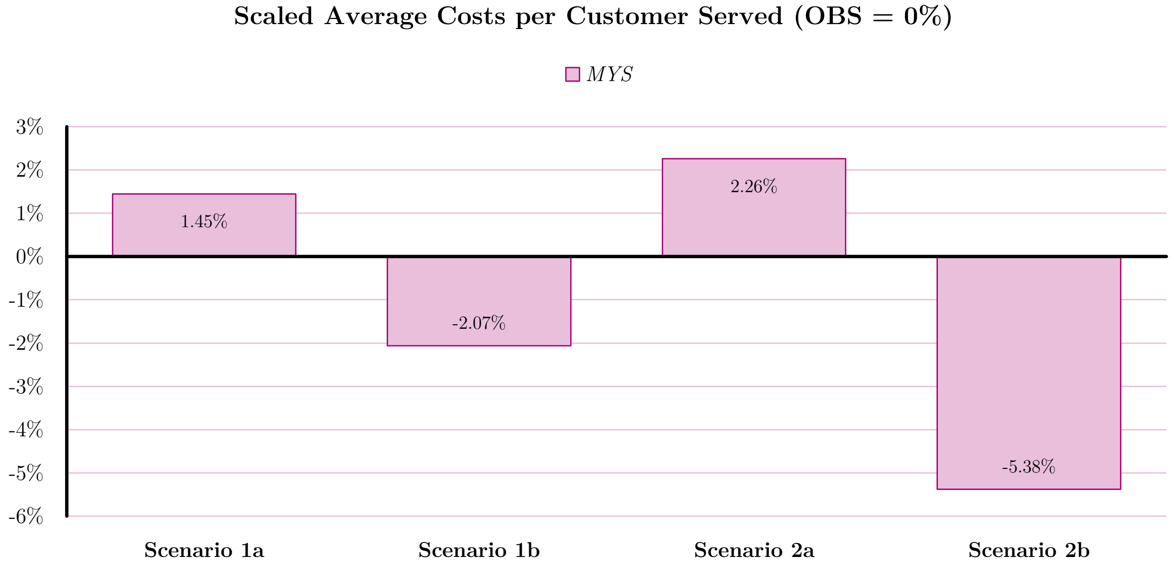


Figure 6.2. *Delivery costs per customer for MYS compared to OBS*

When we look into the average delivery costs per customer, we see that the performance of the strategies for this KPI is congruent with their performance for the average percentage of customers served. This again confirms our hypothesis that in cases where forecasts with regard to the ratio of order types of the customer orders are not very accurate, we can obtain some benefits by making use of a dynamic strategy instead of a static one.

To test whether we have enough statistical evidence to confirm our hypothesis, we apply a paired-t approach to compare the performance for the percentage of customers that is served and costs per customer that is served. The approach we chose is introduced by Law (2015). This method is a suitable statistical method to compare two system configurations in case we have an equal number of observations for the system performance for both configurations. Be aware that all notations we use to describe the approach are introduced by Law (2015) and do not relate to earlier notations used in this thesis.

In our case we have 10 replications for each strategy, so we apply the paired-t approach. We have X_{ij} with $i = MYS, OBS$ and $j = 1, 2, ..., n$ and $n = 10$, j being observations from system configuration (strategy) i. Equation (6.1) shows how the variable for which we construct a 95% confidence interval is calculated. This variable reflects the difference in performance between both strategies.

$$Z_j = X_{MYSj} - X_{OBSj} \tag{6.1}$$

To construct a 95% confidence interval, we compute the sample mean $\mathrm{E}[Z]$ and the sample variance $\mathrm{Var}[Z]$, as well as the t-value from the t-distribution with $n - 1$ degrees of freedom and a probability of

100% − $\alpha/2 = 97.5\%$ with $\alpha = 2$. The 95% confidence interval is given by Equation (6.2), which is a slight adaptation of the formula that Law (2015) uses, but gives the same result.

$$\mathrm{E}[Z] \pm t_{n-1,\,0.\,75} \times \sqrt{\frac{\mathrm{Var}[Z]}{n}} \tag{6.2}$$

In Table 6.1 the results are displayed for the comparison of the performance of MYS and OBS in terms of the percentage of customers served. Note that these are the real results, and not the scaled ones as displayed in the graphs earlier. The sample mean stands for the average percentage point difference between the percentage of customers served for MYS and OBS. We also present the values we used for the other components of Equation (6.2). In Table 6.2 the results are displayed for the comparison of MYS and OBS with regard to the costs per customer served.

Table 6.1. *Statistical comparison of MYS and OBS for the percentage of customers served*

	Scenario 1a	**Scenario 1b**	**Scenario 2a**	**Scenario 2b**
Sample mean	-1.09%	2.71%	-1.55%	5.82%
Sample variance	0.04%	0.23%	0.11%	0.24%
Degrees of freedom (n − 1)	9	9	9	9
t-value	2.26	2.26	2.26	2.26
Confidence interval for difference	**[-2.52%, 0.35%]**	**[-0.69%, 6.11%]**	**[-3.94%, 0.83%]**	**[2.32%, 9.32%]**

Table 6.2. *Statistical comparison of MYS and OBS for the costs per customer served*

	Scenario 1a	**Scenario 1b**	**Scenario 2a**	**Scenario 2b**
Sample mean	€0.31	-€0.44	€0.57	-€1.52
Sample variance	€0.4	€0.43	€0.62	€1.60
Degrees of freedom (n − 1)	9	9	9	9
t-value	2.26	2.26	2.26	2.26
Confidence interval for difference	**[-€0.19, €0.81]**	**[-€0.91, €0.02]**	**[€0.01, €1.14]**	**[-€2.42, -€0.61]**

In Appendix B.1 we report all data points used to calculate the confidence intervals. The results of our statistical comparison indicate that for Scenario 1 for both KPIs we cannot show that MYS or OBS performs better, because all confidence intervals contain 0. However, we see that for Scenario 1b, 0 is almost out of the bounds of the confidence intervals for both KPIs. This indicates that probably if we would do more replications (i.e., the confidence interval gets narrower), we could show that MYS significantly outperforms OBS because 0 would not be in the confidence interval anymore. This is already the case for both KPIs in Scenario 2b. An interesting observation is that for Scenario 2a, with approximately 95% confidence OBS outperforms MYS when it comes to the costs per customer that is served.

Further results from our experiments for Hypothesis 1 can all be found in Appendix B.1. An interesting result to consider is the fact that the response time to a customer request for available time windows for MYS is much larger (three times as large in the most favorable case) compared to the response time to a customer request. This will always be the bottleneck with a dynamic strategy, because changing the fleet composition dynamically requires additional computations, which of course come at the cost of a delay.

Concluding, we see that we have a strong reason to believe that our hypothesis is indeed correct. In cases where forecasts based on historical data are not accurate, MYS tends to outperform OBS, whereas in cases where the forecasts are quite accurate the performance does not seem to be significantly different for both strategies. However, the results call for a better dynamic strategy, because the margins with which MYS outperforms OBS are quite small.

6.2. Hypothesis 2

> *In applications of Use Case 2 and Use Case 4, the performance of the myopic dynamic strategy mentioned in the first hypothesis can be improved by introducing a smart dynamic strategy.*

For this hypothesis we split the results up according to the two use cases mentioned in the statement. For both Use Case 2 (Section 6.2.1) and Use Case 4 (Section 6.2.2) we compare the performance of BAS and MYS, to verify whether the hypothesis indeed can be considered as a valid one. Our aim when putting this hypothesis to a test is to show that by introducing smarter decision mechanisms we can improve the dynamic strategy that we designed for testing the first hypothesis (MYS).

6.2.1. Use Case 2

In Section 6.1 we already showed that MYS tends to outperform OBS for Use Case 2. Here we show that we can outperform both MYS and OBS by introducing BAS, our smart dynamic strategy. Figure 6.3 already shows that on average BAS outperforms MYS and OBS by far. In Figure 6.4 we see that also for the costs per customer that is served BAS outperforms both MYS and OBS with large margins.

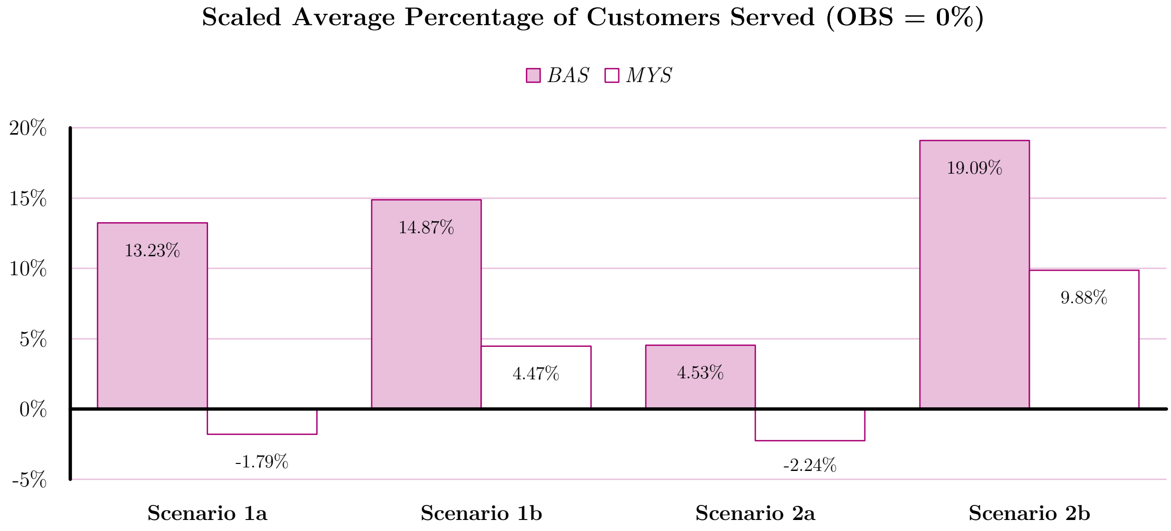


Figure 6.3. *BAS and MYS compared to OBS for the percentage of customers served*

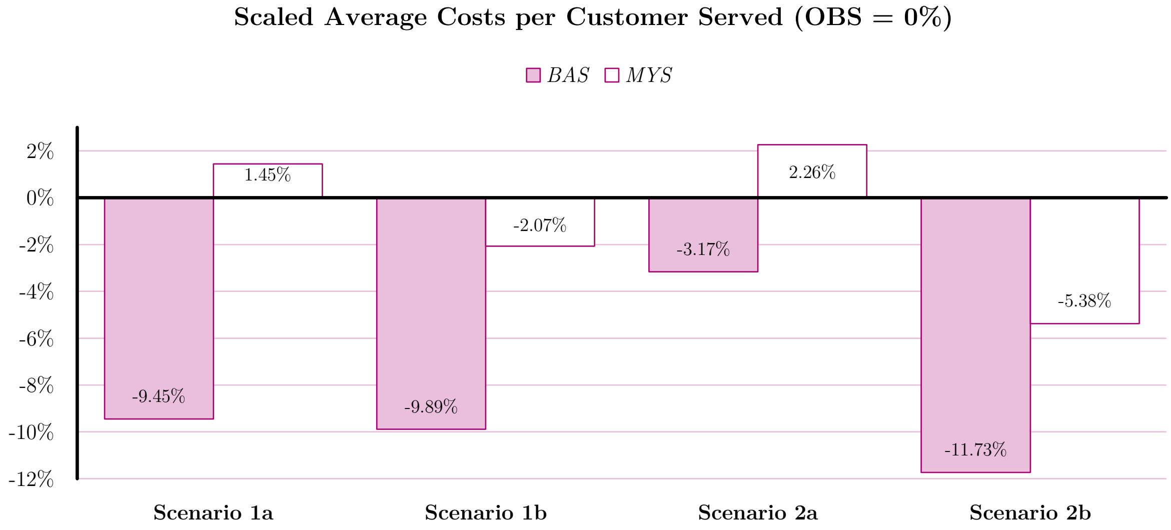


Figure 6.4. *Delivery costs per customer for BAS and MYS compared to OBS*

On average over all scenarios, BAS relatively scores 12.93% higher with regard to the average percentage of customers served than OBS under the same conditions. The average margin with which BAS outperforms OBS with regard to the average costs per customer served equals 8.56%. The latter margin is not as large as the margin for the average percentage of customers served, but is still large enough to state that BAS can be considered to deliver a much better performance than OBS.

An interesting observation is that the performance of BAS differs much more between situation *a* and situation *b* for Scenario 2, compared to Scenario 1. A possible explanation of this phenomenon may be the influence of the spread of customer locations. In the first place, OBS suffers in situation *b* from the impact caused by the fact that the fleet composition formed with OBS does not have enough capacity for Order Type A. This is the case for both Scenario 1 and Scenario 2, and therefore BAS outperforms OBS for all possible Scenarios. However, we just mentioned that the difference for Scenario 2b is relatively large. This is probably due to the fact that, besides the impact of the bad fleet composition, OBS suffers also from the fact that the driving times between customer locations are larger for Scenario 2b. Due to this fact, OBS does not only have capacity problems to serve customers with Order Type A, but OBS also can serve less customers because more time is lost with driving between locations. This explanation is backed up by the fact that the percentage of customers served by OBS deteriorates more from Scenario 2a to Scenario 2b compared to from Scenario 1a to Scenario 1b (see Table B.4 and Table B.5 in Appendix B.2.1). Similarly, we observe a much larger difference in utilization for OBS in Scenario 2 than in Scenario 1. This observation indicates that indeed the driving time may be a bottleneck for OBS in Scenario 2b.

Besides this explanation, we also observe that the deviation in the ratio of order types compared to forecasts based on historical data is larger for Scenario 2b compared to 1b. This increases the negative impact of the bad initial fleet composition that OBS cannot change.

Given the large margins that BAS has compared to MYS in terms of performance for both KPIs, there seems to be hardly any need to compare the strategies with a paired-*t* approach. But for the sake of completeness, we compute the 95% confidence intervals for the difference between BAS and MYS, as we already showed earlier that MYS tends to outperform OBS. The approach we follow is already described in Section 6.1. The difference between BAS and MYS is computed with Equation (6.3). The 95% confidence intervals are again computed with Equation (6.2).

$$Z_j = X_{BASj} - X_{MYSj} \tag{6.3}$$

In Table 6.3 and Table 6.4 the 95% confidence intervals are displayed for respectively the percentage of customers served and the costs per customer served. Note that for the percentage of customers served the sample mean again considers a percentage point difference between the performance of BAS and MYS.

Table 6.3. *Statistical comparison of BAS and MYS for the percentage of customers served*

	Scenario 1a	**Scenario 1b**	**Scenario 2a**	**Scenario 2b**
Sample mean	9.12%	6.30%	4.69%	5.44%
Sample variance	0.06%	0.13%	0.13%	0.13%
Degrees of freedom (n − 1)	9	9	9	9
t-value	2.26	2.26	2.26	2.26
Confidence interval for difference	[**7.42%, 10.83%**]	[**3.76%, 8.85%**]	[**2.15%, 7.24%**]	[**2.84%, 8.03%**]

Table 6.4. *Statistical comparison of BAS and MYS for the costs per customer served*

	Scenario 1a	**Scenario 1b**	**Scenario 2a**	**Scenario 2b**
Sample mean	-€2.33	-€1.6	-€1.37	-€1.79
Sample variance	€0.70	€0. 7	€0.72	€2.40
Degrees of freedom (n − 1)	9	9	9	9
t-value	2.26	2.26	2.26	2.26
Confidence interval for difference	[**-€2.93, -€1.73**]	[**-€2.39, -€0.98**]	[**-€1.98, -€0.77**]	[**-€2.90, -€0.69**]

As we already would expect from the averages, we see indeed that BAS performs much better than MYS for Use Case 2 according to the 95% confidence intervals. This gives us enough statistical evidence to

assume that our hypothesis is indeed valid for Use Case 2. In the next section we look at the difference in performance between BAS and MYS for Use Case 4.
From the remaining results as given in Appendix B.2.1 we observe a few interesting things. First, we see that for all scenarios BAS on average offers more time windows to a customer compared to MYS. Also, the average response time to a request for available time windows for BAS is always less than the response time for MYS. So, from a customer satisfaction point of view, BAS tends to outperform MYS for the scenarios we analyzed.

We see also that BAS has a lower average utilization of the vehicles compared to MYS for all scenarios. This indicates that we may have space in terms of fleet capacity to realize even more improvements. However, a lower utilization does not necessarily mean that we have space for improvements. It can also just be a reflection of the impact of the decisions our strategy takes. In this case for instance, a partial explanation for the lower utilization is that BAS rejects customers that are unattractive based on their large order quantities. This leads to accepting relatively more customer orders with smaller order quantities, that occupy less of the fleet capacity. Therefore, the average utilization of the vehicles is lower. Another aspect that may have a large impact on the average utilization is the tuning of the parameters for BAS. It may be interesting to investigate if we can tune the parameters differently, such that we are able to serve more customers by making use of a larger part of the fleet capacity.

6.2.2. Use Case 4

In Use Case 4, besides the effect of a different ratio of order types compared to the expectations, we consider the possibility to increase or decrease the fleet capacity for the current fleet composition. We may in this use case re-assign drivers from vehicles with less capacity to vehicles with more capacity and vice versa. We again compare BAS and MYS to each other, because both outperform OBS by far if we start with the same initial fleet composition for all strategies. We see that for both the average percentage of customers served (Figure 6.5) and the costs per customer served (Figure 6.6) the differences between BAS and MYS are smaller than for Use Case 2, for all scenarios. However, for all scenarios on average BAS still outperforms MYS. The cause of the fact that the difference between BAS and MYS in performance is smaller for Use Case 4 lays most likely in the fact that we have a higher maximum fleet capacity. This reduces the negative impact when MYS accepts an unattractive customer with a large order quantity, where BAS would have rejected that customer. As we have more fleet capacity available, MYS has more flexibility to still serve future customer orders in such cases for Use Case 4, compared to Use Case 2.

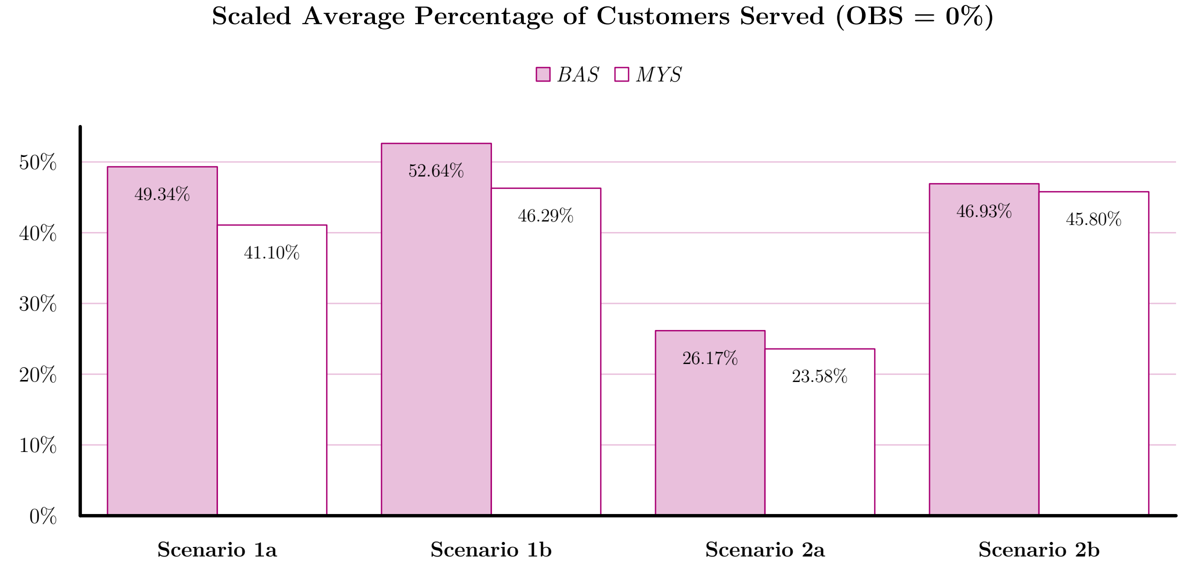


Figure 6.5. *BAS and MYS compared to OBS for the percentage of customers served*

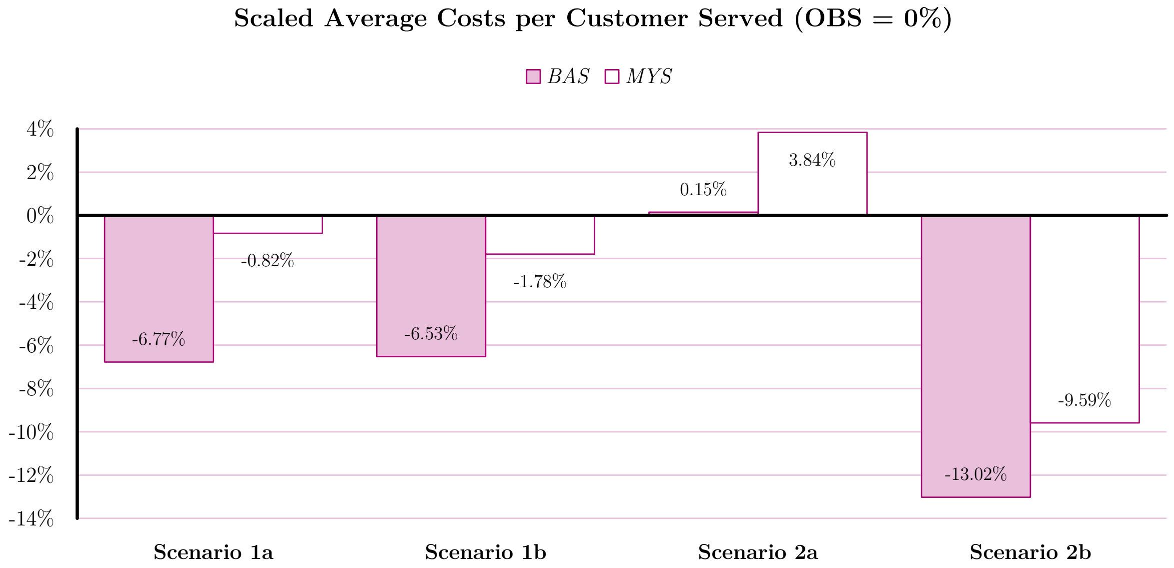


Figure 6.6. *Delivery costs per customer for BAS and MYS compared to OBS*

What calls attention with regard to the performance of BAS and MYS for the average costs per customer served, is the fact that for Scenario 2a both strategies perform slightly worse compared to OBS. This is due to the fact that, although OBS was able to serve fewer customers than the other two strategies, OBS managed to construct very efficient delivery routes in terms of delivery costs. Therefore, the costs per customer that is served are also low for OBS. However, the percentage of customers that is served for OBS in Scenario 2a is so low, that we cannot state that OBS shows a good performance for this scenario.

We again compare the performance for both strategies with a paired-t approach. The difference between the performance of BAS and MYS is computed with Equation (6.3) and the 95% confidence intervals are determined by using Equation (6.2). Note that the difference is again given in percentage points. The performance of both strategies for each replication can be found in Table B.10 in Appendix B.2.2. Also be aware that for the comparison of BAS and MYS we do not use the scaled performance of both strategies compared to OBS, but the real performance. The results of the comparison are displayed in Table 6.5 and Table 6.6.

Table 6.5. *Statistical comparison of BAS and MYS for the percentage of customers served*

	Scenario 1a	**Scenario 1b**	**Scenario 2a**	**Scenario 2b**
Sample mean	5.00%	3.85%	1.79%	0.66%
Sample variance	0.09%	0.29%	0.13%	0.17%
Degrees of freedom (n – 1)	9	9	9	9
t-value	2.26	2.26	2.26	2.26
Confidence interval for difference	**[2.88%, 7.12%]**	**[-0.01%, 7.71%]**	**[-0.78%, 4.36%]**	**[-2.31%, 3.64%]**

Table 6.6. *Statistical comparison of BAS and MYS for the costs per customer served*

	Scenario 1a	**Scenario 1b**	**Scenario 2a**	**Scenario 2b**
Sample mean	-€1.27	-€1.02	-€0. 3	-€0. 7
Sample variance	€0.33	€1.11	€1. 6	€2.36
Degrees of freedom (n – 1)	9	9	9	9
t-value	2.26	2.26	2.26	2.26
Confidence interval for difference	**[-€1.68, -€0.86]**	**[-€1.77, -€0.27]**	**[-€1.93, €0.07]**	**[-€2.07, €0.13]**

We see that for the percentage of customers that is served, only for Scenario 1a we can say with approximately 95% confidence that BAS outperforms MYS. However, for both Scenario 1b and Scenario 2a, 0 is almost out of the bounds of the 95% confidence interval. This indicates that if we would use more replications or reduce our confidence level a little, we would have enough statistical evidence to claim that BAS outperforms MYS also for these scenarios.

An interesting observation is that both for the percentage of customers that is served and for the costs per customer served BAS tends to perform better than MYS in Scenario 1 compared to Scenario 2. This may be due to the same phenomenon we discussed earlier in Section 6.2.1, namely the fact that the spread of the locations in Scenario 2 is quite different from Scenario 1. The average travel times from the depot to the customers are higher for Scenario 2, which may cause the time constraints to have more impact on whether a customer can be accepted or not, compared to the capacity constraints. As BAS in our experiments rejects customers when they are unattractive based on the order quantity and not the additional driving time, this may be an explanation for the better performance of BAS compared to MYS for Scenario 1.

Another important observation is that for all scenarios both BAS and MYS show a much better performance than OBS, so for sure there is a clear benefit in using a dynamic strategy when we start with the same initial fleet composition for all strategies. The fact that we use the same initial fleet composition is also the reason that BAS and MYS have such high scores for the scaled (based on OBS' performance) average percentage of customers served. In Section 6.3, where we validate Hypothesis 3, we consider OBS with a different initial fleet composition, to obtain insights in how BAS performs compared to that strategy.

From the results in Appendix B.2.2, we observe that BAS again on average offers more time windows to a customer than all other strategies and the average response time to a customer is always lower compared to MYS. We also observe that the vehicle utilization for BAS, especially for Scenario 2, is quite low compared to the other strategies. This supports the explanation above for the smaller difference in performance between BAS and MYS for Scenario 2. It seems that BAS on average either rejected too many customer orders, or that BAS on average accepted too many customer orders that increased the driving time a lot. A solution for this problem may be reconsidering the tuning of the parameters for BAS and the criteria that BAS uses to reject a customer. It may in cases similar to Scenario 2 be more beneficial to for instance reject customer orders based on the customer location instead of the order quantities. Customers that order from locations that require a lot of time to reach, are then considered as unattractive instead of customers with large order quantities. This may be an interesting topic for further research.

Summarizing, we can say that on average BAS outperforms MYS, but for some cases we do not have enough statistical evidence to state with confidence that BAS always outperforms MYS in terms of the percentage of customers served and the costs per customer served. However, the other effects that we observe and the fact that on average BAS performs better than MYS in all cases, give us enough reason to state that BAS is a better strategy compared to MYS. These observations confirm our second hypothesis for both Use Case 2 and Use Case 4.

6.3. Hypothesis 3

> *In applications of Use Case 4, we achieve a better performance when we use a smart dynamic strategy with an initial fleet composition that uses vehicles that are as cheap as possible, compared to a static strategy that starts with a composition that has a total fleet capacity that is as large as possible and is therefore costlier.*

As explained in Chapter 3, this hypothesis becomes relevant due to the assumption we make that on average more customer orders come in than can be served with the maximum capacity. The question raises then why we should implement a time-consuming dynamic strategy, if a static strategy potentially also may perform a good job. In Figure 6.7 we see our hypothesis confirmed for all scenarios in terms of the average percentage of customers that is served. This is, especially for Scenario 1a and Scenario 2a, mainly due to the decision mechanisms implemented for BAS with regard to rejecting customers that are not considered to be attractive enough.

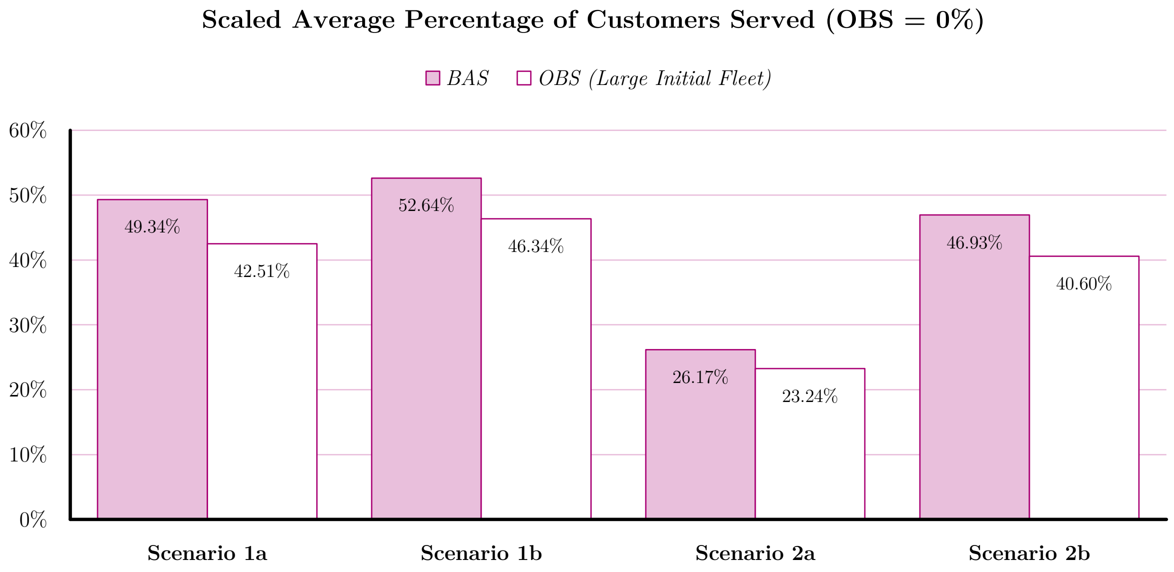


Figure 6.7. *BAS and OBS (Large Initial Fleet) compared to OBS for the percentage of customers served*

When we look at the costs per customer that is served, we see that BAS on average outperforms OBS (Large Initial Fleet) for most of the scenarios we tested. The results are shown in Figure 6.8. Only for Scenario 2a we see that BAS does not outperform OBS (Large Initial Fleet) nor the standard OBS, but the differences are small.

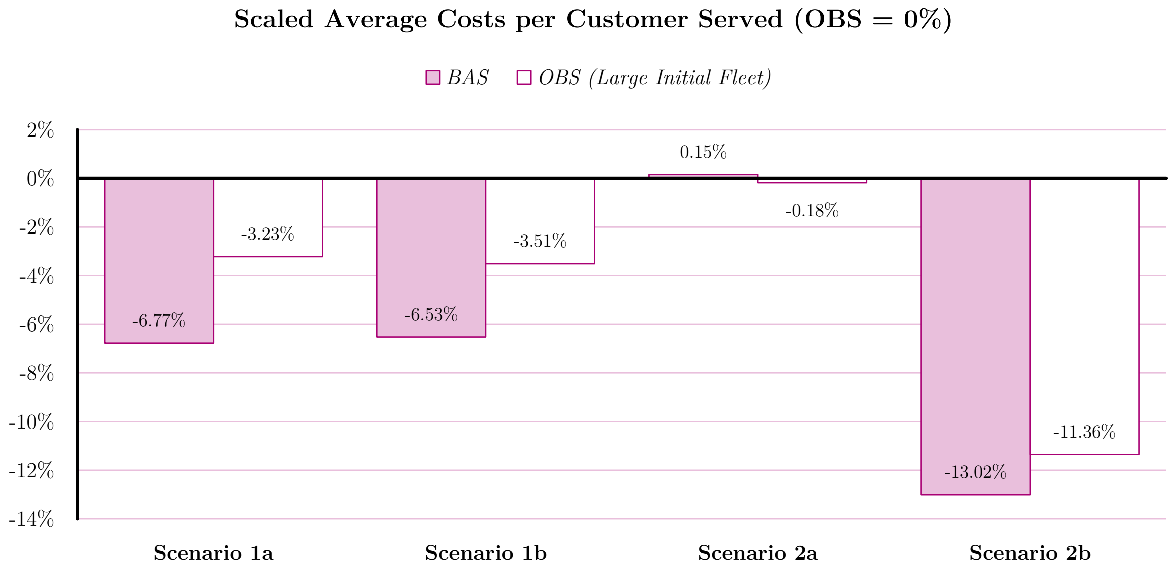


Figure 6.8. *Delivery costs per customer for BAS and OBS (Large Initial Fleet) compared to OBS*

To check whether we indeed have enough statistical evidence to say that BAS outperforms OBS (Large Initial Fleet), we follow the same paired-t approach as described in Section 6.1. The difference between the two strategies is now given by Equation (6.4). The 95% confidence intervals are calculated by making use of Equation (6.2), just as before.

$$Z_j = X_{BASj} - X_{OBS(Large\ Initial\ Fleet)j} \tag{6.4}$$

In Table 6.7 we see the resulting 95% confidence intervals for the percentage point difference in performance of BAS and OBS (Large Initial Fleet) for the percentage of customers served. Table 6.8 shows the 95% confidence intervals for the difference between the two strategies with regard to the costs per customer that is served.

Table 6.7. *Statistical comparison of BAS and OBS (Large Initial Fleet) for the percentage of customers served*

	Scenario 1a	**Scenario 1b**	**Scenario 2a**	**Scenario 2b**
Sample mean	4.15%	3.82%	2.03%	3.74%
Sample variance	0.16%	0.19%	0.06%	0.18%
Degrees of freedom (n − 1)	9	9	9	9
t-value	2.26	2.26	2.26	2.26
Confidence interval for difference	**[1.32%, 6.97%]**	**[0.67%, 6.97%]**	**[0.24%, 3.82%]**	**[0.71%, 6.77%]**

Table 6.8. *Statistical comparison of BAS and OBS (Large Initial Fleet) for the costs per customer served*

	Scenario 1a	**Scenario 1b**	**Scenario 2a**	**Scenario 2b**
Sample mean	-€0.76	-€0.65	€0.0	-€0.47
Sample variance	€0.45	€0.57	€0.67	€3.0
Degrees of freedom (n − 1)	9	9	9	9
t-value	2.26	2.26	2.26	2.26
Confidence interval for difference	**[-€1.24, -€0.28]**	**[-€1.19, -€0.11]**	**[-€0.50, €0.67]**	**[-€1.72, €0.79]**

We see that for the percentage of customers that is served, we can say with approximately 95% confidence that BAS outperforms OBS (Large Initial Fleet) for all scenarios that we simulated. So, as we use this measure to quantify the customer satisfaction, we are confident to say that with regard to customer satisfaction our hypothesis cannot be rejected.

When it comes to the costs per customer served, we see that for Scenario 1a and Scenario 1b we have enough statistical evidence to say with approximately 95% confidence that BAS outperforms OBS (Large Initial Fleet). For Scenario 2a and Scenario 2b we cannot show that there is a significant difference between the two strategies, with 95% confidence.

However, we can show a significant difference for Scenario 1a and Scenario 1b with regard to the costs per customer that is served. The same holds for all scenarios in terms of the percentage of customers that is served. Therefore, we are confident to say that the results give us enough reason to believe Hypothesis 3 is valid. This leads us to the conclusion that in practical applications similar to the scenarios we simulated, it is worthwhile to implement a smart dynamic strategy instead of a static strategy with an initial fleet composition with maximum fleet capacity.

6.4. Hypothesis 4

> *In applications of Use Case 4, a smart dynamic strategy achieves a stable performance for different distribution patterns of customer arrivals over the booking period, as long as we have accurate forecasts of this spread. However, when forecasts are not accurate, the performance of a smart dynamic strategy may deteriorate significantly.*

For our last hypothesis we again split up the results into different sections. In Section 6.4.1 we compare the impact of changing our empirical pattern of customer arrival times to a different pattern in which customer arrival rates increase during the whole booking period. In Section 6.4.2 we give insight into what the impact is of consistently under-forecasting or over-forecasting the cumulative percentage of customers that has arrived at a given point of time. Finally, in Section 6.4.3 we study the impact of having fluctuating forecast errors with regard to the cumulative percentage of customers that has arrived at a certain point of time. Our main goal in this section is giving insights into the robustness of BAS when expectations turn out not to match observations in practice.

6.4.1. Comparison of Different Arrival Time Distributions

Until here, we only studied cases in which the customer arrival times were distributed according to an empirical distribution, as explained in Chapter 5. In this section we study the impact of having an increasing arrival rate over the booking period of one day (as also explained in Chapter 5) on the results for BAS for Use Case 4 in all scenarios. The purpose of this comparison is not pointing out for which distribution BAS shows a better performance, but we want to obtain insights into the stability of the performance of BAS for different distributions of customer arrival times. Figure 6.9 indicates that the differences in performance for both distributions are small when it comes to the average percentage of customers served. Figure 6.10 shows the same observations with regard to the costs per customer that is served. Only for Scenario 1a we see that the difference is somewhat larger. Note that for OBS the distribution of customer arrival times does not make any difference. The only thing that matters for the performance of OBS is the sequence in which customers arrive. We used the same sequence for all strategies that we consider.

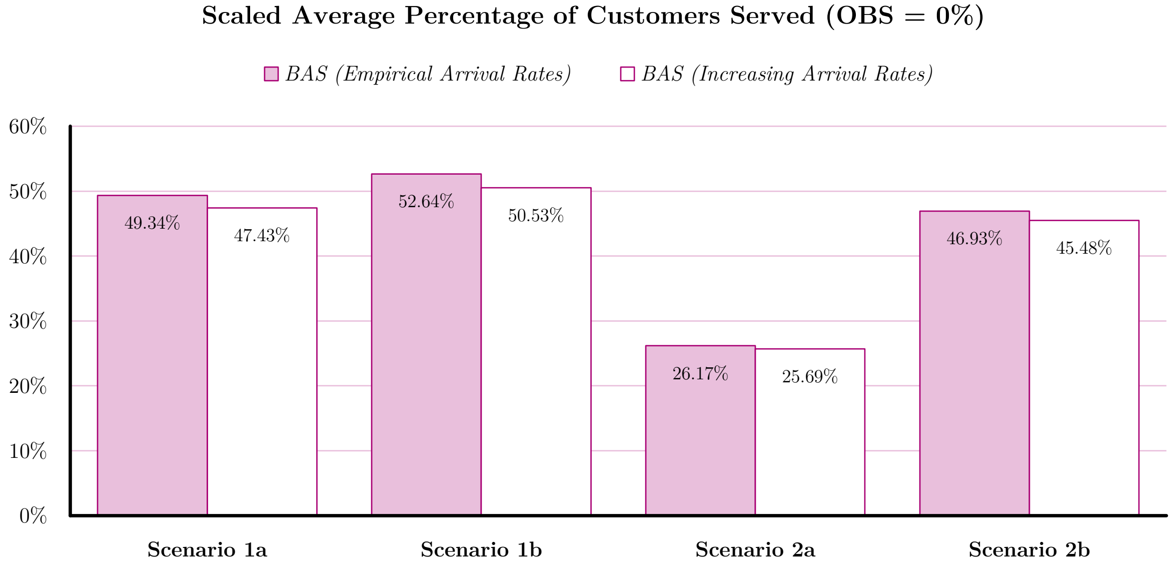


Figure 6.9. *BAS (Empirical) and BAS (Increasing) compared to OBS for the percentage of customers served*

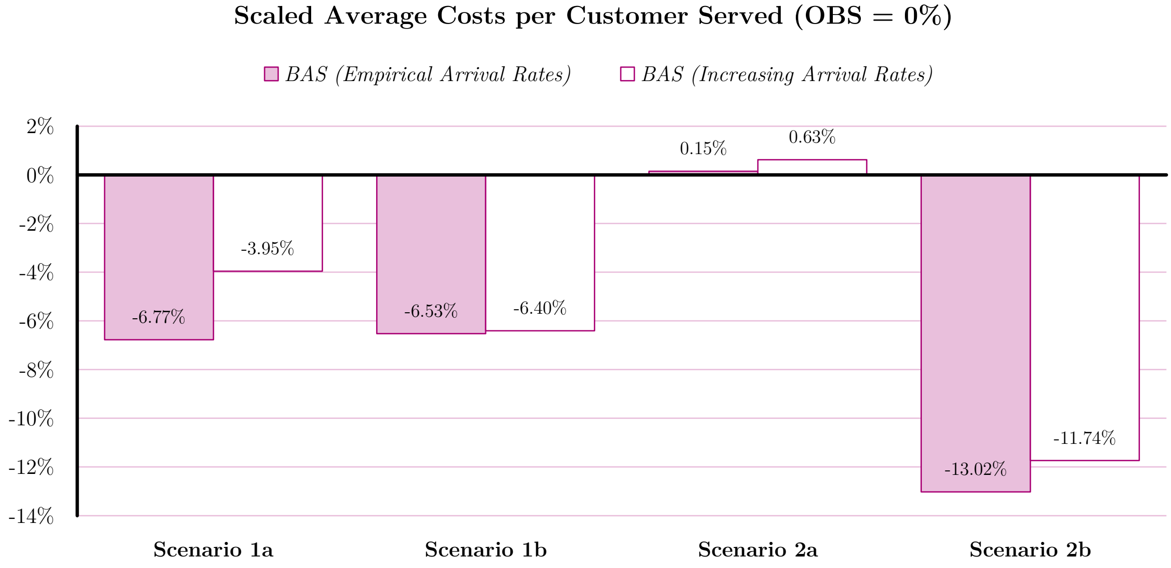


Figure 6.10. *Delivery costs per customer for BAS (Empirical) and BAS (Increasing) compared to OBS*

We now want to know if we indeed have enough statistical evidence to state that the performance of BAS is not significantly different when the distribution of customer arrival times changes, provided that we can accurately forecast this distribution. To that end, we again use a paired-t approach to compare the performance of BAS in both cases. The difference between the two cases is now given by Equation (6.5). The 95% confidence intervals are again computed with Equation (6.2), just as before.

$$Z_j = X_{BAS(Empirical)j} - X_{BAS(Increasing)j} \tag{6.5}$$

In Table 6.9 we see the resulting 95% confidence intervals for the percentage point difference in performance between BAS with an empirical distribution of arrival times and BAS with an increasing arrival rate over the booking period, with regard to the percentage of customers served. Table 6.10 shows the 95% confidence intervals for the difference between the two cases when it comes to the costs per customer that is served.

Table 6.9. *Statistical comparison of BAS (Empirical) and BAS (Increasing) for percentage of customers served*

	Scenario 1a	**Scenario 1b**	**Scenario 2a**	**Scenario 2b**
Sample mean	1.16%	1.28%	0.33%	0.86%
Sample variance	0.03%	0.05%	0.03%	0.12%
Degrees of freedom (n – 1)	9	9	9	9
t-value	2.26	2.26	2.26	2.26
Confidence interval for difference	**[-0.16%, 2.48%]**	**[-0.30%, 2.86%]**	**[-1.01%, 1.66%]**	**[-1.57%, 3.28%]**

Table 6.10. *Statistical comparison of BAS (Empirical) and BAS (Increasing) for costs per customer served*

	Scenario 1a	**Scenario 1b**	**Scenario 2a**	**Scenario 2b**
Sample mean	-€0.60	-€0.03	-€0.12	-€0.36
Sample variance	€0.40	€0.30	€0.37	€1.50
Degrees of freedom (n – 1)	9	9	9	9
t-value	2.26	2.26	2.26	2.26
Confidence interval for difference	**[-€1.06, -€0.15]**	**[-€0.42, €0.37]**	**[-€0.56, €0.32]**	**[-€1.24, €0.52]**

From the 95% confidence intervals that we computed we can conclude that for both KPIs for almost all scenarios we are not able to show a significant difference in performance for the different arrival time distributions. Only for Scenario 1a we find that BAS' performance slightly deteriorates in terms of the costs per customer served when arrival times are distributed with increasing arrival rates.

Nevertheless, we are confident to say that, if we have accurate forecasts of the arrival time distribution, the performance of BAS does not change significantly when this distribution changes, as Hypothesis 4 states. So, for cases that are similar to the scenarios that we simulated, we conclude that the performance of BAS is stable, regardless of the distribution of customer arrival times. However, we do not know yet what happens to BAS' performance when we are not able to accurately forecast the arrival time distribution. Therefore, in Section 6.4.2 and Section 6.4.3 we study the impact on the performance of BAS when we do not have accurate forecasts.

6.4.2. Impact of Fixed Forecast Errors

In this section, we study the impact of fixed forecast errors for the cumulative percentage of customers arrived at a given point in time on the results of BAS for Scenario 1b and Use Case 4, as described in Chapter 5. We may not always be able to accurately forecast this measure, and it is important to know what happens in such cases. Figure 6.11 and Figure 6.12 display the results for the average percentage of customers that is served and the average costs per customer that is served. Instead of taking the performance of OBS as a base, we now take the performance of BAS with no standard forecast error as a base for the performance of BAS with different forecast error values.

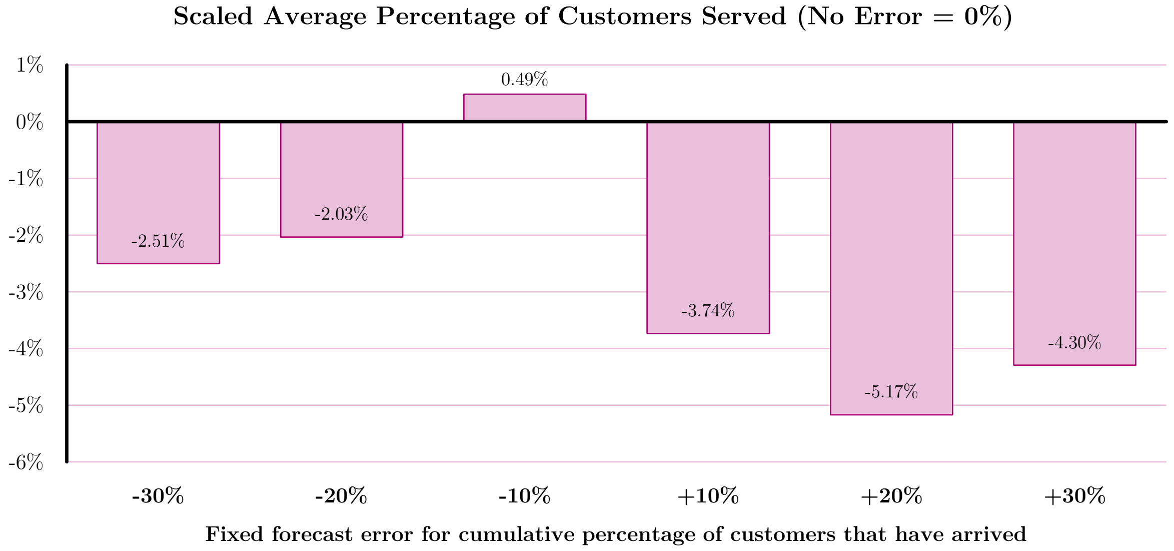


Figure 6.11. *Impact of different fixed forecast error values on percentage of customers served (BAS, Scenario 1b)*

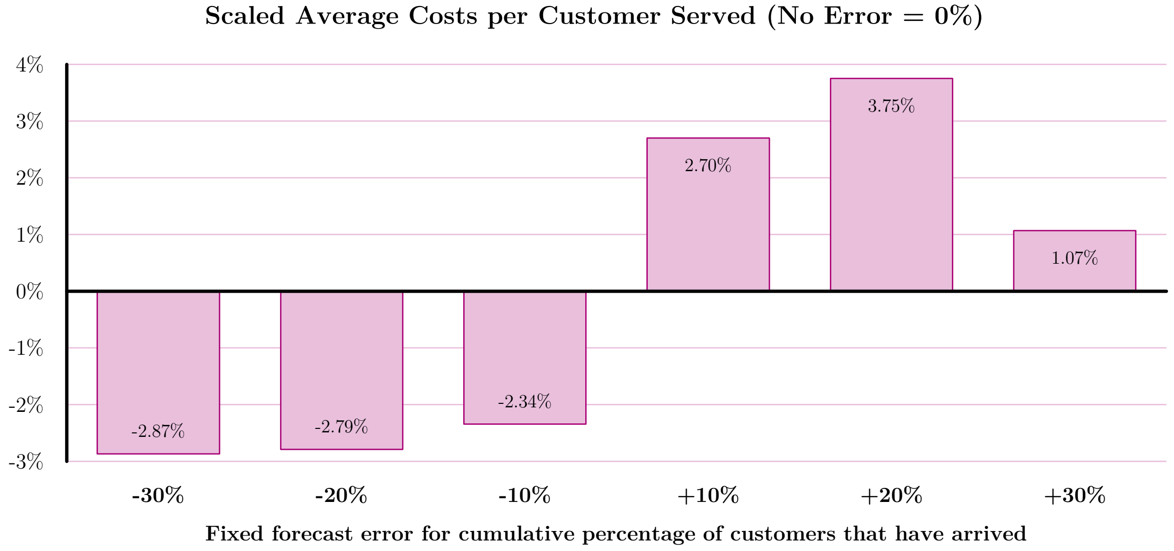


Figure 6.12. *Impact of different forecast error values on the delivery costs per customer (BAS, Scenario 1b)*

From the average results we see that apparently over-forecasting gives worse results compared to under-forecasting the cumulative percentage of customers arrived at a given point in the booking period in this scenario. This is most likely due to the fact that when our forecasts are too high, BAS only increases the fleet capacity after (unnecessarily) rejecting a few customers in case we are not using the maximum capacity of the fleet yet. In case our forecasts are too low, the only thing that happens is that we assume to require more fleet capacity than we actually need. This does not really seem to be a problem, as in our context on average enough customers arrive to fill up the maximum capacity of the delivery fleet.

An interesting observation from the results is the fact that for a fixed forecast error of -10%, we see that the performance for both the average percentage of customers served and the average costs per customer served is better compared to the performance of BAS without forecast errors. This is again a strong indication that the tuning of the parameters for BAS is very important. If we would have changed some thresholds for the cumulative percentage of customers arrived with 10%, we would have obtained better results for BAS in this scenario. This observation therefore emphasizes the need for further research and analysis with regard to the parameter tuning process when ORTEC would consider implementing BAS in its software solutions.

Although we observe significant differences in the results (see Appendix B.4.2) when we have forecast errors, the differences are relatively small. Also, for most of the forecast error values BAS on average still outperforms MYS, OBS and OBS (Large Initial Fleet) and in the cases where we do not outperform them (+20% and +30%) the margins are relatively small.

This gives us enough reason to believe that, although it is a simple strategy, BAS has potential to perform quite well even in cases where forecasts are not very accurate. Moreover, in practice it seems a plausible assumption that e-retailers from the context that we study are capable of producing forecasts with small forecast errors. In many applications of the e-retailer case customers tend to have patterns for the time of the booking period in which they place their orders. E-retailers nowadays have access to lots of data and can use historical data to find those patterns. If for some reason during a certain booking period the distribution follows a completely different pattern, BAS may show a worse performance for that case. We argue however that what really matters in our application context is that on average we have a good and stable performance. As long as the average performance is good, it does not necessarily matter that we have some outliers with a worse performance, as we saw for Scenario 1b.

6.4.3. Impact of Fluctuating Forecast Errors

In this section, we study the impact of having fluctuating forecast errors for the expected cumulative percentage of customers that arrived by a given time. The purpose of investigating this impact is to get an idea how sensitive the performance of BAS is to sometimes over-estimating the hourly arrival rate and other times under-estimating this rate. As mentioned in Chapter 5, we focus on Scenario 1b and Use Case 4. We carry out two experiments. In the first one, we determine our forecasts based on the uniform distribution for the customer arrival times, whereas the customers actually arrive according to the empirical distribution. In the second experiment we exchange the distributions, the forecasted one is now the empirical distribution and the real one the uniform distribution.

To clarify the effect of the fluctuating forecast errors during the booking period, we take a look at the hourly arrival rate. Figure 6.13 gives insight into how the forecast errors for the hourly arrival rate (expected total number of customers = 100) develop over the booking period of one day. We see that for the first experiment, in the first hours of the booking period we are mostly over-forecasting the arrival rate. For the later hours, we are mostly under-forecasting the arrival rate. For the second experiment exactly the contrary is the case.

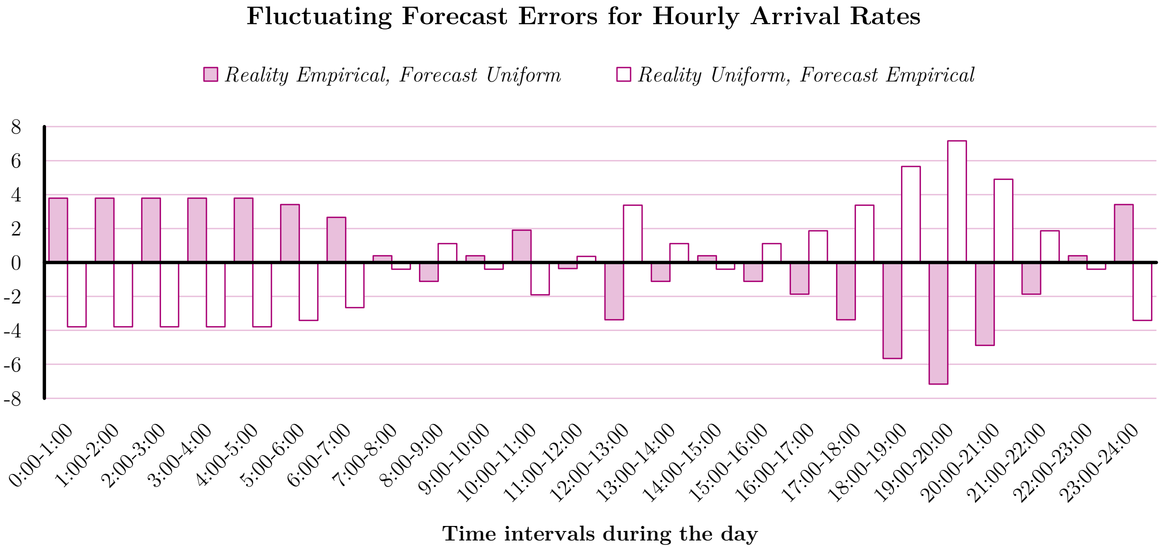


Figure 6.13. *Fluctuating forecast error values for the hourly arrival rates for both experiments (BAS, Scenario 1b)*

On average, the forecast errors are equal to 0. However, the forecast errors show a lot of variance. We now have a better idea of what the variance in our forecasts, that we deliberately created, looks like. Note that there is a direct relation between the cumulative percentage of customers arrived and the hourly arrival rate. If we divide the hourly arrival rate by the expected total number of customers for each hour of the booking period, and we aggregate the results over the hours, we obtain the cumulative distribution function that gives us the cumulative percentage of customer arrived by a certain time. The forecast error for the cumulative percentage of customers arrived can be obtained from Figure 5.12. Figure 6.14 and Figure 6.15 display the results for the average percentage of customers that is served and the average costs per customer that is served. Instead of taking the performance of OBS as a base, we again take the performance of BAS with no forecast error as a base, to compare the results of our experiments to.

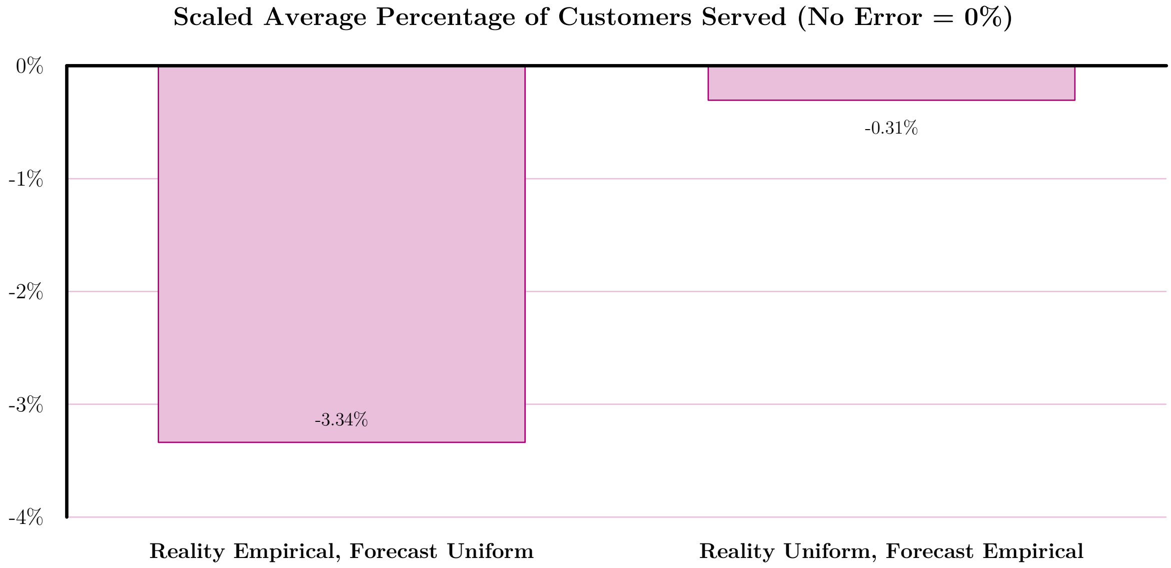


Figure 6.14. *Impact of fluctuating forecast error values on percentage of customers served (BAS, Scenario 1b)*

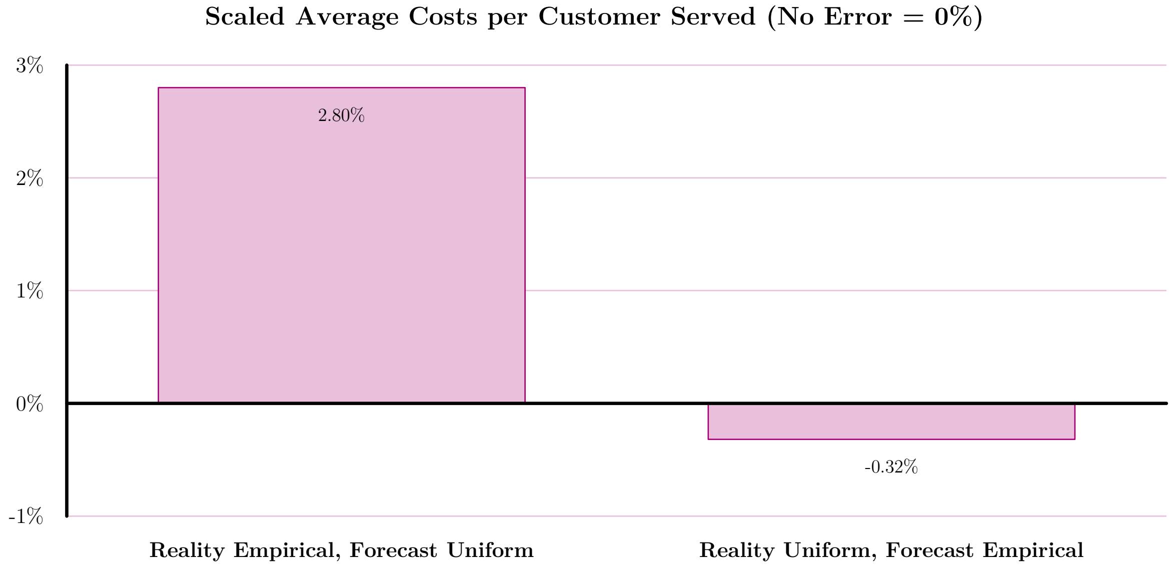


Figure 6.15. *Impact of fluctuating forecast error values on delivery costs per customer (BAS, Scenario 1b)*

What immediately calls our attention is the fact that for the second experiment, the forecast errors hardly have any impact. For the first experiment, we do see that the performance of BAS significantly deteriorates due to the forecast errors. This is also shown by the 95% confidence intervals as given in Appendix B.4.3. If we take a closer look into the results, we see that the results of our two experiments are actually in line with our results regarding the fixed forecast errors. We stated in Section 6.4.2 that the performance of BAS is more sensitive to over-forecasting than to under-forecasting of the cumulative percentage of customers arrived. If we look at Figure 5.12, we see that for our first experiment we are over-forecasting most of the time, because we assume a uniform distribution where the real distribution is an empirical one. Analogously, for our second experiment we are under-forecasting most of the time.

The reason that BAS's performance is sensitive to over-forecasting the cumulative percentage of customers arrived, is that BAS the wrong forecast prevents BAS from increasing the fleet capacity in time. In situations of over-forecasting, most of the times the reason for increasing the fleet capacity will be that more customers were rejected than the configured threshold (p_4) instead of detecting that the utilization is too high or that more customers have arrived than expected.

The results we obtained again warn us that it is important to further develop BAS in such a way that the strategy can deal better with forecast errors, especially in situations of over-forecasting the cumulative percentage of customers arrived by a certain time. One way to improve the performance of BAS is to give more attention to the tuning of its parameters.

We see that our observations in this section confirm our hypothesis. We can indeed achieve a stable performance in different situations, as long as our forecasts are accurate. However, when we have to deal with inaccurate forecasts there is a danger of the performance to deteriorate significantly.

6.5. General Results

In this section we highlight some observations from our results that cannot be directly categorized under one of our hypotheses, but that provide interesting insights. We first perform a benchmark to evaluate how much the strategies we designed can still be improved for the different scenarios we consider (Section 6.5.1). After that, we present some insights with regard to the impact of intermediate optimization calls on the total delivery costs for our scenarios (Section 6.5.2).

6.5.1. Prophet Benchmark

To get an idea of how good our strategies actually perform and if we could do better, we benchmark them against the two prophet strategies that we introduced in Chapter 5. Below we present the results of our comparisons of the different strategies in terms of the average percentage of customers that is served for Use Case 2 (Figure 6.16) and Use Case 4 (Figure 6.18). We do the same for the results of our comparisons for the average costs per customer that is served for Use Case 2 (Figure 6.17) and Use Case 4 (Figure 6.19).

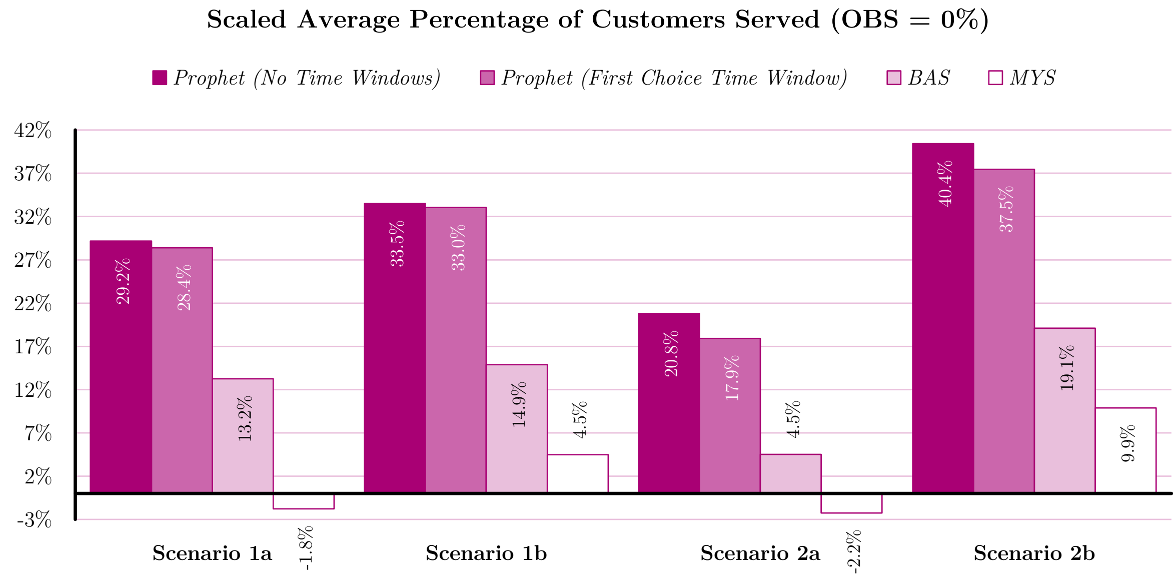


Figure 6.16. *Benchmark of different strategies for the average percentage of customers served (Use Case 2)*

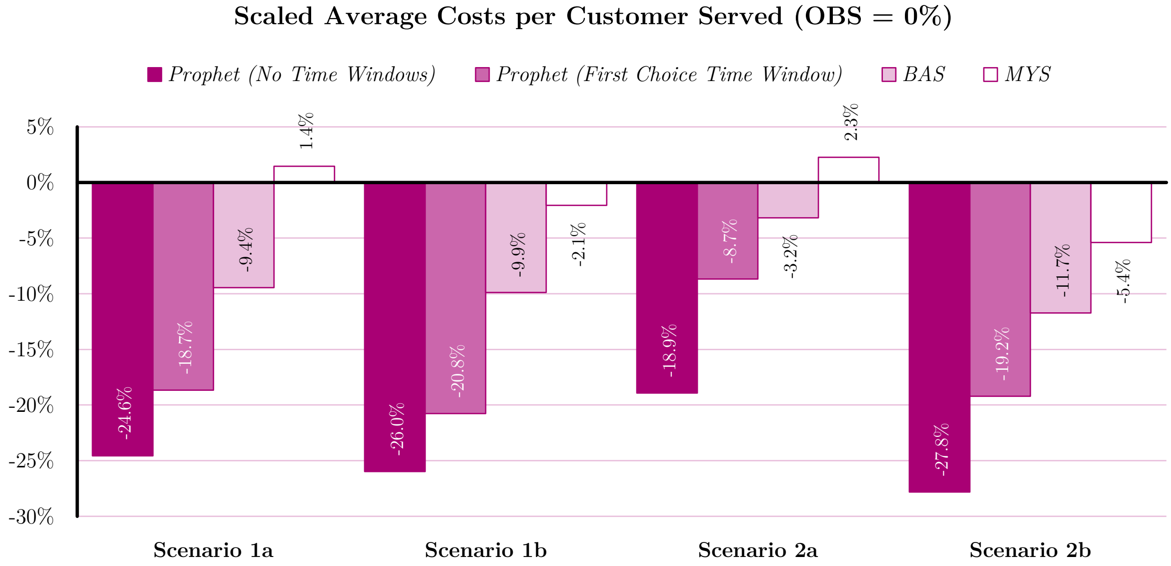


Figure 6.17. *Benchmark of different strategies for the average costs per customer that is served (Use Case 2)*

We see that BAS shows quite a decent overall performance for Use Case 2 compared to the prophet strategies, although we could do better. We have to take into account that the prophet benchmarks can probably never be achieved in reality, due to the many unknown factors in real-life. BAS in most cases achieves a performance somewhere in the middle between the prophet strategies and OBS, and increases the customer satisfaction as well as the route efficiency of an e-retailer significantly compared to OBS. However, we see that there is still space to improve BAS. The prophet strategy that only considers the first preference time windows for instance has less flexibility in the sense of customer satisfaction compared to BAS, because in reality customers would not only select their first preference time window. They may just as well select a second or third preference time window. This additional flexibility may compensate at least for a part of the difference in performance between BAS and our second prophet strategy.

We see again that it could be worthwhile spending some time to find out how the parameters of BAS should be tuned for different applications in practice of the e-retailer case. There is for sure space to improve BAS, as we have shown with the performance of the prophet strategies. We therefore recommend to do further research in the field of parameter tuning for BAS.

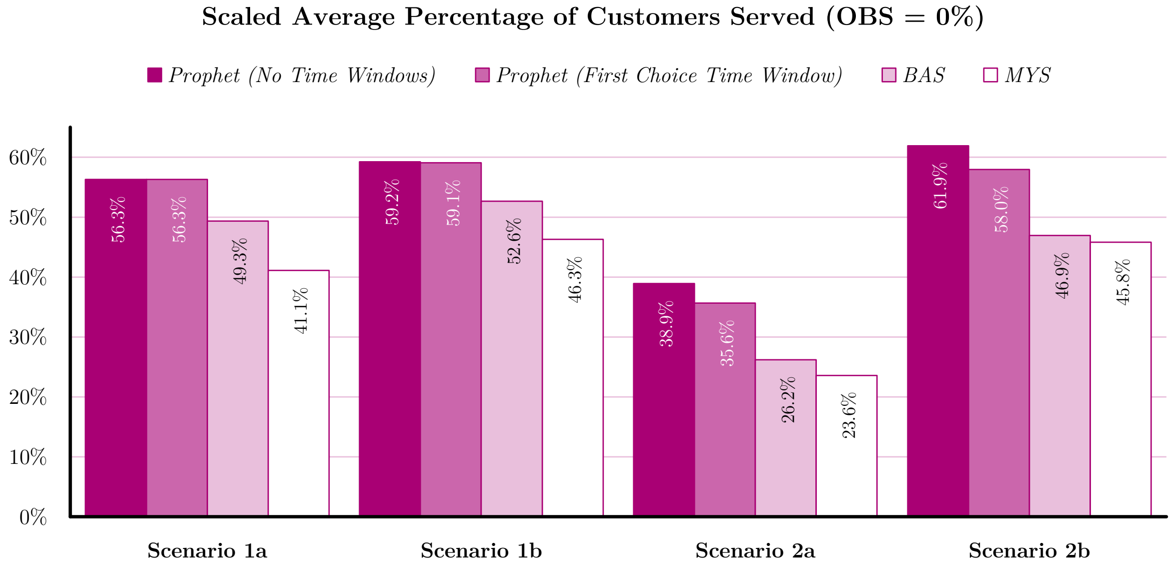


Figure 6.18. *Benchmark of different strategies for the average percentage of customers served (Use Case 4)*

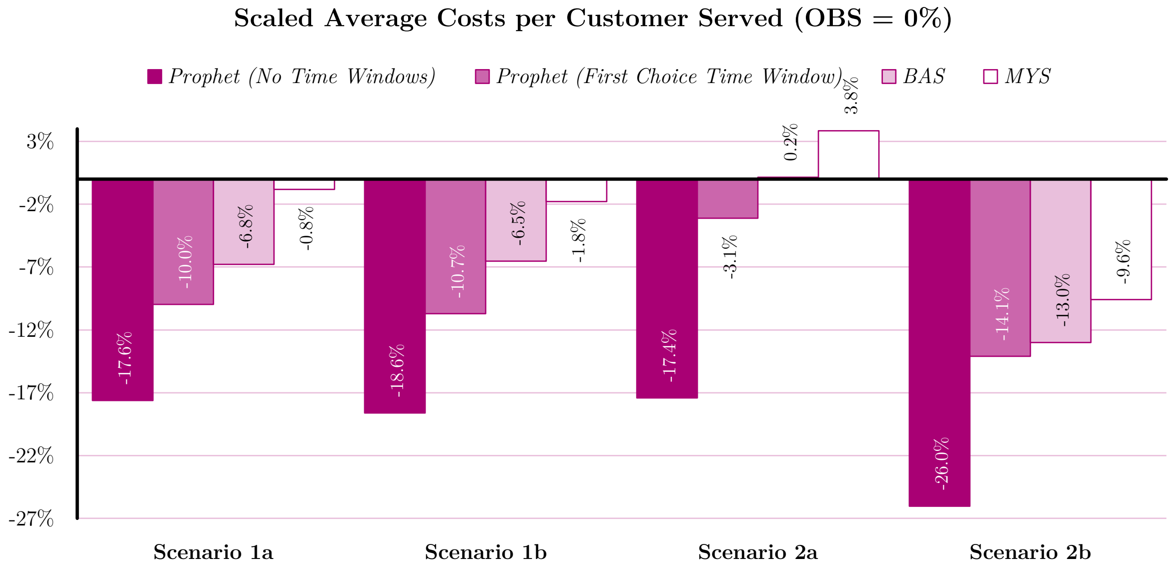


Figure 6.19. *Benchmark of different strategies for the average costs per customer that is served (Use Case 4)*

For Use Case 4 we see the same phenomenon as we saw for Use Case 2. We see that the results of the prophet strategy with no time windows, as may be expected, are always the best for both KPIs for all scenarios. The prophet strategy with the first preference time window already has a worse performance on average. Again, BAS in some cases approaches the performance of this second prophet strategy. We think that, although it may be difficult given that we need to have a low response time to customer requests, BAS can still be improved by for instance tuning the parameters used in a better way.

In Appendix B.5 we present some additional results for the benchmark of our strategies against the prophet strategies. An interesting observation to highlight here is that the prophet strategies structurally have a higher utilization compared to BAS. This again is an indication that there is space for improvement for BAS, at least in terms of available vehicle capacity.

Summarizing, we obtained an upper bound (prophet strategies) and we have a lower bound (OBS performance) for the performance of BAS. Currently we are somewhere in between at least for the scenarios we considered, but we are confident that more improvements can be made in further research.

6.5.2. Impact of Intermediate Optimization Calls

In this section we study the impact of intermediate optimization calls for Use Case 2 and OBS. We investigate whether it is really worthwhile to put efforts in the technical implementation of these intermediate optimization calls. Figure 6.20 and Figure 6.21 show that on average the performance for OBS without intermediate optimization calls is indeed worse compared to OBS with intermediate optimization calls. Especially for Scenario 2, we see quite a large difference both in terms of the percentage of customers that is served as well as in terms of the costs per customer that is served.

In Appendix B.6 we present additional results of the comparison of the two strategies. We also give the resulting 95% confidence intervals obtained with the paired-t approach. In most cases, we see that OBS significantly outperforms OBS (NoOpt) with 95% confidence. The intermediate optimization calls are primarily intended to reduce the total delivery costs. This is due to the fact that we optimize the delivery routes for a given set of customer orders. We do not accept any optimization response in which not all customers are served, but we do also not add any customer orders. Therefore, before and after an optimization call the number of customers that is served remains equal. However, because the routes are typically more efficient, it becomes easier to accept new customers, which is the indirect effect of the intermediate optimization calls. We see this confirmed in Figure 6.20, because for all scenarios that we consider the average percentage of customers served is higher for OBS compared to OBS (NoOpt).

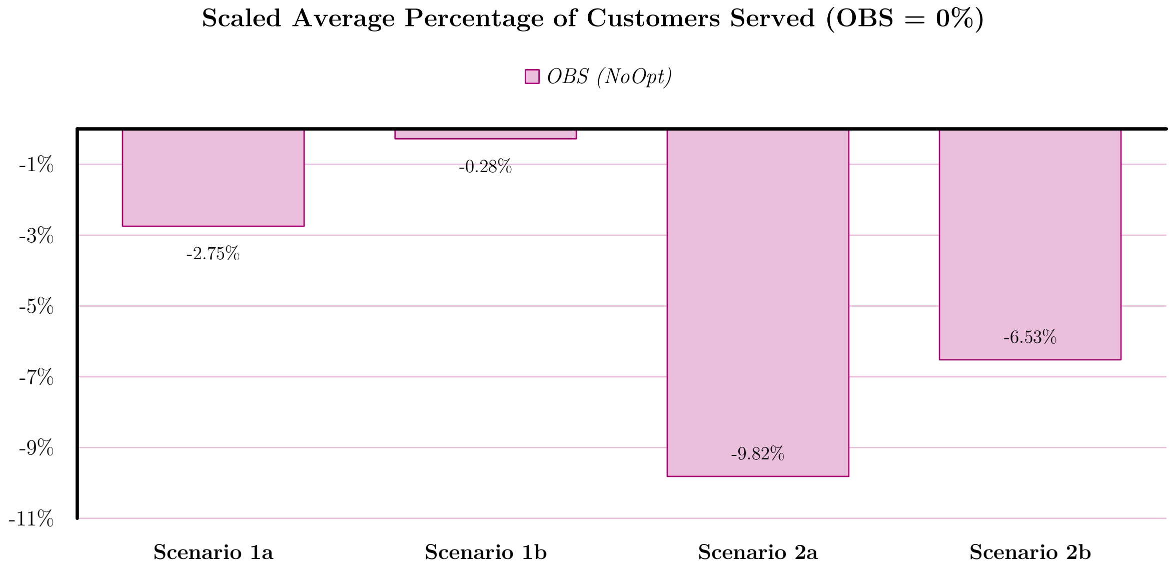


Figure 6.20. *OBS (NoOpt) compared to OBS for the percentage of customers served*

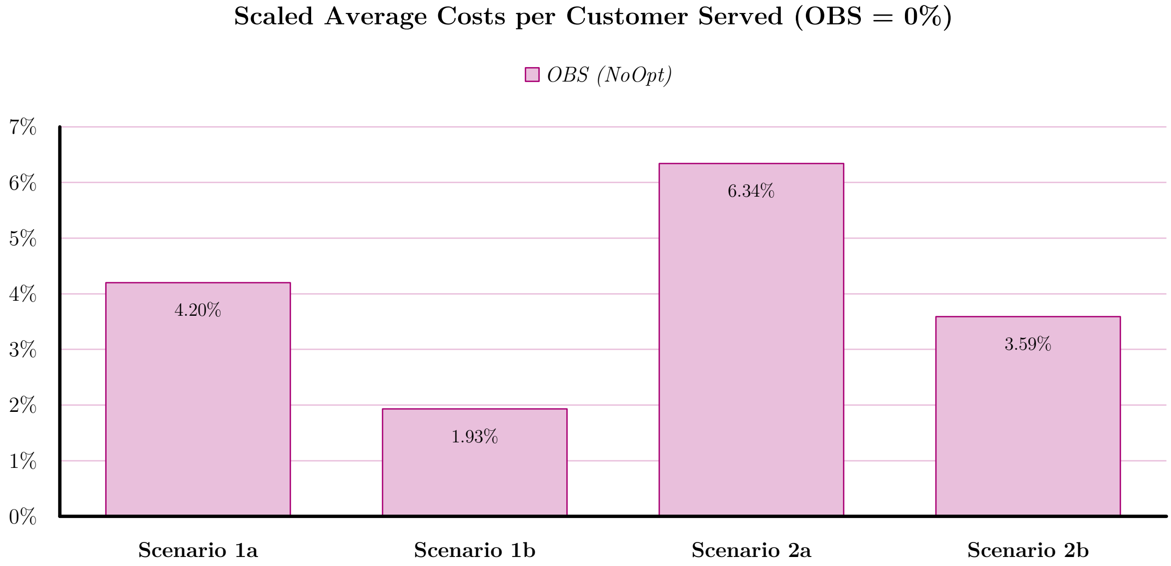


Figure 6.21. *Delivery costs per customer for OBS (NoOpt) compared to OBS*

To obtain more insight into the direct impact of the intermediate optimization calls on the total delivery costs, we analyzed the development of the total costs over the booking period for all replications of Scenario 2a. For this scenario we observe the highest difference between OBS with intermediate optimization and without intermediate optimization. Figure 6.22 presents the results, categorized by replication. Note that we do not show the effect of the final optimization call here, whereas this is taken into account in Figure 6.20 and Figure 6.21. We express the time in the cumulative percentage of customers that have arrived, instead of expressing the time as the percentage of the booking period that has elapsed. This makes sure that the points where the total costs drop after an intermediate optimization call are more equally spread, because we do an intermediate optimization call after every 20 customers instead of after a fixed time of the booking period. When looking careful to the results, we do indeed clearly see the drops in total costs. In some cases, the lines of OBS with intermediate optimization reach a higher value for the total costs than the lines of OBS without intermediate optimization. This can be an indication of having accepted more customers. For all replications, we see that in the end, the total costs are always lower for OBS with intermediate optimization calls, even though we know that more customers have been served on average. The results of our analyzes give us enough confidence to say that indeed implementing intermediate optimization calls is worthwhile, even when they take a lot of time and effort to implement.

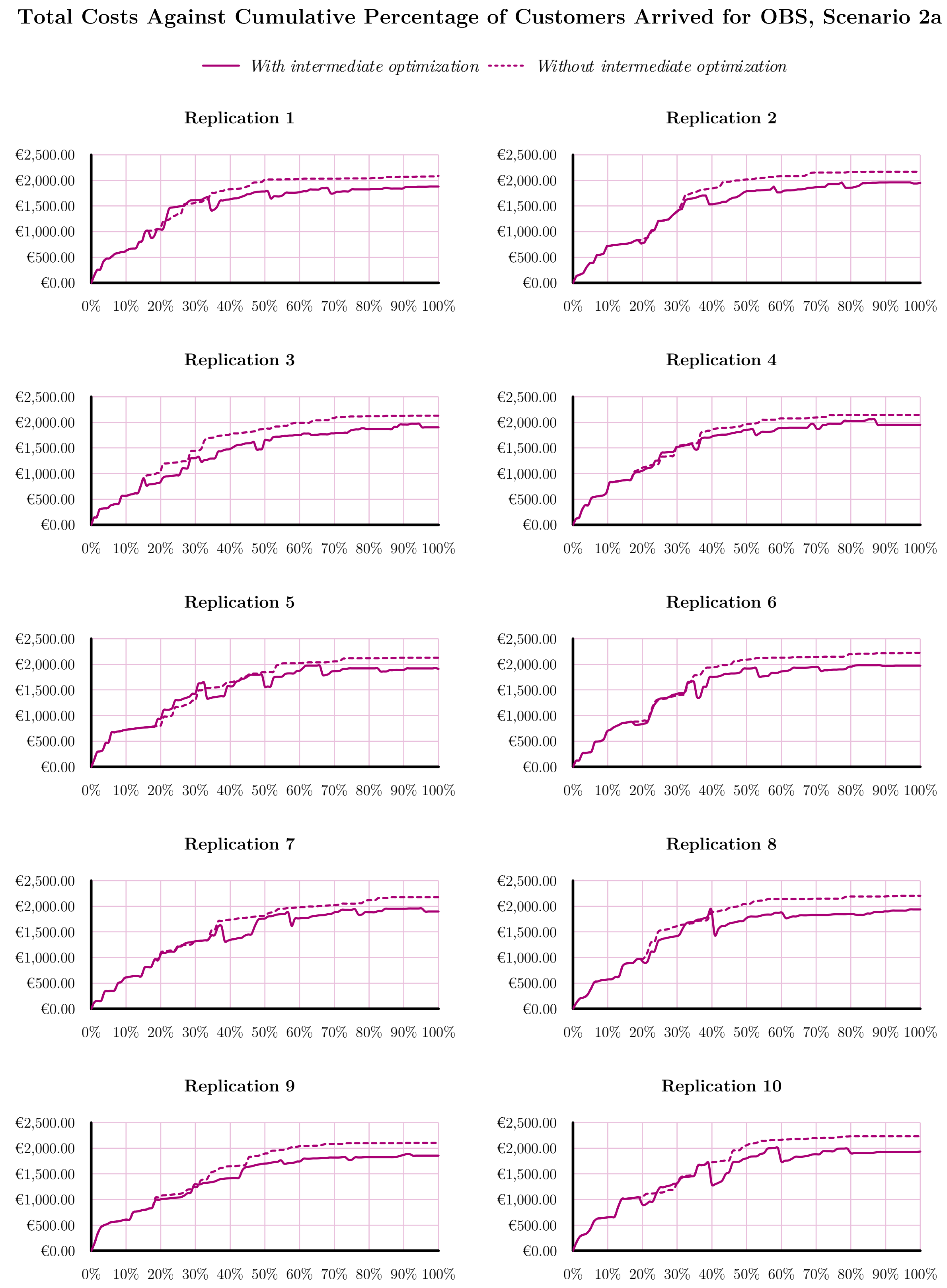


Figure 6.22. *Comparison of the development of the total costs against the cumulative percentage of customers that arrived for OBS with and without intermediate optimization calls (Scenario 2a, Use Case 2)*

6.6. Conclusion

In this chapter we analyzed the results of our computational experiments. In Section 6.1, Section 6.2, Section 6.3 and Section 6.4 we investigated whether or not our results confirm the four hypotheses that we formulated in Chapter 3. In Section 6.5 we analyzed some results of our computational experiments that are not directly related to one of the hypotheses we formulated but may give ORTEC interesting insights which may be useful in practice. We are now able to answer our fifth set of research questions:

5) *What are the insights that we can obtain from the results of our simulations?*
   a. *Do the simulation results confirm the hypotheses defined earlier?*
   b. *What general results do we observe from our experiments?*

Our first hypothesis is formulated to investigate if there are any benefits in using a myopic dynamic strategy (MYS) that can change the fleet composition during the booking period, compared to a static strategy (OBS) that cannot. Our results show that, in cases where forecasts (based on historical data) for the ratio of customer orders with the different order types are not accurate, MYS on average shows a better performance than OBS. In cases where the forecasts are accurate, we cannot show that there is a significant difference between the performance of MYS and OBS for Use Case 2. The results therefore give us reason to believe that our first hypothesis is indeed valid.

Our second hypothesis is formulated to investigate if we can improve a myopic dynamic strategy (MYS) by implementing smarter decision mechanisms for our smart dynamic strategy (BAS). The results that we obtained for both Use Case 2 and Use Case 4 support this hypothesis. Especially for Use Case 2, where we only have small vehicles, the margins with which BAS outperforms MYS are quite large for all our scenarios. Other interesting results are that the average response time to a customer request of available time windows for BAS is lower than for MYS in all our scenarios, and the average number of time windows offered to a customer always higher. This shows that, in terms of customer satisfaction, BAS does a much better job than MYS, and not only on our main KPIs.

Our third hypothesis is formulated to investigate whether it is worthwhile to put much effort in implementing a smart dynamic strategy (BAS) when we know that on average more customers arrive than can be served anyway. In such cases we may just prefer to start with an initial fleet composition that maximizes the fleet capacity and use a static strategy (OBS (Large Initial Fleet)) during the booking period. However, our results indicate that, for the scenarios we study, this turns out not to be true. For all scenarios we see that BAS outperforms OBS (Large Initial Fleet) in terms of customer satisfaction. Also, in terms of route efficiency BAS outperforms OBS (Large Initial Fleet) in most cases. For this reason, we are strongly inclined to consider our third hypothesis as a valid one.

Our last hypothesis is formulated to investigate the robustness of the performance of our smart dynamic strategy (BAS) with regard to changing input distributions for the spread of customer arrivals over the booking period and with regard to forecast errors. As we expected, we cannot find a significant difference in performance when we change the spread of customer arrivals from an empirical pattern to a pattern with an increasing average arrival frequency over the booking period, given that we have accurate forecasts of the new distribution. Furthermore, we see that, if forecast errors with regard to the cumulative percentage of customers that have arrived at a certain time are not too large, BAS on average still shows a better performance than OBS (Large Initial Fleet) and MYS. If the forecast errors become too large though, the results are worse. Although the margins are quite small, this shows us the importance of tuning the parameters for BAS in such a way that forecast errors can be corrected by the strategy. Especially in situations where BAS is over-forecasting the cumulative percentage of customers that have arrived, the performance may deteriorate significantly, both for fixed forecast errors and fluctuating forecast errors. These results give us reason to believe that our fourth hypothesis is a valid one as well.

After obtaining all results for the validation of our hypotheses, we performed a benchmark of the performance of our strategies against two prophet strategies. These strategies know upfront everything about each customer that will arrive during the booking period and may therefore select the most attractive customers in order to serve as many customers as possible. We see that our strategy BAS shows a decent performance, somewhere half the way between the lower bound set by the performance of OBS and the upper bound set by the performance of the prophet strategies. At the same time this shows that there is enough space for future improvements of our strategy BAS.

Besides the benchmark result, we also show the impact of intermediate optimization calls. If we would not make use of intermediate optimization of the delivery routes, we see this directly back in higher costs per customer that is served, and indirectly also in a lower percentage of customers that is served. This confirms the importance of making use of intermediate optimization calls and justifies their implementation in ORTEC's software solutions.

7. Conclusions and Recommendations

After analyzing all results, we now come back to our main research question as formulated in Chapter 1:

> *How can ORTEC deal in a proper way with an unfixed fleet composition when implementing a strategy for operational time slot management for its clients?*

In this chapter we respond this research question in Section 7.1, by summarizing the responses to our five sets of research questions as found in the previous five chapters. In Section 7.2 we reflect on the contribution of our research to literature as well as on the contribution of our research for practical applications. Finally, we conclude this chapter with recommendations for further research in Section 7.3.

7.1. Main Findings

In our search for a proper way to deal with an unfixed fleet composition when implementing a strategy for operational TSM for ORTEC's clients, similar to an e-retailer of our context, we found that not much research has been done on this specific topic. However, there are many publications that provide insights into several related topics, such as the vehicle routing problem, TSM in attended home delivery and the modeling of customer behavior. We find that our e-retailer case is a version of the HFVRPTW with an online component (during the booking period) and an offline component (after the booking period ends). Literature unfortunately does not provide much insights on how to deal with an unfixed fleet composition in a strategy for operational time slot management, in the context of the e-retailer case. Therefore, to obtain these insights, we depend on the results of the computational experiments for the scenarios of the e-retailer case that we consider. We did experiments for the e-retailer case with two different order types, one central depot and three different vehicle types. Of the three vehicle types, two types are small vehicles and are dedicated to the delivery of one order type each. The third vehicle type is large and can deliver any order type, but is more expensive. Our experiments consider use cases with either e-retailers that only own vehicles of the first two vehicle types (Use Case 2), or of all vehicle types (Use Case 4). Also, we only studied cases in which on average more customer demand comes in during the booking period than can be served with the maximum possible capacity of the fleet composition (given the number of drivers that is available to drive a delivery vehicle).

In our context we distinguish between dealing in a static way with an unfixed fleet composition in a strategy for TSM, or in a dynamic way. Although the fleet composition is unfixed during the booking period, a static strategy fixes the fleet composition after determining an initial fleet composition at the start of the booking period. A dynamic strategy does not fix the fleet composition until the final optimization of the delivery routes after the booking period ends. We designed three strategies that ORTEC may implement to deal with an unfixed fleet composition. The first strategy (OBS) is a static strategy and can be implemented without making any changes to the present software solutions. The second strategy (MYS) is a myopic dynamic strategy. This strategy tries to change the fleet composition when a customer order cannot be accepted with the current composition. In this way MYS aims to be able to accept the customer order in the end. The third strategy we designed (BAS) is the strategy that shows the best performance in terms of the average percentage of customers that can be served and the average costs per customer that is served. BAS may change the fleet composition during the booking period and may also reject customers if they are considered to be unattractive. When taking decisions with regard to changing the fleet composition or rejecting customers, BAS tries to find a balance between forecasts based on historical data and observed data from customer orders that have already arrived during the booking period.

We tested the performance of the strategies for two different scenarios with different spreads of customer locations around the central depot and different customers. For both scenarios we consider two cases. The first one (a) is a situation in which the observed percentage of customers for each order type is on average equal to the percentage of customers with that order type according to historical data. For the second case (b) the observed ratio of customers of each order type does not equal the historical ratio. For all scenarios and for all use cases BAS on average shows the best performance of our three strategies. Especially for measures of customer satisfaction BAS outperforms MYS and OBS in most cases, whereas this generally does not go at the cost of the measures for route efficiency that we consider.

To get insight into how good the performance of our strategies actually is, we benchmarked all our strategies against two so-called prophet strategies. These prophet strategies make use of information that is normally not available to an e-retailer, such as the time window preferences of customers and the sequence in which customer orders come in. The strategies know all details of customer orders that will arrive during the booking period upfront and they solve the HFVRPTW for all these customer orders. The aim is to accept as many customers as possible and then minimizing the total delivery costs. The first prophet strategy pretends as if each customer is indifferent with regard to the time window preferences, so each customer would select any time window. For the second prophet strategy a customer may only be accepted when the first preference time window is available for that customer. The prophet strategies both determine the optimal composition for which most customers can be accepted against minimum costs. The results of the benchmark are re-displayed in Figure 7.1 and Figure 7.2.

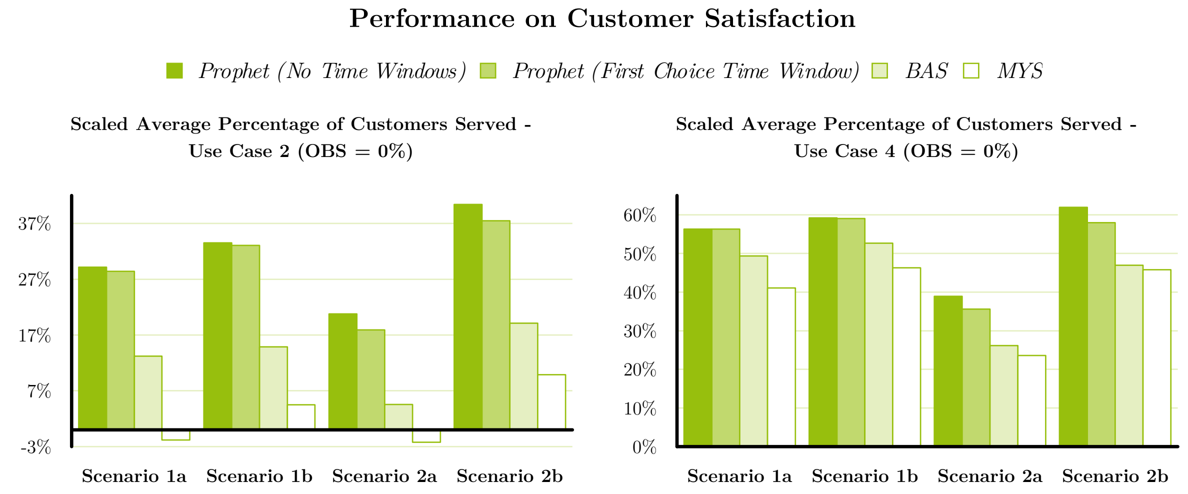


Figure 7.1. *Performance benchmark of our strategies against prophet strategies for customer satisfaction*

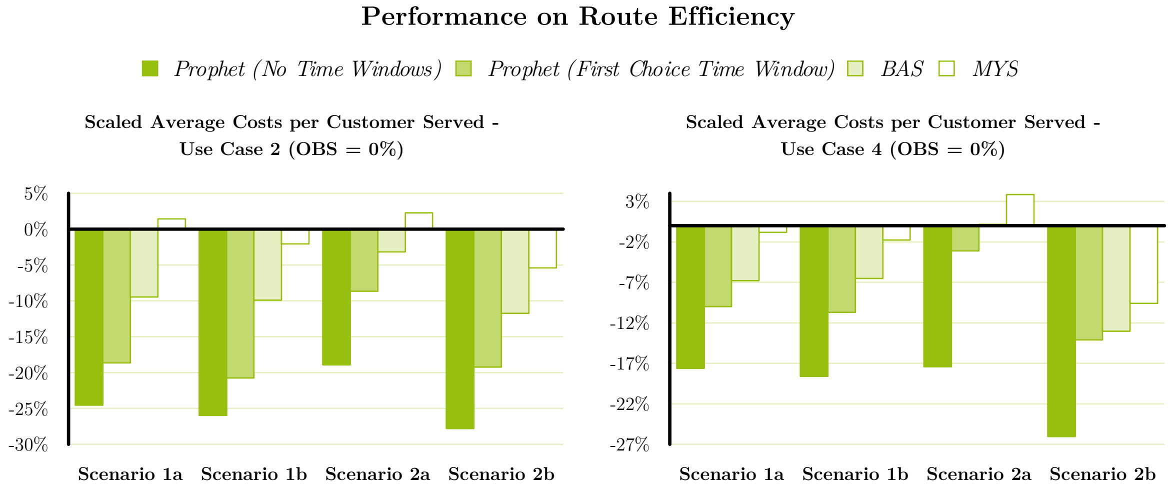


Figure 7.2. *Performance benchmark of our strategies against prophet strategies for route efficiency*

We see that BAS in all cases performs reasonable compared to the prophet strategies, and much better than OBS and MYS. The results we obtained give us confidence that ORTEC can indeed achieve a good performance in terms of customer satisfaction and route efficiency when implementing BAS as a strategy for operational TSM for their clients. Of course, we need to ask ourselves if the results we obtained for our case studies are useful and trustworthy for ORTEC to rely on when we consider practical cases that are different from our scenarios. We address this consideration in Section 7.2. Also, the prophet benchmarks show that we have enough opportunities to further improve BAS. Our results show that especially in terms of sensitivity of the performance to forecast errors BAS can and should be improved. It is therefore vital to do further research on how we can make BAS perform even better, for instance by introducing a better way of tuning its parameters, and obtain even more satisfying results.

7.2. Contribution to Literature and Practice

As we stated before, our research is in a realm of literature in which few contributions have been published. This has to do with the fact that time slot management is a relatively new field of research compared to for instance the VRP. Our findings present an interesting strategy for operational time slot management. Although the strategy we designed is still in a phase of initial development, we are confident that the strategy can be further developed in such a way that we can use the strategy in a broader context for other problems in literature. This is a challenge for the further research intended to improve our strategy.

We believe that the problem we consider can be generalized in such a way that it becomes a very interesting problem for literature, to which little attention has been paid so far. Our findings may be considered as yet another incentive to do more research in the realm of operational time slot management. Especially in an era in which online shopping becomes more and more dominant, good strategies for operational time slot management may make the difference for e-retailers. The results that we obtained in this research underline once more the importance of operational time slot management for e-retailers that are similar to the ones from our context.

Our focus has been quite a practical one, because we did research in the context of ORTEC's software solutions. ORTEC has many (potential) clients that resemble e-retailers from our context and that struggle with operational time slot management. Therefore, our findings are promising for ORTEC, because they indicate that ORTEC can improve the performance of their clients by implementing a strategy like BAS. Of course, before implementing the strategy in its software solutions, ORTEC should first intensively study the case of each client and perform simulations with data that is specific for that client.

It is important to experiment with BAS in different scenarios than the ones we considered, as we also point out in our recommendations for further research. This will create a stronger support for an eventual implementation of the strategy in ORTEC's software solutions. However, we feel confident to say based on our results so far, that it is really worthwhile for ORTEC to consider a potential implementation of BAS in the software solutions offered to ORTEC's clients. The benefit of this research lays in the fact that we used OTS and CVRS to implement our strategies. If ORTEC desires to start using BAS in the future, the fact that BAS already interacts with ORTEC's cloud services makes the implementation much easier. We believe that this is an important contribution for practice of our research.

7.3. Recommendations

In this section we discuss some recommendations for further research that we distinguish. During the process of this research, we observed many issues that can possibly be improved. It would be too much to state all of them here, therefore we made a selection of the most important recommendations.

7.3.1. Increase of dimensions

Our first recommendation is to study more scenarios of the e-retailer case. We propose to consider data from different clients of ORTEC, to obtain insights into the impact on the performance of our strategies.

It would be beneficial to study cases in which more customers arrive during the booking period than the 100 that we limited ourselves to. Due to technical limitations it was not possible to perform experiments with more customer orders, but before we can possible implement our strategies in ORTEC's software solutions it is a must to test them on larger instances.

Besides considering more customers arriving during the booking period, we may also consider a larger delivery fleet and more drivers available to drive the delivery vehicles. It would be interesting as well to study what happens when we have customer locations that are spread across a larger region than the regions we considered, or a smaller region of only one city for instance.

Another factor that may be worthwhile to increase is the number of time windows that is available. We restricted ourselves to 4 time windows of equal length, but there exist practical cases in which more time windows with different lengths are offered by an e-retailer. This may change the decisions taken by BAS as well as the customer choice behavior.

7.3.2. Generalization of the context

We also recommend to investigate how BAS performs in a generalized version of the e-retailer case. So far, we considered the e-retailer case with two order types, three vehicle types and one central depot only. It would be very useful to obtain insights into the performance of BAS when we have more order types, vehicle types and maybe even multiple depots.

Another interesting aspect is the influence of the driver assignment in our problem context. We now considered drivers that all have equal capabilities and an equal number of working hours on a day. In practice, situations may occur where not all drivers can drive any vehicle type, and some drivers work only half a day for instance. This makes the driver assignment much more complex, and it would be interesting to see if we can reach a satisfactory performance with BAS for such cases.

Besides the above considerations, it would also be useful to study how BAS performs in Use Case 3 and in different use cases that we did not consider in this research, to get a more complete overview of how valuable BAS is as a strategy.

The generalization of all these factors allows us to reconstruct many more practical cases than the ones we could reconstruct so far. However, in order for our strategies to keep working for all cases that we can construct, a generalization of the strategies is required as well. Although it may require some effort, we believe that a generalization of the context would not only contribute to ORTEC's practice, but also to the literature that is available on topics related to operational time slot management.

7.3.3. Parameter tuning and improvement of BAS

As we mentioned several times already, it is very important for ORTEC to spend time to develop a good method to tune the parameters that BAS uses, if ORTEC indeed wants to implement BAS. The parameters that BAS uses need to be configured differently for each practical application of the e-retailer case. Our experiments with the prophet strategies already indicated that there is space to improve BAS.

It would also be interesting to investigate if there is a method that can tune the parameters of BAS online, during the booking period, by applying machine learning techniques. Such a method would ideally keep track of the development of our KPIs over the booking period, and recognize certain patterns in this development. By for instance learning offline which decisions are the best to take when a certain pattern occurs, we can save a lot of time to take these decisions when we recognize the pattern online. To create a decision framework for BAS offline, we could make use of historical datasets, divided in training sets and verifications sets. This approach is completely different from the way that BAS is designed now, because currently we fix the parameters during the whole booking period. Therefore, implementing a method that tunes the parameters online, based on information obtained offline, requires a lot of effort. It would probably give sufficient research content to write a whole new thesis about. However, we believe it may be beneficial to explore the possibilities for designing such a method, because it may significantly improve the performance of BAS.

Another recommendation is to study cases in which BAS does not reject customers based on their order quantity, but for instance based on their location (distant customers may be unattractive). In cases where we have data about the order values, it may also be interesting to investigate the performance for BAS when the unattractiveness of customers is determined based on the ratio of the order value and the order quantity (the lower the ratio the less attractive the customer). These are just two examples of different criteria that BAS can use to determine whether a customer is unattractive. We have already seen that in Scenario 2 the first example (rejecting based on location) might have resulted in a better performance for BAS, so we believe it is worthwhile studying this aspect in a deeper way.

A final recommendation regarding BAS, is that we propose to investigate whether BAS could be used on a strategical/tactical level of control, for fleet dimensioning decisions. We could for instance use the limit imposed by the number of drivers to set the maximum amount of delivery vehicles that an e-retailer wants to acquire. If we set the number of vehicles for each vehicle type equal to the number of drivers, BAS may add as many vehicles of a vehicle type as necessary. Starting with an as cheap as possible initial fleet, we make sure that the final composition consists of the required vehicles to ensure a good performance, without having overcapacity in the delivery fleet. If an e-retailer then studies the results for different realizations of its ordering process, the average number of vehicles used by BAS per vehicle type may be a good starting point to base strategical/tactical fleet dimensioning decisions on.

7.3.4. Customer choice behavior

As we have little information about customer choice behavior, we just assumed a simple uniform distribution to model customer preferences. However, in practice much more complex models may be required to accurately model customer preferences. We believe that for studies in the field of operational time slot management the customer choice behavior is an element with an essential influence on the performance of different strategies for operational time slot management. An e-retailer may do as much as possible to offer an as high as possible number of time windows to each customer, but as soon as the list with offered time windows does not contain any of the customer's preferences the e-retailer loses the customer anyway. Therefore, we propose to study the impact of the customer choice behavior in a deeper way, before considering to implement any strategy for operational time slot management.

7.3.5. General recommendations

Nowadays, we see that topics like same-day delivery and stochastic customers that arrive during the service period become more and more relevant in practice. Literature on these topics starts to emerge, and these topics also give a whole new dimension to operational time slot management. It may for most ORTEC clients that are similar to the e-retailers of our context not yet be a relevant topic in their practice, but developing a strategy for operational time slot that can deal with customers that request same-day delivery during the service period would give a huge competitive advantage for ORTEC clients. It may therefore be worthwhile for ORTEC to do research on these topics and how we can for instance incorporate them into BAS.

Another general remark that we want to conclude with, is that we recommend that ORTEC investigates how the way in which optimization algorithms in CVRS are configured can be improved for individual clients. This holds for intermediate optimization calls as well as for the final optimization of routes at the end of the booking period. Currently, not much attention is paid to the configuration for different clients, and most clients use standard templates. It can however be very beneficial to tune the configuration of the optimization algorithms differently for different clients, tailored to their practical needs. Also, to reduce the time that BAS spends verify whether a new fleet composition is feasible it could be beneficial to configure CVRS algorithms in a different way, without decreasing the performance of BAS. This will lead to a higher customer satisfaction, because the average response time to a customer request will decrease.

References

Agatz, N., Campbell, A., Fleischmann, M., & Savelsbergh, M. (2011). Time Slot Management in Attended Home Delivery. *Transportation Science*, *45*(3), 435–449. https://doi.org/10.1287/trsc.1100.0346

Agatz, N., Campbell, A. M., Fleischmann, M., & Savelsbergh, M. (2008). Challenges and opportunities in attended home delivery. In *The vehicle routing problem: Latest advances and new challenges* (pp. 379–396). Boston, MA: Springer.

Asdemir, K., Jacob, V. S., & Krishnan, R. (2009). Dynamic pricing of multiple home delivery options. *European Journal of Operational Research*, *196*(1), 246–257. https://doi.org/10.1016/J.EJOR.2008.03.005

Azi, N., Gendreau, M., & Potvin, J. Y. (2011). A dynamic vehicle routing problem with multiple delivery routes. *Annals of Operations Research*, *199*(1), 103–112. https://doi.org/10.1007/s10479-011-0991-3

Belfiore, P., & Yoshida Yoshizaki, H. T. (2009). Scatter search for a real-life heterogeneous fleet vehicle routing problem with time windows and split deliveries in Brazil. *European Journal of Operational Research*, *199*(3), 750–758. https://doi.org/10.1016/j.ejor.2008.08.003

Bent, R. W., & Van Hentenryck, P. (2004). Scenario-Based Planning for Partially Dynamic Vehicle Routing with Stochastic Customers. *Operations Research*, *52*(6), 977–987. https://doi.org/10.1287/opre.1040.0124

Braekers, K., Ramaekers, K., & Van Nieuwenhuyse, I. (2016). The vehicle routing problem: State of the art classification and review. *Computers and Industrial Engineering*, *99*, 300–313. https://doi.org/10.1016/j.cie.2015.12.007

Brandão, J. (2011). A tabu search algorithm for the heterogeneous fixed fleet vehicle routing problem. *Computers & Operations Research*, *38*(1), 140–151. https://doi.org/10.1016/J.COR.2010.04.008

Bräysy, O., & Gendreau, M. (2005a). Vehicle routing problem with time windows, Part I: Route construction and local search algorithms. *Transportation Science*, *39*(1), 104–118.

Bräysy, O., & Gendreau, M. (2005b). Vehicle routing problem with time windows, Part II: Metaheuristics. *Transportation Science*, *39*(1), 119–139.

Bruck, B. P., Cordeau, J.-F., & Iori, M. (2017). A practical time slot management and routing problem for attended home services. *Omega*. https://doi.org/10.1016/J.OMEGA.2017.11.003

Bühler, D., Klein, R., & Neugebauer, M. (2016). Model-based delivery cost approximation in attended home services. *Computers & Industrial Engineering*, *98*, 78–90. https://doi.org/10.1016/J.CIE.2016.05.014

Campbell, A. M., & Savelsbergh, M. (2006). Incentive Schemes for Attended Home Delivery Services. *Transportation Science*, *40*(3), 327–341. https://doi.org/10.1287/trsc.1050.0136

Campbell, A. M., & Savelsbergh, M. W. P. (2005). Decision Support for Consumer Direct Grocery Initiatives. *Transportation Science*, *39*(3), 313–327. https://doi.org/10.1287/trsc.1040.0105

Chang, Y., & Chen, L. (2007). Solve the Vehicle Routing Problem With Time Windows Via a Genetic Algorithm. *Discrete and Continuous Dynamical Systems Supplement*, 240–249.

Chen, C. F., & Chen, Y. C. (2014). A Dynamic Pricing Model for Time Slot Management in Attended

Home Delivery Logistics. *Transport Research Arena, Paris.*

Clarke, G., & Wright, J. W. (1964). Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. *Operations Research*, *12*(4), 568–581. https://doi.org/10.1287/opre.12.4.568

Cleophas, C., & Ehmke, J. F. (2014). When Are Deliveries Profitable? *Business and Information Systems Engineering*, *6*(3), 153–163. https://doi.org/10.1007/s12599-014-0321-9

Cook, T. M., & Russell, R. A. (1978). A simulation and statistical analysis of stochastic vehicle routing with timing constraints. *Decision Sciences*, *9*(4), 673–687.

Cordeau, J. F., Laporte, G., Savelsbergh, M. W. P., & Vigo, D. (2007). Chapter 6 Vehicle Routing. *Handbooks in Operations Research and Management Science*, *14*(C), 367–428. https://doi.org/10.1016/S0927-0507(06)14006-2

Dantzig, G. B., & Ramser, J. H. (1959). The Truck Dispatching Problem. *Management Science*. https://doi.org/10.1287/mnsc.6.1.80

Dell'Amico, M., Monaci, M., Pagani, C., & Vigo, D. (2007). Heuristic Approaches for the Fleet Size and Mix Vehicle Routing Problem with Time Windows. *Transportation Science*, *41*(4), 516–526. https://doi.org/10.1287/trsc.1070.0190

Desrosiers, J., Dumas, Y., Solomon, M. M., & Soumis, F. (1995). Time Constrained Routing and Scheduling. *Handbooks in Operations Research and Management Science*. https://doi.org/10.1016/S0927-0507(05)80106-9

Dondo, R., & Cerdá, J. (2007). A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows. *European Journal of Operational Research*, *176*(3), 1478–1507. https://doi.org/10.1016/j.ejor.2004.07.077

Ehmke, J. F., & Campbell, A. M. (2014). Customer acceptance mechanisms for home deliveries in metropolitan areas. *European Journal of Operational Research*, *233*(1), 193–207. https://doi.org/10.1016/J.EJOR.2013.08.028

Eilam Tzoreff, T., Granot, D., Granot, F., & Sošić, G. (2002). The vehicle routing problem with pickups and deliveries on some special graphs. *Discrete Applied Mathematics*. https://doi.org/10.1016/S0166-218X(00)00283-3

eMarketer. (2018). Retail Ecommerce Sales Worldwide, 2016-2021. Retrieved March 22, 2018, from https://www.emarketer.com/Chart/Retail-Ecommerce-Sales-Worldwide-2016-2021-trillions-change-of-total-retail-sales/215138

Flood, M. M. (1956). The Traveling-Salesman Problem. *Operations Research*, *4*(1), 61–75. Retrieved from http://www.jstor.org/stable/167517

Gendreau, M., & Tarantilis, C. (2010). Solving large-scale vehicle routing problems with time windows: The state-of-the-art. *Cirrelt*, (4). Retrieved from https://www.cirrelt.ca/DocumentsTravail/CIRRELT-2010-04.pdf

Golden, B., Assad, A., Levy, L., & Gheysens, F. (1984). The fleet size and mix vehicle routing problem. *Computers & Operations Research*, *11*(1), 49–66. https://doi.org/10.1016/0305-0548(84)90007-8

Hernandez, F., Gendreau, M., & Potvin, J. Y. (2017). Heuristics for tactical time slot management: a periodic vehicle routing problem view. *International Transactions in Operational Research*. https://doi.org/10.1111/itor.12403

Hungerländer, P., Maier, K., Pöcher, J., Rendl, A., & Truden, C. (2018). Solving an On-line Capacitated Vehicle Routing Problem with Structured Time Windows. In A. Fink, A. Fügenschuh, & M. Geiger (Eds.), *Operations Research Proceedings 2016* (pp. 127–132). Springer, Cham. https://doi.org/10.1007/978-3-319-55702-1_18

Jiang, J., Ng, K. M., Poh, K. L., & Teo, K. M. (2014). Vehicle routing problem with a heterogeneous fleet and time windows. *Expert Systems with Applications*, *41*(8), 3748–3760. https://doi.org/10.1016/j.eswa.2013.11.029

Kallehauge, B., Larsen, J., Madsen, O. B. G., & Solomon, M. M. (2005). Vehicle routing problem with time windows. In *Column generation* (pp. 67–98). Springer.

Klein, R., Neugebauer, M., Ratkovitch, D., & Steinhardt, C. (2017). Differentiated Time Slot Pricing under Routing Considerations in Attended Home Delivery. *Transportation Science*. https://doi.org/https://doi.org/10.1287/trsc.2017.0738

Knight, K. W., & Hofer, J. P. (1968). Vehicle scheduling with timed and connected calls: A case study. *Journal of the Operational Research Society*, *19*(3), 299–310.

Koç, Ç., Bektaş, T., Jabali, O., & Laporte, G. (2015). A hybrid evolutionary algorithm for heterogeneous fleet vehicle routing problems with time windows. *Computers and Operations Research*, *64*, 11–27. https://doi.org/10.1016/j.cor.2015.05.004

Koç, Ç., Bektaş, T., Jabali, O., & Laporte, G. (2016). Thirty years of heterogeneous vehicle routing. *European Journal of Operational Research*, *249*(1), 1–21. https://doi.org/10.1016/j.ejor.2015.07.020

Kritikos, M. N., & Ioannou, G. (2013). The heterogeneous fleet vehicle routing problem with overloads and time windows. *International Journal of Production Economics*, *144*(1), 68–75. https://doi.org/10.1016/j.ijpe.2013.01.020

Law, A. M. (2015). *Simulation Modeling and Analysis* (5th ed.). New York, NY: McGraw-Hill Education.

Lenstra, J. K., & Rinnooy Kan, A. H. G. (1981). Complexity of vehicle routing and scheduling problems. *Networks*. https://doi.org/10.1002/net.3230110211

Li, F., Golden, B., & Wasil, E. (2007). A record-to-record travel algorithm for solving the heterogeneous fleet vehicle routing problem. *Computers and Operations Research*, *34*(9), 2734–2742. https://doi.org/10.1016/j.cor.2005.10.015

Munari, P., Dollevoet, T., & Spliet, R. (2016). A generalized formulation for vehicle routing problems, (1), 1–19. Retrieved from http://arxiv.org/abs/1606.01935

Nguyen, D. H., De Leeuw, S., & Dullaert, W. E. H. (2018). Consumer Behaviour and Order Fulfilment in Online Retailing: A Systematic Review. *International Journal of Management Reviews*, *20*(2), 255–276. https://doi.org/10.1111/ijmr.12129

ORTEC B.V. (2018). ORTEC - Optimize Your World Corporate Website. Retrieved March 22, 2018, from https://www.ortec.com/

Paraskevopoulos, D. C., Repoussis, P. P., Tarantilis, C. D., Ioannou, G., & Prastacos, G. P. (2008). A reactive variable neighborhood tabu search for the heterogeneous fleet vehicle routing problem with time windows. *Journal of Heuristics*, *14*(5), 425–455. https://doi.org/10.1007/s10732-007-9045-z

Pessoa, A., Uchoa, E., & De Aragão, M. P. (2009). A robust branch-cut-and-price algorithm for the heterogeneous fleet vehicle routing problem. *Networks*. https://doi.org/10.1002/net.20330

Pullen, H. G. M., & Webb, M. H. J. (1967). A computer application to a transport scheduling problem. *The Computer Journal*, *10*(1), 10–13. https://doi.org/10.1093/comjnl/10.1.10

Punakivi, M., & Saranen, J. (2001). Identifying the success factors in e-grocery home delivery. *International Journal of Retail & Distribution Management*, *29*(4), 156–163. https://doi.org/10.1108/09590550110387953

Solomon, M. M. (1987). Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints. *Operations Research*, *35*(2), 254–265. https://doi.org/10.1287/opre.35.2.254

Taillard, E. D. (1999). A heuristic column generation method for the heterogeneous fleet VRP. *RAIRO - Operations Research*. https://doi.org/10.1051/ro:1999101

Vigo, D., & Toth, P. (2014). *Vehicle Routing; Problems, Methods, and Applications. MOS-SIAM Series on Optimization*. https://doi.org/doi:10.1137/1.9781611973594

Yang, X., & Strauss, A. K. (2017). An approximate dynamic programming approach to attended home delivery management. *European Journal of Operational Research*, *263*(3), 935–945. https://doi.org/10.1016/J.EJOR.2017.06.034

Yang, X., Strauss, A. K., Currie, C. S. M., & Eglese, R. (2016). Choice-Based Demand Management and Vehicle Routing in E-Fulfillment. *Transportation Science*, *50*(2), 473–488. https://doi.org/10.1287/trsc.2014.0549

Yepes, V., & Medina, J. (2006). Economic Heuristic Optimization for Heterogeneous Fleet VRPHESTW. *Journal of Transportation Engineering*, *132*, 303–311. https://doi.org/10.1061/(ASCE)0733-947X(2006)132:4(303)

The icons that are used in Figure 1.1, Figure 2.3, Figure 3.2, Figure 4.3 and Figure 4.4 are adapted versions from:

- Icons created by **Freepik** from www.flaticon.com
- Icons created by **Smashicons** from www.flaticon.com
- *Optimize* by **H Alberto Gongora** from the Noun Project
- *Delivery Car* by **BomSymbols** from the Noun Project
- *Courier service* by **Symbolon** from the Noun Project
- *Bicycle* by **PictoBike** from the Noun Project

Appendix A. Validation of Distributions

In this appendix we validate the distributions (Chapter 5) we use in our simulation tool for determining the number of customers in a simulation run (Section A.1) and determining the spread of the customer arrivals over the booking period, which is one day in our case. For the latter we distinguish between a spread according to an empirical distribution (Section A.2) according to a common pattern in practice which we use in most of our experiments, a spread with an increasing arrival rate over the booking period (Section A.3) and an uniform spread over the booking period (Section A.4). For each distribution we first perform a visual test in which we the expected and observed densities, as well as comparing the expected cumulative distribution with the observed cumulative distribution. Subsequently, we perform a chi-square goodness-of-fit test, according to the method introduced by Law (2015).

A.1. Distribution of Number of Customers

For the distribution of the number of customers that arrive during the booking period we deal with a practical restriction with regard to the load for ORTEC's servers. We cannot make a lot of requests with many customers in parallel, so we limited the average number of customers to 100. We model the number of customers that arrive during the booking period with a Poisson random variable $N = 2 \times X$ where $X \sim$ Poisson(50). We define N as X multiplied by 2 so we have a somewhat larger spread compared to for instance the case in which $N = Y \sim$ Poisson(100), as shown in Equation (A.1):

$$\text{Var}[2X] = 2^2 \times \text{Var}[X] = 4 \times \lambda_X = 200 > 100 \;= \lambda_Y = \text{Var}[Y] \tag{A.1}$$

We performed a validation test for the first 100 realizations of the number of customers in our simulation tool, which are displayed in Table A.1.

Table A.1. *The first 100 values that our simulation tool generates for the number of customer arrivals per day*

1-10	11-20	21-30	31-40	41-50	51-60	61-70	71-80	81-90	91-100
116	108	104	96	82	82	108	110	66	98
102	86	94	74	96	96	88	120	102	86
126	110	66	108	102	130	90	94	106	120
114	102	84	84	124	108	112	98	88	114
120	88	92	108	80	96	80	100	112	106
112	110	96	72	80	102	106	90	130	120
104	96	82	56	96	102	98	110	78	84
98	108	96	98	102	114	104	78	150	110
108	114	88	96	108	102	114	78	110	116
100	104	120	94	70	100	86	88	88	96

A.1.1. Visual Validation

To perform a visual test whether the data generated by the simulation tool is distributed according the distribution we would expect, we divided the data into bins and we created a histogram in which we compare the expected densities with the observed densities according to the data generated by the simulation tool, as shown in Figure A.1. At first glance, we see that in general the pattern generated by the simulation tool resembles the expected pattern. However, for some bins we see a large difference between the expected density and the observed density according to the data generated by the simulation tool. Figure A.2 shows the same deviations when we look at the observed cumulative distribution function according to the generated data compared to the theoretical expectations. For some bins there are deviations, for which other bins compensate again. To test whether these deviations are significant, we perform a chi-square goodness-of-fit test in Section A.1.2.

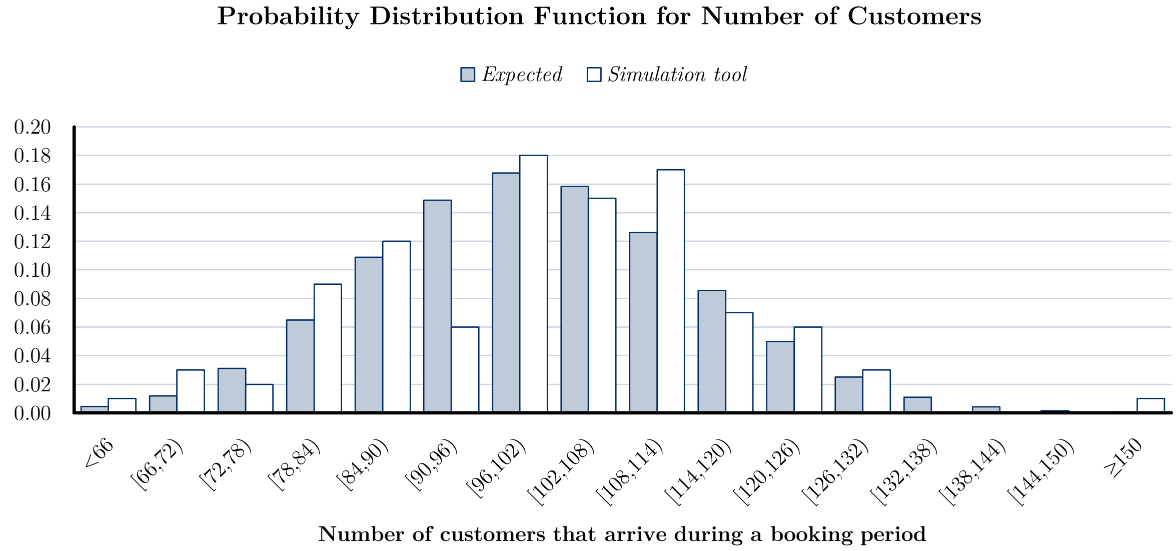


Figure A.1. *Observed probability density per bin for the simulation tool compared to expected probability density*

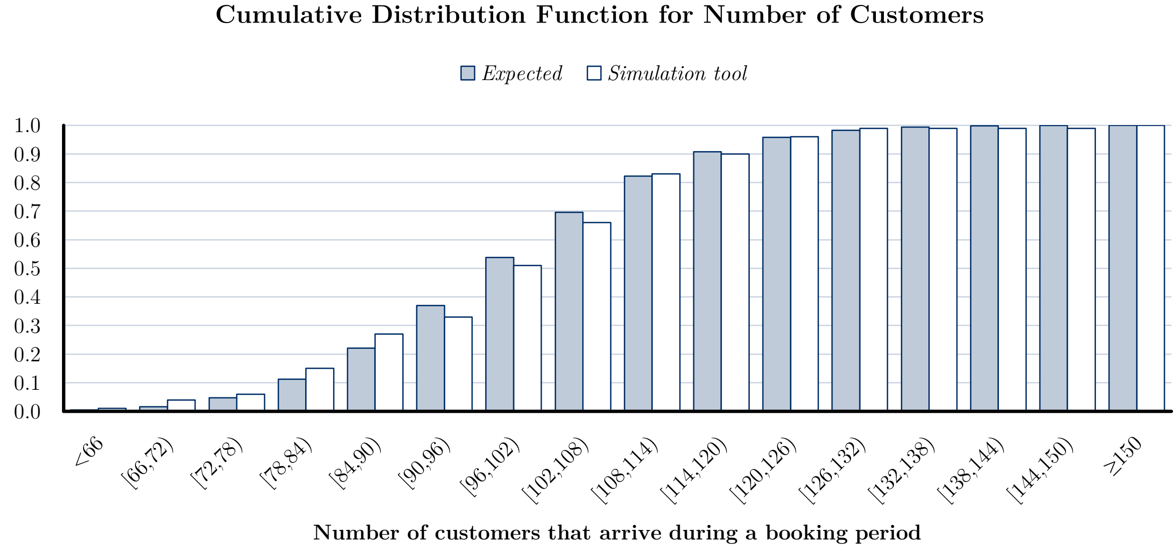


Figure A.2. *Observed cumulative probability per bin for the simulation tool compared to theoretical expectations*

A.1.2. Chi-Square Goodness-of-Fit Test

As mentioned, for the chi-square goodness-of-fit test we make use of the method proposed by Law (2015). This method basically determines the bounds of each bin in such a way, that the expected frequency for each bin is equal. However, when we deal with discrete data types, as is our case, Law (2015) points out that this may not be possible. In that case we should try to set the bounds for each bin in such a way that the expected frequency is approximately equal. Table A.2 shows the bins that we set, the expected frequency for each bin, the observed frequency for each bin and the mean square error for each bin. The lower bounds for the bins are inclusive bounds, and the upper bounds are exclusive bounds. This is an important observation for determining the expected frequencies, because for discrete probability distributions, unlike continuous probability distributions, $\mathrm{P}(N) \leq n \neq \mathrm{P}(N) < n$. Therefore, we need to take into account that upper bounds are exclusive and lower bounds inclusive when we calculate the expected frequencies.

Table A.2. *Expected frequencies and observed frequencies used for the chi-square goodness-of-fit test*

Bin	Expected frequency	Observed frequency	Mean square error
[0, 78)	4.74	6	0.3367
[78, 84)	6.49	9	0.9690
[84, 88)	6.75	6	0.0835
[88, 90)	4.12	6	0.8530
[90, 92)	4.58	2	1.4555
[92, 94)	4.98	1	3.1819
[94, 96)	5.30	3	0.9975
[96, 98)	5.52	10	3.6363
[98, 100)	5.63	5	0.0710
[100, 102)	5.63	3	1.2304
[102, 104)	5.52	8	1.1119
[104, 106)	5.31	4	0.3230
[106, 108)	5.01	3	0.8058
[108, 110)	4.64	8	2.4369
[110, 112)	4.22	6	0.7545
[112, 114)	3.76	3	0.1553
[114, 118)	6.15	7	0.1177
[118, 122)	4.42	5	0.0753
[122, 128)	4.03	2	1.0238
[12 , ∞)	3.18	3	0.0107

To calculate the expected frequencies in a bin i we make use of Equation (A.2), in which λ stands for the total number of realizations, which equals 100 in our case because we perform 100 draws. λ_i stands for the expected number of realizations of N in bin i. Finally, UB_i and LB_i respectively represent the upper bound and the lower bound of bin i.

$$\mathrm{E}[\lambda_i] = \lambda \times \big(\mathrm{P}(N < UB_i) - \mathrm{P}(N \geq LB_i)\big) \tag{A.2}$$

Observe that for the first bin, we use a lower bound of 0, because it is impossible that a negative number of customers arrive during the booking period and therefore N will never be negative. For the last bin, we use infinity as the upper bound, which makes $\mathrm{P}(N < UB_i)$ equal to 1. The results of the calculations are displayed in the column for expected frequencies of Table A.2.

Following the procedure for the chi-square goodness-of-fit test as described by Law (2015), we calculate the mean square error for bin i by making use of Equation (A.3), in which O_i stands of the observed number of realizations in bin i.

$$\frac{(O_i - \mathrm{E}[\lambda_i])^2}{\mathrm{E}[\lambda_i]} \tag{A.3}$$

The test statistic is computed by taking the sum of the mean square error over all bins. Its value in this case turns out to be 19.6297. The critical value is taken from a chi-square distribution with 9 degrees of freedom and a confidence level of 5% is used: $\chi^2_{,\,0.\;5} = 30.1435 > 1\;.62\;7$. Therefore, with 95% confidence we do not reject the hypothesis that the data generated by our simulation tool is distributed according to the distribution described earlier ($N = 2 \times X$ where $X \sim$ Poisson(50)).

A.2. Empirical Customer Arrival Rates

In this section we validate the implementation of the empirical distribution of customer arrival times in our simulation tool. Just as for the distribution of the number of customers, we consider 100 replications. In each replication the number of customers equals the corresponding value from Table A.1. We generate arrival times according to the empirical distribution for all the customers in a replication for each of the 100 replications. After obtaining all data, we validate the output of our simulation tool against the theoretical expectations. First, we carry out a quick visual test. After that, we perform a chi-square goodness-of-fit test.

A.2.1. Visual Validation

The first step for a visual validation of the realizations generated by our simulation tool is dividing the data into bins of equal size. Our data consists of arrival times of customers expressed as percentages of the length of the booking period, which is 24 hours in our case. For each replication we divided the arrival times over 24 bins with a length of one hour. The observed frequencies we divided by the number of customers that arrived in total in the corresponding replication to obtain the probability density per bin. The next step consists of taking the average of the obtained densities over all replication for each bin, so we can construct the average probability density function according to the data generated by our simulation tool. As we deal with continuous data (describing time) here, we speak of a probability density function and we display this function as a line in Figure A.3. We can see that the function generated by our simulation tool matches quite well with the theoretical expectations with regard to the probability density function. Figure A.4 shows the same in terms of the cumulative distribution function.

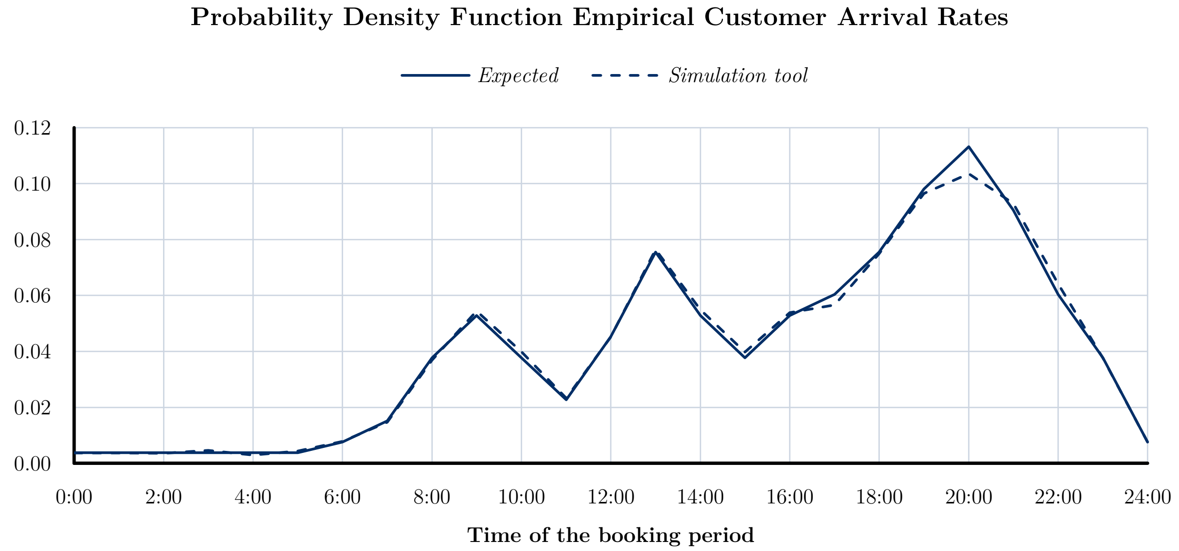


Figure A.3. *Observed probability density function for the simulation tool compared to expected probability density*

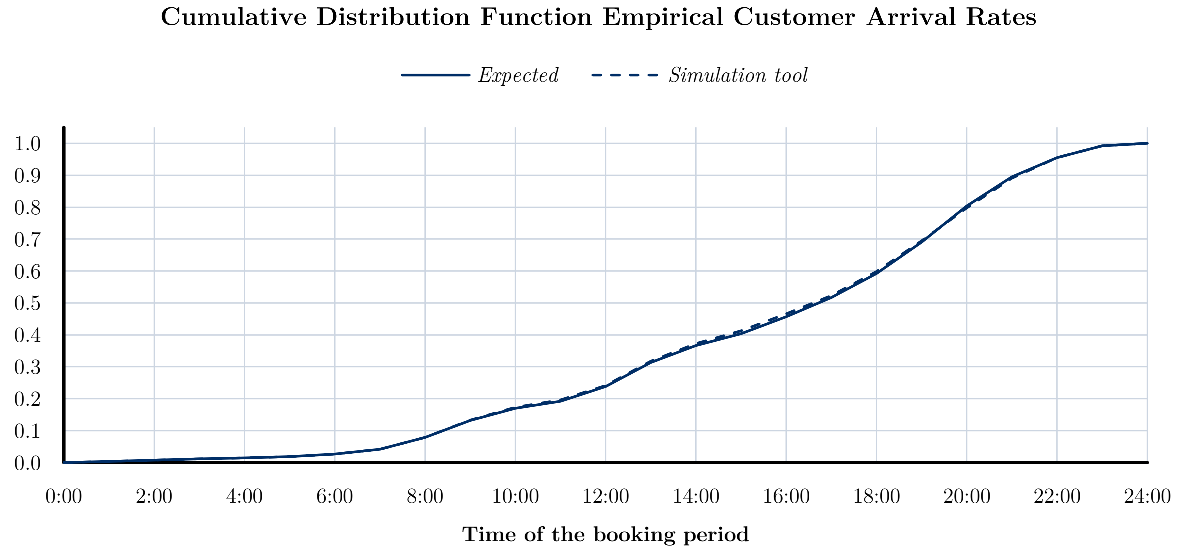


Figure A.4. *Observed cumulative probability function for the simulation tool compared to theoretical expectations*

A.2.2. Chi-Square Goodness-of-Fit Test

We make use of the same approach to carry out the chi-square goodness-of-fit test as in Appendix A.1. We changed the bounds for the 24 bins formed before in such a way that the expected frequency for each bin is equal. As we deal with an empirical distribution, this is somewhat more complex because we do not have a formula that defines the cumulative distribution function. We only have the cumulative probability defined for each hour of the day, as displayed in Figure A.4. We therefore first determine the density for a bin, which equals 100% divided by 24, the number of bins. We then can accumulate the density over the bins, so we obtain the desired cumulative probability at the upper bound of the bin. The upper bound can then be determined by comparing this cumulative probability to the values of the cumulative distribution function that are already available. By making use of linear interpolation we retrieve the upper bound for a bin. The lower bound for a bin equals the upper bound of the previous bin (or 0 for the first bin), because we deal with a continuous distribution. In that case it does not matter whether the upper and lower bounds of the bins are inclusive or exclusive.

We then compare the expected frequency in each bin to the average observed frequency in each bin. The average is again determined over the arrival times for the first 100 replications. The average number of customers that arrived over the first 100 replications equals 99.38, so the expected frequencies are equal to 99.38/24 which equals 4.14. Our test statistic is again computed by taking the sum of the mean square error (Equation (A.3)) over all bins. Its value in this case turns out to be 0.2426, as follows from Table A.3. This is much lower than the critical value from a chi-square distribution with 23 degrees of freedom and a confidence level of 5%: $\chi^2_{23,\ 0.\ 5} = 35.1725 > 0.2426$. Based on the test results, we can state with confidence that the data generated by our simulation tool on average indeed is distributed according to the empirical distribution that we would expect.

Table A.3. *Expected frequencies and observed frequencies used for the chi-square goodness-of-fit test*

Bin	Expected frequency	Observed frequency	Mean square error
0:00-7:00	4.14	4.15	0.00002
7:00-8:04	4.14	4.17	0.00021
8:04-8:51	4.14	4.15	0.00002
8:51-9:55	4.14	4.28	0.00468
9:55-11:21	4.14	4.33	0.00864
11:21-12:09	4.14	3.94	0.00974
12:09-12:42	4.14	4.35	0.01057
12:42-13:22	4.14	4.15	0.00002
13:22-14:14	4.14	4.50	0.03115
14:14-15:14	4.14	4.23	0.00192
15:14-16:01	4.14	4.27	0.00403
16:01-16:43	4.14	4.00	0.00479
16:43-17:19	4.14	3.74	0.03880
17:19-17:52	4.14	4.27	0.00403
17:52-18:19	4.14	3.95	0.00879
18:19-18:45	4.14	4.18	0.00037
18:45-19:09	4.14	3.90	0.01401
19:09-19:31	4.14	3.63	0.06302
19:31-19:53	4.14	4.05	0.00199
19:53-20:19	4.14	3.94	0.00974
20:19-20:47	4.14	4.33	0.00864
20:47-21:22	4.14	4.32	0.00775
21:22-22:05	4.14	4.32	0.00775
22:05-24:00	4.14	4.23	0.00192

A.3. Increasing Customer Arrival Rates

In this section we validate the implementation of the distribution of the spread of customer arrival times with increasing customer arrival rates over the booking period. We follow the same procedure as described in Section A.2.

A.3.1. Visual Validation

In the same way as we did for the empirical distribution in Section A.2, we again compare the probability density function and the cumulative distribution function as generated by our simulation tool over 100 replications to the expected functions according to the theoretical distribution. The theoretical function used for the cumulative distribution function equals $\left({}^{x}\!/_{24}\right)^{2} = {}^{x^2}\!/_{576}$. We consider this function on the interval of $0 \leq x \leq 24$ in which x equals the time in hours. The probability density function can then be derived and equals ${}^{x}\!/_{2}$. As we see in Figure A.5 and Figure A.6, our simulation tool seems to do quite a good job when considering the average over the first 100 replications.

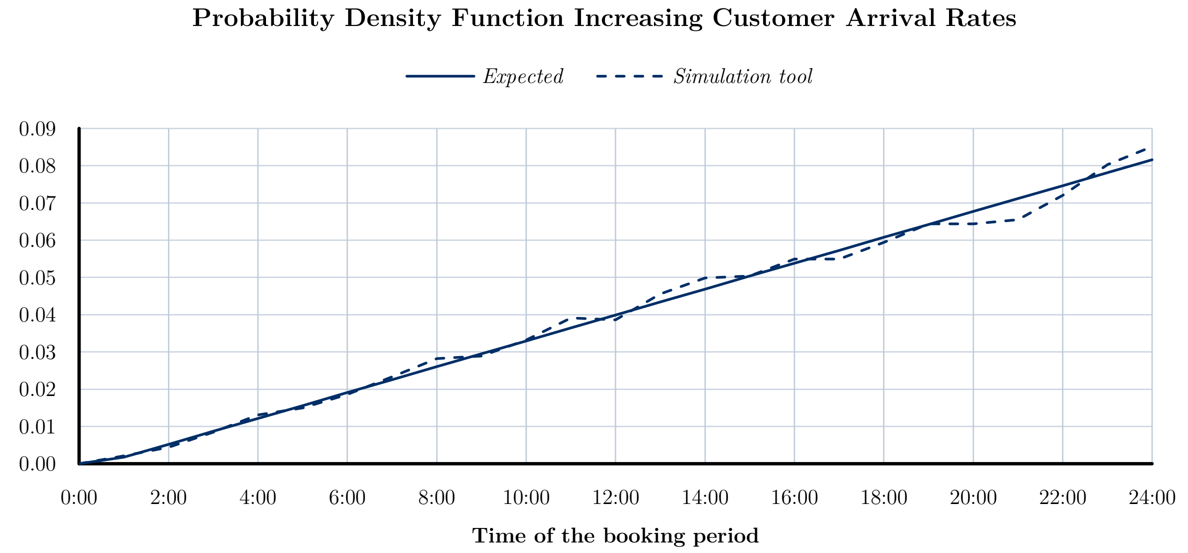


Figure A.5. *Observed probability density function for the simulation tool compared to expected probability density*

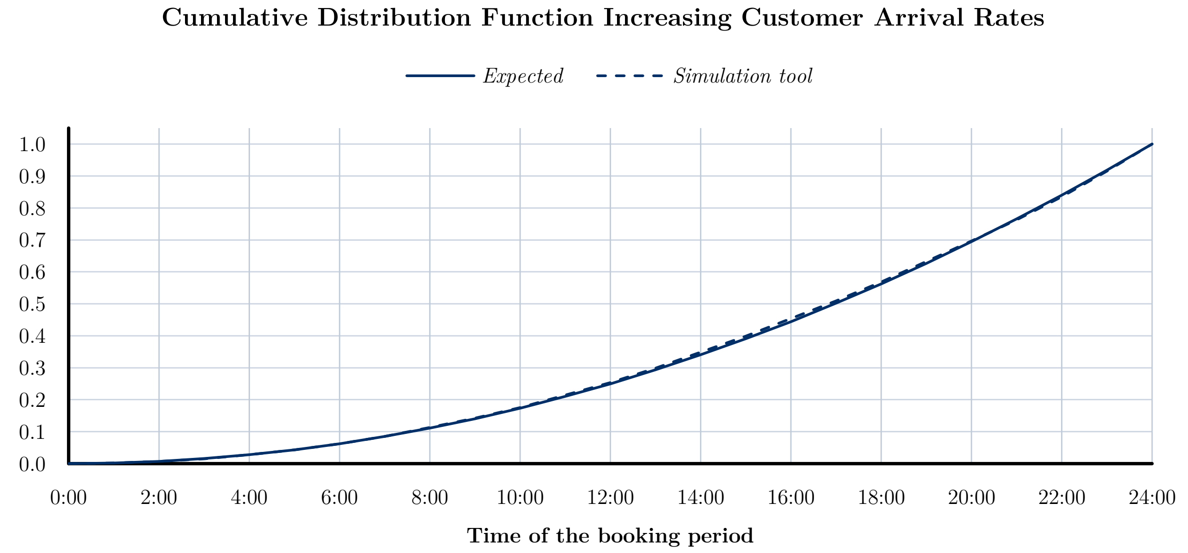


Figure A.6. *Observed cumulative probability function for the simulation tool compared to theoretical expectations*

A.3.2. Chi-Square Goodness-of-Fit Test

Just as we did in Section A.2, we again determine the bounds of the bins in such a way that the expected frequency is equal to 99.38/24 for each of the 24 bins. We determine the cumulative density for each bin in the same way as before. Given this cumulative density y_i we can determine the upper bound in hours (x_i) for a bin i by calculating the inverse of the cumulative distribution function: $x_i = \sqrt{576 y_i}$. The results of the test are displayed in Table A.4. Our test statistic sums up to 0.2720 in this case. The critical value we use here is the same as in Section A.2, because we have the same number of bins and we use the same confidence level: $\chi^2_{23,\ 0.\ 5} = 35.1725 > 0.2720$. This confirms that we cannot reject the hypothesis that the data generated by our simulation tool is distributed according to the expected distribution.

Table A.4. *Expected frequencies and observed frequencies used for the chi-square goodness-of-fit test*

Bin	Expected frequency	Observed frequency	Mean square error
0:00-4:53	4.14	4.13	0.00003
4:53-6:55	4.14	4.18	0.00037
6:55-8:29	4.14	4.23	0.00192
8:29-9:47	4.14	4.33	0.00864
9:47-10:57	4.14	4.28	0.00468
10:57-12:00	4.14	4.00	0.00479
12:00-12:57	4.14	4.40	0.01622
12:57-13:51	4.14	4.38	0.01381
13:51-14:41	4.14	3.94	0.00974
14:41-15:29	4.14	4.49	0.02944
15:29-16:14	4.14	4.08	0.00089
16:14-16:58	4.14	4.02	0.00353
16:58-17:39	4.14	3.85	0.02043
17:39-18:19	4.14	4.16	0.00009
18:19-18:58	4.14	4.24	0.00237
18:58-19:35	4.14	3.91	0.01287
19:35-20:11	4.14	3.91	0.01287
20:11-20:47	4.14	3.82	0.02486
20:47-21:21	4.14	3.71	0.04483
21:21-21:54	4.14	4.29	0.00537
21:54-22:26	4.14	4.24	0.00237
22:26-22:58	4.14	4.17	0.00021
22:58-23:29	4.14	4.59	0.04872
23:29-23:59	4.14	4.03	0.00297

A.4. Uniform Customer Arrival Rates

In this section we validate the implementation of a uniform distribution of customer arrival times over the booking period. We follow the same procedure as described in Section A.2.

A.4.1. Visual Validation

Just as described in Section A.2, we first compare the probability density function as observed from the data generated by our simulation tool with the expected probability density function according to the theoretical distribution. The theoretical cumulative probability function is given by $^{x}/_{24}$, resulting in a probability density function that equals $^{1}/_{24}$. Figure A.7 shows the result after 100 replications, which shows some distortions of the density function generated by the simulation tool compared to the expected density function. The distortions are quite small however, and we see in Figure A.8 that the cumulative distribution generated by the simulation tool over 100 replications hardly shows any deviations from the expected cumulative distribution function according to the uniform distribution. These results support the assumption that the data generated by the simulation tool indeed has the same distribution as expected. To put this to a statistical test, we perform a chi-square goodness-of-fit test in the next section.

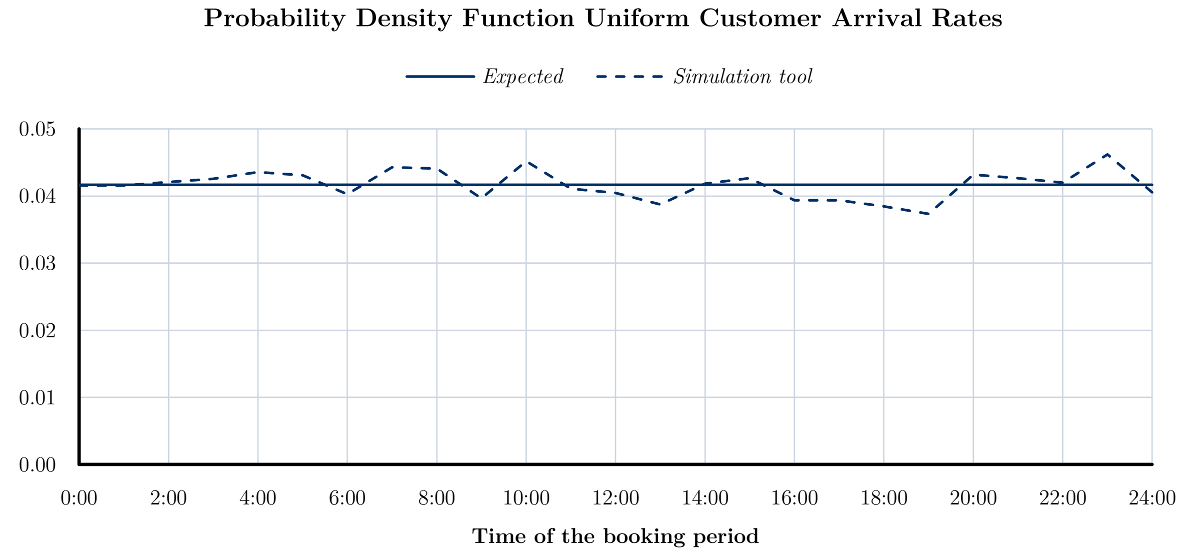


Figure A.7. *Observed probability density function for the simulation tool compared to expected probability density*

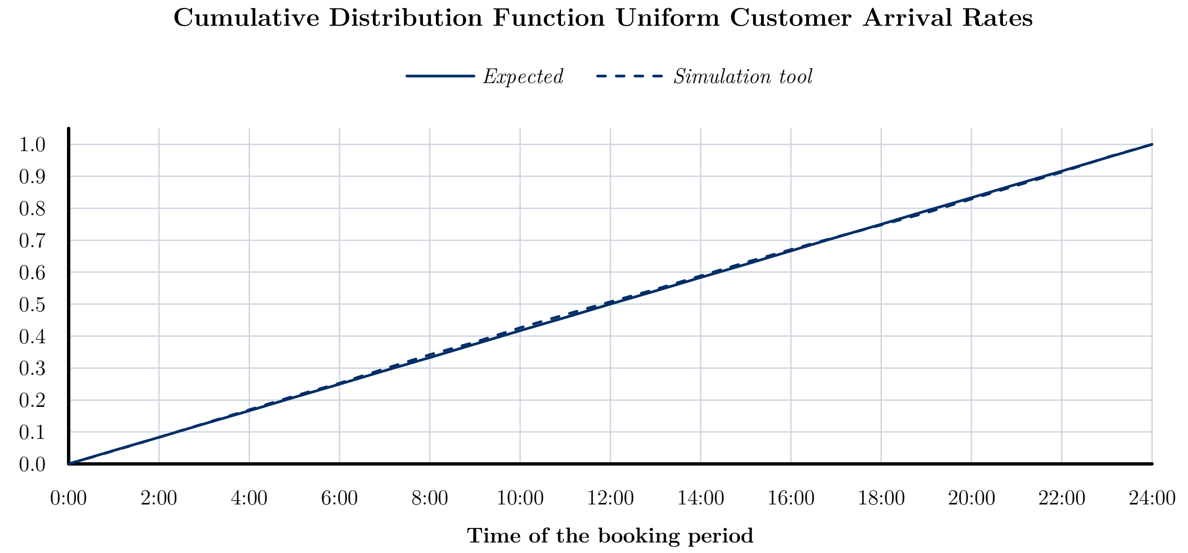


Figure A.8. *Observed cumulative probability function for the simulation tool compared to theoretical expectations*

A.4.2. Chi-Square Goodness-of-Fit Test

The chi-square goodness-of-fit test was carried out in the same way as before. The expected frequencies for each bin are based on a total average of 99.38 customers that arrive during the booking period. As we deal with a uniform distribution, to obtain an equal expected frequency for each bin it suffices to have bins of equal length. We therefore again use 24 bins, each with a length of one hour. The results of the chi-square goodness-of-fit test are displayed in Table A.5.

Our test statistic is again computed by taking the sum of the mean square error (Equation (A.3)) over all bins. Its value in this case turns out to be 0.2720. This is much lower than the critical value from a chi-square distribution with 23 degrees of freedom and a confidence level of 5%: $\chi^2_{23,\ 0.\ 5} = 35.1725 > 0.2720$. Based on the test results, we can state with confidence that the data generated by our simulation tool on average indeed is distributed according to the uniform distribution that we would expect.

Table A.5. *Expected frequencies and observed frequencies used for the chi-square goodness-of-fit test*

Bin	Expected frequency	Observed frequency	Mean square error
0:00-1:00	4.14	4.13	0.00003
1:00-2:00	4.14	4.18	0.00037
2:00-3:00	4.14	4.23	0.00192
3:00-4:00	4.14	4.33	0.00864
4:00-5:00	4.14	4.28	0.00468
5:00-6:00	4.14	4.00	0.00479
6:00-7:00	4.14	4.40	0.01622
7:00-8:00	4.14	4.38	0.01381
8:00-9:00	4.14	3.94	0.00974
9:00-10:00	4.14	4.49	0.02944
10:00-11:00	4.14	4.08	0.00089
11:00-12:00	4.14	4.02	0.00353
12:00-13:00	4.14	3.85	0.02043
13:00-14:00	4.14	4.16	0.00009
14:00-15:00	4.14	4.24	0.00237
15:00-16:00	4.14	3.91	0.01287
16:00-17:00	4.14	3.91	0.01287
17:00-18:00	4.14	3.82	0.02486
18:00-19:00	4.14	3.71	0.04483
19:00-20:00	4.14	4.29	0.00537
20:00-21:00	4.14	4.24	0.00237
21:00-22:00	4.14	4.17	0.00021
22:00-23:00	4.14	4.59	0.04872
23:00-24:00	4.14	4.03	0.00297

Appendix B. Computational Results

In this appendix we provide additional results which were not included in the main report. The results are categorized based on the hypotheses as defined in Chapter 3.

B.1. Hypothesis 1

Table B.1 presents the results of the comparisons we did for Hypothesis 1. Whenever we distinguish a better performance for one of the strategies on a certain KPI, we show the result for that strategy and KPI in bold. For the KPIs where it is not straightforward whether it is better to have a higher or a lower score, we leave it to the reader to judge which result is best.

Table B.1. *Results Hypothesis 1 per scenario and strategy*

	Scenario 1a		**Scenario 1b**		**Scenario 2a**		**Scenario 2b**	
KPI	*MYS*	*OBS*	*MYS*	*OBS*	*MYS*	*OBS*	*MYS*	*OBS*
Customers served (% of total)	0.5964	**0.6072**	**0.6333**	0.6062	0.6772	**0.6927**	**0.6479**	0.5896
Costs per customer (€)	21.67	**21.36**	**21.06**	21.50	25.88	**25.31**	**26.71**	28.23
Total driving time (seconds)	**57655**	58390	62870	**61263**	**106184**	106201	102458	**98284**
Total duration (seconds)	**98515**	99736	104726	**101625**	**145490**	146179	140684	**133462**
Total distance (km)	**901.92**	907.88	957.70	**945.16**	1850.01	**1842.07**	1817.14	**1738.38**
Total costs (€)	**1407.69**	1415.66	1453.35	**1425.12**	**1914.29**	1916.52	1881.01	**1815.93**
Number of delivery routes	8	8	8	7.9	8	8	8	7.9
Number vehicles used of type 1	3.6	4	5	4	3.6	4	5.6	4
Number vehicles used of type 2	4.4	4	3	3.9	4.4	4	2.4	3.9
Average utilization (%)	0.9868	0.9872	0.9828	0.9580	0.9584	0.9612	0.9588	0.8836
Number of customers served	65.2	**66.4**	**69.4**	66.6	74.2	**75.9**	**70.7**	64.6
Time windows offered	2.36	**2.41**	**2.50**	2.40	2.49	**2.55**	**2.37**	2.19
Response time (seconds)	0.3673	**0.1364**	0.7212	**0.1397**	0.8505	**0.1947**	1.4088	**0.1783**
Running time (seconds)	348	**320**	401	**324**	412	**358**	471	**309**
Schedule update time (seconds)	**0.1284**	0.1390	0.1573	**0.1274**	**0.1468**	0.1733	0.1583	**0.1373**
Quick optimize time (seconds)	46	**45**	48	**46**	**46**	49	46	**42**
Final optimize time (seconds)	57	**56**	59	59	**62**	67	58	**56**
Change fleet time (seconds)	0.4973	**0**	1.6097	**0**	2.2228	**0**	3.7944	**0**

Table B.2 presents the data that we used for the paired-t approach to compare the percentage of customers that are served for both MYS and OBS.

Table B.2. *Data used for comparing the percentage of customers served for MYS and OBS*

	Scenario 1a			**Scenario 1b**			**Scenario 2a**			**Scenario 2b**		
	MYS	*OBS*	***Diff***	*MYS*	*OBS*	***Diff***	*MYS*	*OBS*	***Diff***	*MYS*	*OBS*	***Diff***
Replication 1	50.00%	54.31%	**-4.31%**	61.21%	60.34%	**0.86%**	63.79%	63.79%	**0.00%**	57.76%	54.31%	**3.45%**
Replication 2	58.82%	59.80%	**-0.98%**	65.69%	62.75%	**2.94%**	67.65%	76.47%	**-8.82%**	72.55%	66.67%	**5.88%**
Replication 3	49.21%	50.79%	**-1.59%**	53.97%	52.38%	**1.59%**	64.29%	65.08%	**-0.79%**	58.73%	56.35%	**2.38%**
Replication 4	64.04%	64.91%	**-0.88%**	64.04%	63.16%	**0.88%**	57.89%	64.04%	**-6.14%**	55.26%	49.12%	**6.14%**
Replication 5	55.00%	53.33%	**1.67%**	60.00%	61.67%	**-1.67%**	63.33%	65.00%	**-1.67%**	56.67%	57.50%	**-0.83%**
Replication 6	62.50%	64.29%	**-1.79%**	75.00%	70.54%	**4.46%**	69.64%	69.64%	**0.00%**	59.82%	58.04%	**1.79%**
Replication 7	60.58%	64.42%	**-3.85%**	63.46%	57.69%	**5.77%**	72.12%	73.08%	**-0.96%**	71.15%	58.65%	**12.50%**
Replication 8	70.41%	70.41%	**0.00%**	71.43%	57.14%	**14.29%**	69.39%	69.39%	**0.00%**	71.43%	56.12%	**15.31%**
Replication 9	64.81%	62.96%	**1.85%**	55.56%	55.56%	**0.00%**	74.07%	72.22%	**1.85%**	68.52%	63.89%	**4.63%**
Replication 10	61.00%	62.00%	**-1.00%**	63.00%	65.00%	**-2.00%**	75.00%	74.00%	**1.00%**	76.00%	69.00%	**7.00%**
Mean	59.64%	60.72%	**-1.09%**	63.33%	60.62%	**2.71%**	67.72%	69.27%	**-1.55%**	64.79%	58.96%	**5.82%**
Variance	0.44%	0.38%	**0.04%**	0.41%	0.27%	**0.23%**	0.29%	0.21%	**0.11%**	0.61%	0.36%	**0.24%**
n	10	10	**10**	10	10	**10**	10	10	**10**	10	10	**10**
α	0.05	0.05	**0.05**	0.05	0.05	**0.05**	0.05	0.05	**0.05**	0.05	0.05	**0.05**
t-value	2.26	2.26	**2.26**	2.26	2.26	**2.26**	2.26	2.26	**2.26**	2.26	2.26	**2.26**
Lower bound CI	54.89%	56.33%	**-2.52%**	58.74%	56.90%	**-0.69%**	63.86%	65.98%	**-3.94%**	59.19%	54.69%	**2.32%**
Upper bound CI	64.39%	65.12%	**0.35%**	67.93%	64.35%	**6.11%**	71.58%	72.56%	**0.83%**	70.39%	63.24%	**9.32%**

Table B.3 presents the data that we used for the paired-t approach to compare the average delivery costs per customer served for both MYS and OBS.

Table B.3. *Data used for comparing the costs per customer served for MYS and OBS*

	Scenario 1a			**Scenario 1b**			**Scenario 2a**			**Scenario 2b**		
	MYS	*OBS*	***Diff***	*MYS*	*OBS*	***Diff***	*MYS*	*OBS*	***Diff***	*MYS*	*OBS*	***Diff***
Replication 1	€23. 4	€22.21	**€1.73**	€20.15	€20.73	**-€0.58**	€25.41	€25.41	**€0.00**	€2 . 1	€2 .73	**€0.08**
Replication 2	€23.32	€22. 7	**€0.45**	€22.27	€23.0	**-€0.81**	€26.37	€24.	**€1.39**	€25.35	€27.04	**-€1.69**
Replication 3	€21.54	€20. 0	**€0.64**	€22.11	€22.41	**-€0.30**	€23. 3	€23.22	**€0.61**	€24.	€26.17	**-€1.28**
Replication 4	€1 . 7	€20.11	**-€0.14**	€20.15	€20.0	**€0.07**	€2 .31	€26.76	**€1.55**	€30.22	€31. 4	**-€1.62**
Replication 5	€21. 5	€21.	**-€0.14**	€20.74	€20.72	**€0.02**	€26.3	€24.50	**€1.89**	€27. 4	€27. 2	**€0.12**
Replication 6	€21.1	€21.1	**-€0.01**	€1 .12	€1 .10	**-€0.98**	€25.6	€25.27	**€0.42**	€2 .06	€2 .00	**-€0.94**
Replication 7	€22.0	€20. 4	**€1.25**	€21.40	€22.65	**-€1.25**	€25.47	€24.	**€0.49**	€24.55	€2 .1	**-€3.63**
Replication 8	€21.36	€21.6	**-€0.32**	€20.26	€21.56	**-€1.30**	€27. 5	€2 .01	**-€0.16**	€25. 6	€2 .44	**-€3.48**
Replication 9	€1 .56	€1 . 0	**-€0.24**	€22. 2	€22. 4	**-€0.02**	€23.70	€23. 1	**-€0.11**	€25.64	€27.42	**-€1.78**
Replication 10	€21.90	€22.03	**-€0.13**	€22.55	€21. 4	**€0.71**	€25. 1	€26.17	**-€0.36**	€24.72	€25.6	**-€0.96**
Mean	€21.67	€21.36	**€0.31**	€21.06	€21.50	**-€0.44**	€25.	€25.31	**€0.57**	€26.71	€2 .23	**-€1.52**
Variance	€1.76	€0. 4	**€0.49**	€2.12	€1.73	**€0.43**	€2.1	€1. 6	**€0.62**	€4.53	€3.37	**€1.60**
n	10	10	**10**	10	10	**10**	10	10	**10**	10	10	**10**
α	0.05	0.05	**0.05**	0.05	0.05	**0.05**	0.05	0.05	**0.05**	0.05	0.05	**0.05**
t-value	2.26	2.26	**2.26**	2.26	2.26	**2.26**	2.26	2.26	**2.26**	2.26	2.26	**2.26**
Lower bound CI	€20.72	€20.67	**-€0.19**	€20.02	€20.56	**-€0.91**	€24. 3	€24.31	**€0.01**	€25.1	€26. 2	**-€2.42**
Upper bound CI	€22.62	€22.06	**€0.81**	€22.10	€22.44	**€0.02**	€26. 4	€26.31	**€1.14**	€2 .24	€2 .55	**-€0.61**

B.2. Hypothesis 2

In this section we present the results for Hypothesis 2, split up in Use Case 2 and Use Case 4.

B.2.1. Use Case 2

Table B.4 and Table B.5 present the results of the comparisons we did for Hypothesis 2 and Use Case 2. Whenever we distinguish a better performance for one of the strategies on a certain KPI, we show the result for that strategy and KPI in bold. For the KPIs where it is not straightforward whether it is better to have a higher or a lower score, we leave it to the reader to judge which result is best.

Table B.4. *Results Hypothesis 2 per scenario and strategy for Use Case 2*

	Scenario 1a			**Scenario 1b**		
KPI	*BAS*	*MYS*	*OBS*	*BAS*	*MYS*	*OBS*
Customers served (% of total)	**0.6876**	0.5964	0.6072	**0.6964**	0.6333	0.6062
Costs per customer (€)	**19.34**	21.67	21.36	**19.37**	21.06	21.50
Total driving time (seconds)	61218	**57655**	58390	64253	62870	**61263**
Total duration (seconds)	105774	**98515**	99736	108449	104726	**101625**
Total distance (km)	921.27	**901.92**	907.88	952.31	957.70	**945.16**
Total costs (€)	1451.89	**1407.69**	1415.66	1472.96	1453.35	**1425.12**
Number of delivery routes	8	8	8	8	8	7.9
Number vehicles used of type 1	3.8	3.6	4	4.8	5	4
Number vehicles used of type 2	4.2	4.4	4	3.2	3	3.9
Average utilization (%)	0.9778	0.9868	0.9872	0.9672	0.9828	0.9580
Number of customers served	**75.2**	65.2	66.4	**76.3**	69.4	66.6
Time windows offered	**2.73**	2.36	2.41	**2.77**	2.50	2.40
Response time (seconds)	0.2374	0.3673	**0.1364**	0.9833	0.7212	**0.1397**
Running time (seconds)	363	348	**320**	445	401	**324**
Schedule update time (seconds)	0.1420	**0.1284**	0.1390	**0.1260**	0.1573	0.1274
Quick optimize time (seconds)	49	46	**45**	49	48	**46**
Final optimize time (seconds)	68	57	**56**	69	**59**	**59**
Change fleet time (seconds)	9.6602	0.4973	**0**	32.7721	1.6097	**0**

Table B.5. *Results Hypothesis 2 per scenario and strategy for Use Case 2 (continued)*

	Scenario 2a			**Scenario 2b**		
KPI	*BAS*	*MYS*	*OBS*	*BAS*	*MYS*	*OBS*
Customers served (% of total)	**0.7241**	0.6772	0.6927	**0.7022**	0.6479	0.5896
Costs per customer (€)	**24.51**	25.88	25.31	**24.92**	26.71	28.23
Total driving time (seconds)	108665	**106184**	106201	105229	102458	**98284**
Total duration (seconds)	149369	**145490**	146179	145255	140684	**133462**
Total distance (km)	1876.21	1850.01	**1842.07**	1837.58	1817.14	**1738.38**
Total costs (€)	1941.07	**1914.29**	1916.52	1910.49	1881.01	**1815.93**
Number of delivery routes	8	8	8	8	8	7.9
Number vehicles used of type 1	4	3.6	4	5.3	5.6	4
Number vehicles used of type 2	4	4.4	4	2.7	2.4	3.9
Average utilization (%)	0.9271	0.9584	0.9612	0.9283	0.9588	0.8836
Number of customers served	**79.4**	74.2	75.9	**76.8**	70.7	64.6
Time windows offered	**2.70**	2.49	2.55	**2.56**	2.37	2.19
Response time (seconds)	0.2025	0.8505	**0.1947**	0.8287	1.4088	**0.1783**
Running time (seconds)	375	412	**358**	436	471	**309**
Schedule update time (seconds)	**0.1436**	0.1468	0.1733	0.1506	0.1583	**0.1373**
Quick optimize time (seconds)	50	**46**	49	50	46	**42**
Final optimize time (seconds)	76	**62**	67	69	58	**56**
Change fleet time (seconds)	0.0135	2.2228	**0**	36.9009	3.7944	**0**

Table B.6 presents the data that we used for the paired-t approach to compare the percentage of customers that are served for both BAS and MYS.

Table B.6. *Data used for comparing the percentage of customers served for BAS and MYS for Use Case 2*

	Scenario 1a			**Scenario 1b**			**Scenario 2a**			**Scenario 2b**		
	BAS	*MYS*	***Diff***	*BAS*	*MYS*	***Diff***	*BAS*	*MYS*	***Diff***	*BAS*	*MYS*	***Diff***
Replication 1	62.93%	50.00%	**12.93%**	68.10%	61.21%	**6.90%**	69.83%	63.79%	**6.03%**	63.79%	57.76%	**6.03%**
Replication 2	71.57%	58.82%	**12.75%**	70.59%	65.69%	**4.90%**	77.45%	67.65%	**9.80%**	77.45%	72.55%	**4.90%**
Replication 3	55.56%	49.21%	**6.35%**	61.11%	53.97%	**7.14%**	65.87%	64.29%	**1.59%**	63.49%	58.73%	**4.76%**
Replication 4	74.56%	64.04%	**10.53%**	70.18%	64.04%	**6.14%**	67.54%	57.89%	**9.65%**	62.28%	55.26%	**7.02%**
Replication 5	61.67%	55.00%	**6.67%**	65.83%	60.00%	**5.83%**	71.67%	63.33%	**8.33%**	67.50%	56.67%	**10.83%**
Replication 6	70.54%	62.50%	**8.04%**	77.68%	75.00%	**2.68%**	70.54%	69.64%	**0.89%**	66.07%	59.82%	**6.25%**
Replication 7	69.23%	60.58%	**8.65%**	65.38%	63.46%	**1.92%**	74.04%	72.12%	**1.92%**	69.23%	71.15%	**-1.92%**
Replication 8	77.55%	70.41%	**7.14%**	76.53%	71.43%	**5.10%**	72.45%	69.39%	**3.06%**	78.57%	71.43%	**7.14%**
Replication 9	75.00%	64.81%	**10.19%**	62.96%	55.56%	**7.41%**	78.70%	74.07%	**4.63%**	76.85%	68.52%	**8.33%**
Replication 10	69.00%	61.00%	**8.00%**	78.00%	63.00%	**15.00%**	76.00%	75.00%	**1.00%**	77.00%	76.00%	**1.00%**
Mean	68.76%	59.64%	**9.12%**	69.64%	63.33%	**6.30%**	72.41%	67.72%	**4.69%**	70.22%	64.79%	**5.44%**
Variance	0.47%	0.44%	**0.06%**	0.37%	0.41%	**0.13%**	0.18%	0.29%	**0.13%**	0.43%	0.61%	**0.13%**
n	10	10	**10**	10	10	**10**	10	10	**10**	10	10	**10**
α	0.05	0.05	**0.05**	0.05	0.05	**0.05**	0.05	0.05	**0.05**	0.05	0.05	**0.05**
t-value	2.26	2.26	**2.26**	2.26	2.26	**2.26**	2.26	2.26	**2.26**	2.26	2.26	**2.26**
Lower bound CI	63.87%	54.89%	**7.42%**	65.27%	58.74%	**3.76%**	69.41%	63.86%	**2.15%**	65.53%	59.19%	**2.84%**
Upper bound CI	73.65%	64.39%	**10.83%**	74.00%	67.93%	**8.85%**	75.40%	71.58%	**7.24%**	74.92%	70.39%	**8.03%**

Table B.7 presents the data that we used for the paired-t approach to compare the average delivery costs per customer served for both BAS and MYS.

Table B.7. *Data used for comparing the costs per customer served for BAS and MYS for Use Case 2*

	Scenario 1a			**Scenario 1b**			**Scenario 2a**			**Scenario 2b**		
	BAS	*MYS*	***Diff***	*BAS*	*MYS*	***Diff***	*BAS*	*MYS*	***Diff***	*BAS*	*MYS*	***Diff***
Replication 1	€1 . 4	€23. 4	**-€4.00**	€1 .66	€20.15	**-€1.49**	€23.37	€25.41	**-€2.04**	€26.57	€2 . 1	**-€3.24**
Replication 2	€1 .64	€23.32	**-€3.68**	€20.53	€22.27	**-€1.74**	€24.56	€26.37	**-€1.81**	€23.5	€25.35	**-€1.76**
Replication 3	€1 .51	€21.54	**-€2.03**	€1 . 4	€22.11	**-€2.27**	€22.61	€23. 3	**-€1.22**	€23. 7	€24.	**-€0.92**
Replication 4	€1 .32	€1 . 7	**-€1.65**	€1 .1	€20.15	**-€0.96**	€25.5	€2 .31	**-€2.73**	€26.11	€30.22	**-€4.11**
Replication 5	€20.23	€21. 5	**-€1.62**	€1 .17	€20.74	**-€1.57**	€24.02	€26.3	**-€2.37**	€24.1	€27. 4	**-€3.76**
Replication 6	€19.39	€21.1	**-€1.79**	€17.10	€1 .12	**-€1.02**	€25.32	€25.6	**-€0.37**	€26.55	€2 .06	**-€1.51**
Replication 7	€1 . 0	€22.0	**-€2.29**	€21.16	€21.40	**-€0.24**	€25.20	€25.47	**-€0.27**	€25.57	€24.55	**€1.02**
Replication 8	€1 .62	€21.36	**-€1.74**	€1 .10	€20.26	**-€1.16**	€26.71	€27. 5	**-€1.14**	€24.70	€25. 6	**-€1.26**
Replication 9	€17.41	€1 .56	**-€2.15**	€20.13	€22. 2	**-€2.69**	€22.43	€23.70	**-€1.27**	€23. 6	€25.64	**-€1.68**
Replication 10	€1 .5	€21. 0	**-€2.32**	€1 . 6	€22.55	**-€3.69**	€25.2	€25. 1	**-€0.52**	€24.01	€24.72	**-€0.71**
Mean	€1 .34	€21.67	**-€2.33**	€1 .37	€21.06	**-€1.68**	€24.51	€25.	**-€1.37**	€24. 2	€26.71	**-€1.79**
Variance	€0.71	€1.76	**€0.70**	€1.27	€2.12	**€0.97**	€1. 1	€2.1	**€0.72**	€1.36	€4.53	**€2.40**
n	10	10	**10**	10	10	**10**	10	10	**10**	10	10	**10**
α	0.05	0.05	**0.05**	0.05	0.05	**0.05**	0.05	0.05	**0.05**	0.05	0.05	**0.05**
t-value	2.26	2.26	**2.26**	2.26	2.26	**2.26**	2.26	2.26	**2.26**	2.26	2.26	**2.26**
Lower bound CI	€1 .74	€20.72	**-€2.93**	€1 .57	€20.02	**-€2.39**	€23.52	€24. 3	**-€1.98**	€24.0	€25.1	**-€2.90**
Upper bound CI	€1 . 5	€22.62	**-€1.73**	€20.1	€22.10	**-€0.98**	€25.50	€26. 4	**-€0.77**	€25.76	€2 .24	**-€0.69**

B.2.2. Use Case 4

Table B.8 and Table B.9 present the results of the comparisons we did for Hypothesis 2 and Use Case 4. Whenever we distinguish a better performance for one of the strategies on a certain KPI, we show the result for that strategy and KPI in bold. For the KPIs where it is not straightforward whether it is better to have a higher or a lower score, we leave it to the reader to judge which result is best.

Table B.8. *Results Hypothesis 2 per scenario and strategy for Use Case 4*

	Scenario 1a			**Scenario 1b**		
KPI	*BAS*	*MYS*	*OBS*	*BAS*	*MYS*	*OBS*
Customers served (% of total)	**0.9068**	0.8568	0.6072	**0.9253**	0.8868	0.6062
Costs per customer (€)	**19.92**	21.19	21.36	**20.10**	21.12	21.50
Total driving time (seconds)	69903	71031	**58390**	76147	76855	**61263**
Total duration (seconds)	131919	131127	**99736**	137989	137347	**101625**
Total distance (km)	994.44	1039.20	**907.88**	1086.24	1134.02	**945.16**
Total costs (€)	1977.66	1984.76	**1415.66**	2034.67	2040.94	**1425.12**
Number of delivery routes	8	8	8	8	8	7.9
Number vehicles used of type 1	1.7	2.1	4	2.9	2.7	4
Number vehicles used of type 2	2.3	1.9	4	1.1	1.3	3.9
Number vehicles used of type 3	4	4	0	4	4	0
Average utilization (%)	0.9607	0.9723	0.9872	0.9436	0.9666	0.9580
Number of customers served	**99.5**	93.8	66.4	**101.6**	97.0	66.6
Time windows offered	**3.59**	3.34	2.41	**3.64**	3.42	2.40
Response time (seconds)	1.5658	6.0713	**0.1493**	2.3487	5.2489	**0.1003**
Running time (seconds)	600	982	**320**	668	865	**330**
Schedule update time (seconds)	0.1722	0.2019	**0.1653**	0.1628	0.1573	**0.1495**
Quick optimize time (seconds)	61	**45**	**45**	58	**43**	46
Final optimize time (seconds)	94	64	**55**	95	**49**	63
Change fleet time (seconds)	11.7177	36.0852	**0**	12.4940	46.9158	**0**

Table B.9. *Results Hypothesis 2 per scenario and strategy for Use Case 4 (continued)*

	Scenario 2a			**Scenario 2b**		
KPI	*BAS*	*MYS*	*OBS*	*BAS*	*MYS*	*OBS*
Customers served (% of total)	**0.8740**	0.8560	0.6927	**0.8678**	0.8612	0.5906
Costs per customer (€)	25.35	26.28	**25.31**	**24.55**	25.52	28.22
Total driving time (seconds)	116955	119377	**106201**	106304	113068	**98475**
Total duration (seconds)	167925	170347	**146179**	157304	164356	**133707**
Total distance (km)	1954.42	2027.42	**1842.07**	1768.16	1916.89	**1741.61**
Total costs (€)	2424.53	2460.47	**1916.52**	2328.85	2403.37	**1817.93**
Number of delivery routes	8	8	8	8	8	7.9
Number vehicles used of type 1	2.1	1.6	4	3.3	3.2	4
Number vehicles used of type 2	1.9	2.4	4	0.7	0.8	3.9
Number vehicles used of type 3	4	4	0	4	4	0
Average utilization (%)	0.8997	0.9313	0.9612	0.8913	0.9290	0.8852
Number of customers served	**95.9**	93.7	75.9	**95.1**	94.2	64.7
Time windows offered	**3.27**	3.10	2.55	**3.25**	3.13	2.19
Response time (seconds)	1.5681	11.3494	**0.1495**	3.3232	8.7187	**0.1862**
Running time (seconds)	641	1535	**342**	790	1254	**314**
Schedule update time (seconds)	**0.1507**	0.2051	0.1748	0.1566	0.1790	**0.1431**
Quick optimize time (seconds)	66	**41**	47	62	**41**	43
Final optimize time (seconds)	108	**53**	63	91	**52**	55
Change fleet time (seconds)	2.6599	67.7509	**0**	4.7538	64.7018	**0**

Table B.10 presents the data that we used for the paired-t approach to compare the percentage of customers that are served for both BAS and MYS.

Table B.10. *Data used for comparing the percentage of customers served for BAS and MYS for Use Case 4*

	Scenario 1a			**Scenario 1b**			**Scenario 2a**			**Scenario 2b**		
	BAS	*MYS*	***Diff***	*BAS*	*MYS*	***Diff***	*BAS*	*MYS*	***Diff***	*BAS*	*MYS*	***Diff***
Replication 1	84.48%	77.59%	**6.90%**	88.79%	81.90%	**6.90%**	84.48%	79.31%	**5.17%**	81.90%	86.21%	**-4.31%**
Replication 2	91.18%	85.29%	**5.88%**	89.22%	92.16%	**-2.94%**	89.22%	85.29%	**3.92%**	94.12%	93.14%	**0.98%**
Replication 3	83.33%	73.81%	**9.52%**	84.13%	69.84%	**14.29%**	83.33%	76.19%	**7.14%**	82.54%	75.40%	**7.14%**
Replication 4	92.98%	86.84%	**6.14%**	95.61%	93.86%	**1.75%**	81.58%	79.82%	**1.75%**	82.46%	79.82%	**2.63%**
Replication 5	86.67%	80.00%	**6.67%**	92.50%	84.17%	**8.33%**	85.83%	80.83%	**5.00%**	80.83%	78.33%	**2.50%**
Replication 6	95.54%	89.29%	**6.25%**	99.11%	98.21%	**0.89%**	84.82%	85.71%	**-0.89%**	83.93%	83.93%	**0.00%**
Replication 7	92.31%	88.46%	**3.85%**	94.23%	92.31%	**1.92%**	90.38%	93.27%	**-2.88%**	90.38%	93.27%	**-2.88%**
Replication 8	91.84%	92.86%	**-1.02%**	95.92%	98.98%	**-3.06%**	89.80%	92.86%	**-3.06%**	94.90%	93.88%	**1.02%**
Replication 9	93.52%	91.67%	**1.85%**	89.81%	82.41%	**7.41%**	93.52%	90.74%	**2.78%**	90.74%	85.19%	**5.56%**
Replication 10	95.00%	91.00%	**4.00%**	96.00%	93.00%	**3.00%**	91.00%	92.00%	**-1.00%**	86.00%	92.00%	**-6.00%**
Mean	90.68%	85.68%	**5.00%**	92.53%	88.68%	**3.85%**	87.40%	85.60%	**1.79%**	86.78%	86.12%	**0.66%**
Variance	0.19%	0.42%	**0.09%**	0.20%	0.81%	**0.29%**	0.15%	0.40%	**0.13%**	0.28%	0.46%	**0.17%**
n	10	10	**10**	10	10	**10**	10	10	**10**	10	10	**10**
α	0.05	0.05	**0.05**	0.05	0.05	**0.05**	0.05	0.05	**0.05**	0.05	0.05	**0.05**
t-value	2.26	2.26	**2.26**	2.26	2.26	**2.26**	2.26	2.26	**2.26**	2.26	2.26	**2.26**
Lower bound CI	87.59%	81.05%	**2.88%**	89.31%	82.25%	**-0.01%**	84.61%	81.06%	**-0.78%**	82.99%	81.25%	**-2.31%**
Upper bound CI	93.78%	90.31%	**7.12%**	95.75%	95.12%	**7.71%**	90.18%	90.15%	**4.36%**	90.57%	90.98%	**3.64%**

Table B.11 presents the data that we used for the paired-*t* approach to compare the average delivery costs per customer served for both BAS and MYS.

Table B.11. *Data used for comparing the costs per customer served for BAS and MYS for Use Case 4*

	Scenario 1a			**Scenario 1b**			**Scenario 2a**			**Scenario 2b**		
	BAS	*MYS*	***Diff***	*BAS*	*MYS*	***Diff***	*BAS*	*MYS*	***Diff***	*BAS*	*MYS*	***Diff***
Replication 1	€1 .	€21. 0	**-€2.02**	€1 .61	€20.73	**-€1.12**	€24.30	€24.70	**-€0.40**	€24.1	€25.66	**-€1.48**
Replication 2	€20.66	€22.27	**-€1.61**	€22.27	€22.21	**€0.06**	€26.06	€2 .54	**-€2.48**	€24.30	€25.33	**-€1.03**
Replication 3	€1 .67	€20.7	**-€2.11**	€20.26	€23.25	**-€2.99**	€22.7	€26.1	**-€3.41**	€21. 2	€24. 2	**-€3.10**
Replication 4	€1 .53	€20.56	**-€1.03**	€1 . 5	€1 . 0	**-€0.95**	€27.10	€26. 3	**€0.27**	€24.70	€25.1	**-€0.49**
Replication 5	€20.2	€21.16	**-€0.88**	€1 . 1	€20.66	**-€1.85**	€24.27	€26.5	**-€2.32**	€24.02	€26.6	**-€2.66**
Replication 6	€1 .20	€20.63	**-€1.43**	€1 .40	€1 . 7	**-€0.57**	€25.76	€25.12	**€0.64**	€24.52	€26.15	**-€1.63**
Replication 7	€20.46	€21.43	**-€0.97**	€21.25	€20. 5	**€0.40**	€25.4	€25.77	**-€0.29**	€24.73	€24.14	**€0.59**
Replication 8	€21.74	€21.	**-€0.14**	€20.31	€20. 2	**-€0.61**	€27.27	€26.62	**€0.65**	€25.2	€26.15	**-€0.86**
Replication 9	€1 .70	€1 .	**-€1.29**	€1 . 3	€22.14	**-€2.21**	€24.3	€25.2	**-€0.89**	€24.23	€25.50	**-€1.27**
Replication 10	€20.03	€21.26	**-€1.23**	€21.2	€21.65	**-€0.36**	€26.0	€27.20	**-€1.11**	€27.6	€25.44	**€2.25**
Mean	€1 . 2	€21.1	**-€1.27**	€20.10	€21.12	**-€1.02**	€25.35	€26.2	**-€0.93**	€24.55	€25.52	**-€0.97**
Variance	€0.	€0.50	**€0.33**	€1.54	€1.54	**€1.11**	€1.	€1.2	**€1.96**	€2.06	€0.51	**€2.36**
n	10	10	**10**	10	10	**10**	10	10	**10**	10	10	**10**
α	0.05	0.05	**0.05**	0.05	0.05	**0.05**	0.05	0.05	**0.05**	0.05	0.05	**0.05**
t-value	2.26	2.26	**2.26**	2.26	2.26	**2.26**	2.26	2.26	**2.26**	2.26	2.26	**2.26**
Lower bound CI	€1 .24	€20.6	**-€1.68**	€1 .21	€20.23	**-€1.77**	€24.34	€25.47	**-€1.93**	€23.52	€25.01	**-€2.07**
Upper bound CI	€20.5	€21.6	**-€0.86**	€20.	€22.01	**-€0.27**	€26.36	€27.10	**€0.07**	€25.57	€26.03	**€0.13**

B.3. Hypothesis 3

Table B.12 and Table B.13 present the results of the comparisons we did for Hypothesis 3. Whenever we distinguish a better performance for one of the strategies on a certain KPI, we show the result for that strategy and KPI in bold. For the KPIs where it is not straightforward whether it is better to have a higher or a lower score, we leave it to the reader to judge which result is best.

Table B.12. *Results Hypothesis 3 per scenario and strategy*

	Scenario 1a			**Scenario 1b**		
KPI	*BAS*	*OBS (Large)*	*OBS*	*BAS*	*OBS (Large)*	*OBS*
Customers served (% of total)	**0.9068**	0.8654	0.6072	**0.9253**	0.8872	0.6062
Costs per customer (€)	**19.92**	20.67	21.36	**20.10**	20.75	21.50
Total driving time (seconds)	69903	68011	**58390**	76147	74992	**61263**
Total duration (seconds)	131919	128533	**99736**	137989	135454	**101625**
Total distance (km)	994.44	976.85	**907.88**	1086.24	1089.48	**945.16**
Total costs (€)	1977.66	1953.29	**1415.66**	2034.67	2013.27	**1425.12**
Number of delivery routes	8	8	8	8	7.9	7.9
Number vehicles used of type 1	1.7	2	4	2.9	2	4
Number vehicles used of type 2	2.3	2	4	1.1	1.9	3.9
Number vehicles used of type 3	4	4	0	4	4	0
Average utilization (%)	0.9607	0.9806	0.9872	0.9436	0.9544	0.9580
Number of customers served	**99.5**	94.7	66.4	**101.6**	97.3	66.6
Time windows offered	**3.59**	3.44	2.41	**3.64**	3.49	2.40
Response time (seconds)	1.5658	0.1765	**0.1493**	2.3487	0.1953	**0.1003**
Running time (seconds)	600	459	**320**	668	526	**330**
Schedule update time (seconds)	0.1722	0.2017	**0.1653**	0.1628	0.2186	**0.1495**
Quick optimize time (seconds)	61	64	**45**	58	74	**46**
Final optimize time (seconds)	94	88	**55**	95	97	**63**
Change fleet time (seconds)	11.7177	**0**	**0**	12.4940	**0**	**0**

Table B.13. *Results Hypothesis 3 per scenario and strategy (continued)*

	Scenario 2a			Scenario 2b		
KPI	*BAS*	*OBS (Large)*	*OBS*	*BAS*	*OBS (Large)*	*OBS*
Customers served (% of total)	**0.8740**	0.8537	0.6927	**0.8678**	0.8304	0.5906
Costs per customer (€)	25.35	**25.27**	25.31	**24.55**	25.02	28.22
Total driving time (seconds)	116955	110411	**106201**	106304	105099	**98475**
Total duration (seconds)	167925	160817	**146179**	157304	154557	**133707**
Total distance (km)	1954.42	**1819.75**	1842.07	1768.16	1768.27	**1741.61**
Total costs (€)	2424.53	2358.31	**1916.52**	2328.85	2276.00	**1817.93**
Number of delivery routes	8	8	8	8	7.6	7.9
Number vehicles used of type 1	2.1	2	4	3.3	2	4
Number vehicles used of type 2	1.9	2	4	0.7	1.6	3.9
Number vehicles used of type 3	4	4	0	4	4	0
Average utilization (%)	0.8997	0.9158	0.9612	0.8913	0.8789	0.8852
Number of customers served	**95.9**	93.6	75.9	**95.1**	91.1	64.7
Time windows offered	**3.27**	3.20	2.55	**3.25**	3.13	2.19
Response time (seconds)	1.5681	0.1981	**0.1495**	3.3232	0.2247	**0.1862**
Running time (seconds)	641	545	**342**	790	494	**314**
Schedule update time (seconds)	**0.1507**	0.2031	0.1748	0.1566	0.1687	**0.1431**
Quick optimize time (seconds)	66	75	**47**	62	67	**43**
Final optimize time (seconds)	108	110	**63**	91	102	**55**
Change fleet time (seconds)	2.6599	**0**	**0**	4.7538	**0**	**0**

Table B.14 presents the data that we used for the paired-t approach to compare the percentage of customers that are served for both BAS and OBS (Large Initial Fleet).

Table B.14. *Data used for comparing the percentage of customers served for BAS and OBS (Large Initial Fleet)*

	Scenario 1a			Scenario 1b			Scenario 2a			Scenario 2b		
	BAS	*MYS*	***Diff***	*BAS*	*MYS*	***Diff***	*BAS*	*MYS*	***Diff***	*BAS*	*MYS*	***Diff***
Replication 1	84.48%	73.28%	**11.21%**	88.79%	82.76%	**6.03%**	84.48%	79.31%	**5.17%**	81.90%	80.17%	**1.72%**
Replication 2	91.18%	87.25%	**3.92%**	89.22%	94.12%	**-4.90%**	89.22%	84.31%	**4.90%**	94.12%	86.27%	**7.84%**
Replication 3	83.33%	75.40%	**7.94%**	84.13%	77.78%	**6.35%**	83.33%	81.75%	**1.59%**	82.54%	78.57%	**3.97%**
Replication 4	92.98%	89.47%	**3.51%**	95.61%	96.49%	**-0.88%**	81.58%	80.70%	**0.88%**	82.46%	78.07%	**4.39%**
Replication 5	86.67%	79.17%	**7.50%**	92.50%	84.17%	**8.33%**	85.83%	80.00%	**5.83%**	80.83%	79.17%	**1.67%**
Replication 6	95.54%	92.86%	**2.68%**	99.11%	99.11%	**0.00%**	84.82%	83.04%	**1.79%**	83.93%	80.36%	**3.57%**
Replication 7	92.31%	90.38%	**1.92%**	94.23%	88.46%	**5.77%**	90.38%	90.38%	**0.00%**	90.38%	87.50%	**2.88%**
Replication 8	91.84%	94.90%	**-3.06%**	95.92%	89.80%	**6.12%**	89.80%	87.76%	**2.04%**	94.90%	81.63%	**13.27%**
Replication 9	93.52%	91.67%	**1.85%**	89.81%	81.48%	**8.33%**	93.52%	94.44%	**-0.93%**	90.74%	91.67%	**-0.93%**
Replication 10	95.00%	91.00%	**4.00%**	96.00%	93.00%	**3.00%**	91.00%	92.00%	**-1.00%**	86.00%	87.00%	**-1.00%**
Mean	90.68%	86.54%	**4.15%**	92.53%	88.72%	**3.82%**	87.40%	85.37%	**2.03%**	86.78%	83.04%	**3.74%**
Variance	0.19%	0.59%	**0.16%**	0.20%	0.49%	**0.19%**	0.15%	0.29%	**0.06%**	0.28%	0.22%	**0.18%**
n	10	10	**10**	10	10	**10**	10	10	**10**	10	10	**10**
α	0.05	0.05	**0.05**	0.05	0.05	**0.05**	0.05	0.05	**0.05**	0.05	0.05	**0.05**
t-value	2.26	2.26	**2.26**	2.26	2.26	**2.26**	2.26	2.26	**2.26**	2.26	2.26	**2.26**
Lower bound CI	87.59%	81.03%	**1.32%**	89.31%	83.68%	**0.67%**	84.61%	81.49%	**0.24%**	82.99%	79.69%	**0.71%**
Upper bound CI	93.78%	92.05%	**6.97%**	95.75%	93.75%	**6.97%**	90.18%	89.24%	**3.82%**	90.57%	86.39%	**6.77%**

Table B.15 presents the data that we used for the paired-t approach to compare the average delivery costs per customer served for both BAS and OBS (Large Initial Fleet).

Table B.15. *Data used for comparing the costs per customer served for BAS and OBS (Large Initial Fleet)*

	Scenario 1a			**Scenario 1b**			**Scenario 2a**			**Scenario 2b**		
	BAS	*MYS*	***Diff***	*BAS*	*MYS*	***Diff***	*BAS*	*MYS*	***Diff***	*BAS*	*MYS*	***Diff***
Replication 1	€1 .61	€20.31	**-€0.70**	€24.30	€24.57	**-€0.27**	€24.1	€26.0	**-€1.90**	€1 .61	€20.31	**-€0.70**
Replication 2	€22.27	€21. 7	**€0.30**	€26.06	€26.13	**-€0.07**	€24.30	€25.03	**-€0.73**	€22.27	€21. 7	**€0.30**
Replication 3	€20.26	€21.60	**-€1.34**	€22.7	€23.30	**-€0.52**	€21. 2	€24.32	**-€2.50**	€20.26	€21.60	**-€1.34**
Replication 4	€1 . 5	€1 . 6	**-€1.01**	€27.10	€25.76	**€1.34**	€24.70	€25.	**-€1.18**	€1 . 5	€1 . 6	**-€1.01**
Replication 5	€1 . 1	€20.52	**-€1.71**	€24.27	€25.26	**-€0.99**	€24.02	€24.53	**-€0.51**	€1 . 1	€20.52	**-€1.71**
Replication 6	€1 .40	€1 .62	**-€0.22**	€25.76	€26.16	**-€0.40**	€24.52	€26.51	**-€1.99**	€1 .40	€1 .62	**-€0.22**
Replication 7	€21.25	€20.74	**€0.51**	€25.4	€24.73	**€0.75**	€24.73	€23.43	**€1.30**	€21.25	€20.74	**€0.51**
Replication 8	€20.31	€20. 7	**-€0.66**	€27.27	€27. 4	**-€0.67**	€25.2	€26.33	**-€1.04**	€20.31	€20. 7	**-€0.66**
Replication 9	€1 . 3	€21.44	**-€1.51**	€24.3	€23.13	**€1.26**	€24.23	€23.66	**€0.57**	€1 . 3	€21.44	**-€1.51**
Replication 10	€21.2	€21.44	**-€0.15**	€26.0	€25.67	**€0.42**	€27.6	€24.40	**€3.29**	€21.2	€21.44	**-€0.15**
Mean	€20.10	€20.75	**-€0.65**	€25.35	€25.27	**€0.08**	€24.55	€25.02	**-€0.47**	€20.10	€20.75	**-€0.65**
Variance	€1.54	€0.	**€0.57**	€1.	€2.03	**€0.67**	€2.06	€1.25	**€3.08**	€1.54	€0.	**€0.57**
n	10	10	**10**	10	10	**10**	10	10	**10**	10	10	**10**
α	0.05	0.05	**0.05**	0.05	0.05	**0.05**	0.05	0.05	**0.05**	0.05	0.05	**0.05**
t-value	2.26	2.26	**2.26**	2.26	2.26	**2.26**	2.26	2.26	**2.26**	2.26	2.26	**2.26**
Lower bound CI	€1 .21	€20.04	**-€1.19**	€24.34	€24.24	**-€0.50**	€23.52	€24.22	**-€1.72**	€1 .21	€20.04	**-€1.19**
Upper bound CI	€20.99	€21.45	**-€0.11**	€26.36	€26.2	**€0.67**	€25.57	€25. 2	**€0.79**	€20.	€21.45	**-€0.11**

B.4. Hypothesis 4

In this section we present the results that we used to put Hypothesis 4 to a test. The results are split into two sections. In the first section we present the results used to compare an increasing arrival rate pattern to the empirical arrival rate pattern that is more representative for practice. In the second section we give the results used to analyze the impact of forecast errors on the performance of BAS.

B.4.1. Comparison of Different Arrival Time Distributions

Table B.16 and Table B.17 present the results of the comparisons we did for Hypothesis 4 with regard to a changing arrival rate pattern. In all cases the strategy we consider is BAS. Whenever we distinguish a better performance for one of the arrival time distributions on a certain KPI, we show the result for that distribution and KPI in bold. For the KPIs where it is not straightforward whether it is better to have a higher or a lower score, we leave it to the reader to judge which result is best.

Table B.16. *Results Hypothesis 4 per scenario, strategy and arrival time distribution*

	Scenario 1a			**Scenario 1b**		
KPI	*BAS (Empirical)*	*BAS (Increasing)*	*OBS*	*BAS (Empirical)*	*BAS (Increasing)*	*OBS*
Customers served (% of total)	**0.9068**	0.8953	0.6072	**0.9253**	0.9125	0.6062
Costs per customer (€)	**19.92**	20.52	21.36	**20.10**	20.13	21.50
Total driving time (seconds)	69903	72985	**58390**	76147	74439	**61263**
Total duration (seconds)	131919	134887	**99736**	137989	135615	**101625**
Total distance (km)	994.44	1059.71	**907.88**	1086.24	1063.66	**945.16**
Total costs (€)	1977.66	2010.01	**1415.66**	2034.67	2008.57	**1425.12**
Number of delivery routes	8	8	8	8	7.9	7.9
Number vehicles used of type 1	1.7	1.8	4	2.9	3	4
Number vehicles used of type 2	2.3	2.2	4	1.1	0.9	3.9
Number vehicles used of type 3	4	4	0	4	4	0
Average utilization (%)	0.9607	0.9679	0.9872	0.9436	0.9471	0.9580
Number of customers served	**99.5**	98.2	66.4	**101.6**	100.1	66.6
Time windows offered	**3.59**	3.53	2.41	**3.64**	3.59	2.40
Response time (seconds)	1.5658	2.3452	**0.1493**	2.3487	3.5686	**0.1003**
Running time (seconds)	600	740	**320**	668	875	**330**
Schedule update time (seconds)	0.1722	0.2720	**0.1653**	0.1628	0.2318	**0.1495**
Quick optimize time (seconds)	61	71	**45**	58	70	**46**
Final optimize time (seconds)	94	88	**55**	95	98	**63**
Change fleet time (seconds)	11.7177	16.3804	**0**	12.4940	20.4181	**0**

Table B.17. *Results Hypothesis 4 per scenario, strategy and arrival time distribution (continued)*

	Scenario 2a			Scenario 2b		
KPI	*BAS (Empirical)*	*BAS (Increasing)*	*OBS*	*BAS (Empirical)*	*BAS (Increasing)*	*OBS*
Customers served (% of total)	**0.8740**	0.8707	0.6927	**0.8678**	0.8592	0.5906
Costs per customer (€)	25.35	25.47	**25.31**	**24.55**	24.91	28.22
Total driving time (seconds)	116955	116348	**106201**	106304	107500	**98475**
Total duration (seconds)	167925	167414	**146179**	157304	158080	**133707**
Total distance (km)	1954.42	1970.30	**1842.07**	1768.16	1797.83	**1741.61**
Total costs (€)	2424.53	2426.59	**1916.52**	2328.85	2339.89	**1817.93**
Number of delivery routes	8	8	8	8	8	7.9
Number vehicles used of type 1	2.1	2.1	4	3.3	3.7	4
Number vehicles used of type 2	1.9	1.9	4	0.7	0.3	3.9
Number vehicles used of type 3	4	4	0	4	4	0
Average utilization (%)	0.8997	0.9167	0.9612	0.8913	0.8994	0.8852
Number of customers served	**95.9**	95.5	75.9	**95.1**	94.0	64.7
Time windows offered	**3.27**	3.26	2.55	3.25	**3.26**	2.19
Response time (seconds)	1.5681	2.1304	**0.1495**	3.3232	4.3260	**0.1862**
Running time (seconds)	641	817	**342**	790	940	**314**
Schedule update time (seconds)	**0.1507**	0.2696	0.1748	0.1566	0.2048	**0.1431**
Quick optimize time (seconds)	66	83	**47**	62	62	**43**
Final optimize time (seconds)	108	118	**63**	91	79	**55**
Change fleet time (seconds)	2.6599	4.6397	**0**	4.7538	9.2231	**0**

Table B.18 presents the data that we used for the paired-t approach to compare the percentage of customers that are served for BAS with an empirical arrival rate pattern and an increasing arrival rate pattern.

Table B.18. *Data used to compare the percentage of customers served for BAS (Empirical) and BAS (Increasing)*

	Scenario 1a			Scenario 1b			Scenario 2a			Scenario 2b		
	BAS (E)	*BAS (I)*	***Diff***	*BAS (E)*	*BAS (I)*	***Diff***	*BAS (E)*	*BAS (I)*	***Diff***	*BAS (E)*	*BAS (I)*	***Diff***
Replication 1	84.48%	83.62%	**0.86%**	88.79%	88.79%	**0.00%**	84.48%	81.90%	**2.59%**	81.90%	79.31%	**2.59%**
Replication 2	91.18%	91.18%	**0.00%**	89.22%	90.20%	**-0.98%**	89.22%	89.22%	**0.00%**	94.12%	90.20%	**3.92%**
Replication 3	83.33%	81.75%	**1.59%**	84.13%	80.16%	**3.97%**	83.33%	83.33%	**0.00%**	82.54%	76.19%	**6.35%**
Replication 4	92.98%	93.86%	**-0.88%**	95.61%	96.49%	**-0.88%**	81.58%	83.33%	**-1.75%**	82.46%	79.82%	**2.63%**
Replication 5	86.67%	85.00%	**1.67%**	92.50%	86.67%	**5.83%**	85.83%	81.67%	**4.17%**	80.83%	81.67%	**-0.83%**
Replication 6	95.54%	91.96%	**3.57%**	99.11%	99.11%	**0.00%**	84.82%	85.71%	**-0.89%**	83.93%	82.14%	**1.79%**
Replication 7	92.31%	91.35%	**0.96%**	94.23%	93.27%	**0.96%**	90.38%	90.38%	**0.00%**	90.38%	91.35%	**-0.96%**
Replication 8	91.84%	93.88%	**-2.04%**	95.92%	93.88%	**2.04%**	89.80%	88.78%	**1.02%**	94.90%	94.90%	**0.00%**
Replication 9	93.52%	91.67%	**1.85%**	89.81%	87.96%	**1.85%**	93.52%	95.37%	**-1.85%**	90.74%	91.67%	**-0.93%**
Replication 10	95.00%	91.00%	**4.00%**	96.00%	96.00%	**0.00%**	91.00%	91.00%	**0.00%**	86.00%	92.00%	**-6.00%**
Mean	90.68%	89.53%	**1.16%**	92.53%	91.25%	**1.28%**	87.40%	87.07%	**0.33%**	86.78%	85.92%	**0.86%**
Variance	0.19%	0.19%	**0.03%**	0.20%	0.32%	**0.05%**	0.15%	0.21%	**0.03%**	0.28%	0.45%	**0.12%**
n	10	10	**10**	10	10	**10**	10	10	**10**	10	10	**10**
α	0.05	0.05	**0.05**	0.05	0.05	**0.05**	0.05	0.05	**0.05**	0.05	0.05	**0.05**
t-value	2.26	2.26	**2.26**	2.26	2.26	**2.26**	2.26	2.26	**2.26**	2.26	2.26	**2.26**
Lower bound CI	87.59%	86.40%	**-0.16%**	89.31%	87.23%	**-0.30%**	84.61%	83.79%	**-1.01%**	82.99%	81.12%	**-1.57%**
Upper bound CI	93.78%	92.66%	**2.48%**	95.75%	95.28%	**2.86%**	90.18%	90.34%	**1.66%**	90.57%	90.73%	**3.28%**

Table B.19 presents the data that we used for the paired-t approach to compare the average delivery costs per customer served for BAS with an empirical arrival rate pattern and an increasing arrival rate pattern.

Table B.19. *Data used for comparing the costs per customer served for BAS (Empirical) and BAS (Increasing)*

	Scenario 1a			**Scenario 1b**			**Scenario 2a**			**Scenario 2b**		
	BAS (E)	*BAS (I)*	***Diff***	*BAS (E)*	*BAS (I)*	***Diff***	*BAS (E)*	*BAS (I)*	***Diff***	*BAS (E)*	*BAS (I)*	***Diff***
Replication 1	€1 .	€20.71	**-€0.83**	€1 .61	€1 .	**-€0.38**	€24.30	€25.32	**-€1.02**	€24.1	€25.44	**-€1.26**
Replication 2	€20.66	€22.10	**-€1.44**	€22.27	€22.55	**-€0.28**	€26.06	€26.06	**€0.00**	€24.30	€24.	**-€0.58**
Replication 3	€1 .67	€1 .23	**-€0.56**	€20.26	€20. 6	**-€0.70**	€22.7	€23.12	**-€0.34**	€21. 2	€23. 1	**-€1.99**
Replication 4	€1 .53	€1 . 4	**€0.69**	€1 . 5	€1 .	**-€0.04**	€27.10	€26.15	**€0.95**	€24.70	€24.47	**€0.23**
Replication 5	€20.2	€1 .	**€0.29**	€1 . 1	€1 .45	**-€0.64**	€24.27	€25.23	**-€0.96**	€24.02	€24.61	**-€0.59**
Replication 6	€1 .20	€20.2	**-€1.08**	€18.40	€17.	**€0.52**	€25.76	€26.05	**-€0.29**	€24.52	€26.14	**-€1.62**
Replication 7	€20.46	€21.22	**-€0.76**	€21.25	€21.07	**€0.18**	€25.4	€25.17	**€0.31**	€24.73	€23.44	**€1.29**
Replication 8	€21.74	€22.6	**-€0.95**	€20.31	€1 .54	**€0.77**	€27.27	€27.56	**-€0.29**	€25.2	€26.3	**-€1.09**
Replication 9	€1 .70	€1 .43	**-€0.73**	€1 . 3	€20.35	**-€0.42**	€24.3	€23. 0	**€0.49**	€24.23	€23. 6	**€0.27**
Replication 10	€20.03	€20.6	**-€0.66**	€21.2	€20.57	**€0.72**	€26.0	€26.14	**-€0.05**	€27.6	€25. 6	**€1.73**
Mean	€1 . 2	€20.52	**-€0.60**	€20.10	€20.13	**-€0.03**	€25.35	€25.47	**-€0.12**	€24.55	€24. 1	**-€0.36**
Variance	€0.	€1.53	**€0.40**	€1.54	€1.6	**€0.30**	€1.	€1.57	**€0.37**	€2.06	€1.07	**€1.50**
n	10	10	**10**	10	10	**10**	10	10	**10**	10	10	**10**
α	0.05	0.05	**0.05**	0.05	0.05	**0.05**	0.05	0.05	**0.05**	0.05	0.05	**0.05**
t-value	2.26	2.26	**2.26**	2.26	2.26	**2.26**	2.26	2.26	**2.26**	2.26	2.26	**2.26**
Lower bound CI	€1 .24	€1 .63	**-€1.06**	€1 .21	€1 .20	**-€0.42**	€24.34	€24.57	**-€0.56**	€23.52	€24.17	**-€1.24**
Upper bound CI	€20.5	€21.40	**-€0.15**	€20.	€21.05	**€0.37**	€26.36	€26.37	**€0.32**	€25.57	€25.65	**€0.52**

B.4.2. Impact of Fixed Forecast Errors

Table B.20 presents the results of the comparisons we did for Hypothesis 4 with regard to the impact of fixed forecast errors. In all cases the strategy we consider is BAS. Whenever we distinguish a better performance for one of the forecast error values on a certain KPI, we show the corresponding result in bold. For the KPIs where it is not straightforward whether it is better to have a higher or a lower score, we leave it to the reader to judge which result is best.

Table B.20. *Results Hypothesis 4 for different fixed forecast error values (BAS, Scenario 1b)*

	Scenario 1b						
KPI	*-30%*	*-20%*	*-10%*	*Expected*	*+10%*	*+20%*	*+30%*
Customers served (% of total)	0.9021	0.9065	**0.9298**	0.9253	0.8907	0.8775	0.8856
Costs per customer (€)	**19.52**	19.54	19.63	20.10	20.64	20.85	20.31
Total driving time (seconds)	71584	**70500**	73469	76147	75411	74944	71893
Total duration (seconds)	129664	**129150**	134897	137989	135747	133960	131209
Total distance (km)	994.03	**987.47**	1036.90	1086.24	1093.73	1103.01	1024.55
Total costs (€)	**1929.70**	1938.63	1998.75	2034.67	2010.83	1998.89	1965.15
Number of delivery routes	7.5	7.7	7.9	8	7.8	7.8	7.9
Number vehicles used of type 1	2.3	2.7	2.8	2.9	2.9	2.9	3.2
Number vehicles used of type 2	1.2	1	1.1	1.1	0.9	0.9	0.8
Number vehicles used of type 3	4	4	4	4	4	4	3.9
Average utilization (%)	0.8173	0.8396	0.9169	0.9436	0.9393	0.9196	0.9369
Number of customers served	99.1	99.6	**102.1**	101.6	97.7	96.2	97.1
Time windows offered	3.59	3.61	**3.70**	3.64	3.49	3.43	3.41
Response time (seconds)	2.1947	**2.0324**	2.5249	2.3487	3.1777	3.8989	4.1781
Running time (seconds)	786	683	757	**668**	674	716	767
Schedule update time (seconds)	0.2066	**0.1298**	0.1827	0.1628	0.1687	0.1902	0.2142
Quick optimize time (seconds)	71	65	67	58	47	**42**	45
Final optimize time (seconds)	119	110	109	95	66	**50**	56
Change fleet time (seconds)	2.3720	**2.1796**	2.4133	12.4940	29.9544	40.6213	42.1502

Table B.21 and Table B.22 present the results of the comparison of the percentage of customers served using a paired-t approach. We compare the performance for each different forecast error value to the base case where we have no standard forecast error. Table B.23 and Table B.24 present the results of the comparison in terms of the average delivery costs per customer served. The differences between two cases are computed with Equation (B.1), in which $j \in$ {-30%, -20%, -10%, 0%, +10%, +20%, +30%}. The 95% confidence intervals are computed with Equation (6.2).

$$Z_j = X_{BAS0\%} - X_{BASj},\ j \neq 0\% \tag{B.1}$$

Table B.21. *Comparison percentage of customers served for different fixed forecast errors (BAS, Scenario 1b) - I*

	Expected -30%			**Expected -20%**			**Expected -10%**		
	0%	*-30%*	***Diff***	*0%*	*-20%*	***Diff***	*0%*	*-10%*	***Diff***
Replication 1	88.79%	87.93%	**0.86%**	88.79%	88.79%	**0.00%**	88.79%	90.52%	**-1.72%**
Replication 2	89.22%	89.22%	**0.00%**	89.22%	90.20%	**-0.98%**	89.22%	93.14%	**-3.92%**
Replication 3	84.13%	84.92%	**-0.79%**	84.13%	84.92%	**-0.79%**	84.13%	85.71%	**-1.59%**
Replication 4	95.61%	92.98%	**2.63%**	95.61%	93.86%	**1.75%**	95.61%	95.61%	**0.00%**
Replication 5	92.50%	88.33%	**4.17%**	92.50%	89.17%	**3.33%**	92.50%	91.67%	**0.83%**
Replication 6	99.11%	94.64%	**4.46%**	99.11%	96.43%	**2.68%**	99.11%	98.21%	**0.89%**
Replication 7	94.23%	91.35%	**2.88%**	94.23%	92.31%	**1.92%**	94.23%	94.23%	**0.00%**
Replication 8	95.92%	93.88%	**2.04%**	95.92%	93.88%	**2.04%**	95.92%	95.92%	**0.00%**
Replication 9	89.81%	88.89%	**0.93%**	89.81%	87.96%	**1.85%**	89.81%	89.81%	**0.00%**
Replication 10	96.00%	90.00%	**6.00%**	96.00%	89.00%	**7.00%**	96.00%	95.00%	**1.00%**
Mean	92.53%	90.21%	**2.32%**	92.53%	90.65%	**1.88%**	92.53%	92.98%	**-0.45%**
Variance	0.20%	0.09%	**0.05%**	0.20%	0.12%	**0.05%**	0.20%	0.13%	**0.02%**
n	10	10	**10**	10	10	**10**	10	10	**10**
α	0.05	0.05	**0.05**	0.05	0.05	**0.05**	0.05	0.05	**0.05**
t-value	2.26	2.26	**2.26**	2.26	2.26	**2.26**	2.26	2.26	**2.26**
Lower bound CI	89.31%	88.06%	**0.79%**	89.31%	88.20%	**0.23%**	89.31%	90.38%	**-1.55%**
Upper bound CI	95.75%	92.37%	**3.85%**	95.75%	93.10%	**3.53%**	95.75%	95.59%	**0.65%**

Table B.22. *Comparison percentage of customers served for different fixed forecast errors (BAS, Scenario 1b) - II*

	Expected +10%			**Expected +20%**			**Expected +30%**		
	0%	*-30%*	***Diff***	*0%*	*-20%*	***Diff***	*0%*	*-10%*	***Diff***
Replication 1	88.79%	81.90%	**6.90%**	88.79%	88.79%	**0.00%**	88.79%	86.21%	**2.59%**
Replication 2	89.22%	91.18%	**-1.96%**	89.22%	89.22%	**0.00%**	89.22%	89.22%	**0.00%**
Replication 3	84.13%	80.95%	**3.17%**	84.13%	73.02%	**11.11%**	84.13%	75.40%	**8.73%**
Replication 4	95.61%	93.86%	**1.75%**	95.61%	92.98%	**2.63%**	95.61%	91.23%	**4.39%**
Replication 5	92.50%	85.00%	**7.50%**	92.50%	85.00%	**7.50%**	92.50%	89.17%	**3.33%**
Replication 6	99.11%	96.43%	**2.68%**	99.11%	93.75%	**5.36%**	99.11%	93.75%	**5.36%**
Replication 7	94.23%	94.23%	**0.00%**	94.23%	89.42%	**4.81%**	94.23%	92.31%	**1.92%**
Replication 8	95.92%	92.86%	**3.06%**	95.92%	91.84%	**4.08%**	95.92%	93.88%	**2.04%**
Replication 9	89.81%	83.33%	**6.48%**	89.81%	81.48%	**8.33%**	89.81%	82.41%	**7.41%**
Replication 10	96.00%	91.00%	**5.00%**	96.00%	92.00%	**4.00%**	96.00%	92.00%	**4.00%**
Mean	92.53%	89.07%	**3.46%**	92.53%	87.75%	**4.78%**	92.53%	88.56%	**3.98%**
Variance	0.20%	0.33%	**0.09%**	0.20%	0.41%	**0.12%**	0.20%	0.34%	**0.07%**
n	10	10	**10**	10	10	**10**	10	10	**10**
α	0.05	0.05	**0.05**	0.05	0.05	**0.05**	0.05	0.05	**0.05**
t-value	2.26	2.26	**2.26**	2.26	2.26	**2.26**	2.26	2.26	**2.26**
Lower bound CI	89.31%	84.99%	**1.26%**	89.31%	83.17%	**2.26%**	89.31%	84.39%	**2.09%**
Upper bound CI	95.75%	93.16%	**5.65%**	95.75%	92.32%	**7.30%**	95.75%	92.72%	**5.86%**

Table B.23. *Comparison costs per customer served for different fixed forecast errors (BAS, Scenario 1b) - I*

	Expected -30%			**Expected -20%**			**Expected -10%**		
	0%	*-30%*	***Diff***	*0%*	*-20%*	***Diff***	*0%*	*-10%*	***Diff***
Replication 1	€1 .61	€17. 6	**€1.65**	€1 .61	€1 . 0	**€0.71**	€1 .61	€1 .67	**€0.94**
Replication 2	€22.27	€20. 7	**€1.40**	€22.27	€22.24	**€0.03**	€22.27	€22.24	**€0.03**
Replication 3	€20.26	€1 .63	**€0.63**	€20.26	€1 .27	**€0.99**	€20.26	€1 .	**€0.37**
Replication 4	€1 . 5	€1 .1	**-€0.34**	€1 . 5	€1 .56	**€0.29**	€1 . 5	€1 .42	**€0.43**
Replication 5	€1 . 1	€1 .77	**€0.04**	€1 . 1	€1 .62	**€0.19**	€1 . 1	€1 .52	**€0.29**
Replication 6	€1 .40	€1 .0	**€0.32**	€1 .40	€17. 0	**€0.50**	€1 .40	€1 .7	**-€0.38**
Replication 7	€21.25	€20.44	**€0.81**	€21.25	€1 . 2	**€1.33**	€21.25	€20.33	**€0.92**
Replication 8	€20.31	€20.42	**-€0.11**	€20.31	€1 .27	**€1.04**	€20.31	€1 .7	**€1.53**
Replication 9	€1 . 3	€1 .54	**€0.39**	€1 . 3	€1 . 1	**€1.02**	€1 . 3	€1 .72	**€0.21**
Replication 10	€21.2	€20.32	**€0.97**	€21.2	€21.7	**-€0.49**	€21.2	€20. 2	**€0.37**
Mean	€20.10	€1 .52	**€0.58**	€20.10	€1 .54	**€0.56**	€20.10	€1 .63	**€0.47**
Variance	€1.54	€1.03	**€0.42**	€1.54	€1.	**€0.32**	€1.54	€1.57	**€0.29**
n	10	10	**10**	10	10	**10**	10	10	**10**
α	0.05	0.05	**0.05**	0.05	0.05	**0.05**	0.05	0.05	**0.05**
t-value	2.26	2.26	**2.26**	2.26	2.26	**2.26**	2.26	2.26	**2.26**
Lower bound CI	€1 .21	€1 .7	**€0.11**	€1 .21	€1 .53	**€0.16**	€1 .21	€1 .73	**€0.09**
Upper bound CI	€20.	€20.25	**€1.04**	€20.	€20.55	**€0.96**	€20.	€20.52	**€0.85**

Table B.24. *Comparison costs per customer served for different fixed forecast errors (BAS, Scenario 1b) – II*

	Expected +10%			**Expected +20%**			**Expected +30%**		
	0%	*-30%*	***Diff***	*0%*	*-20%*	***Diff***	*0%*	*-10%*	***Diff***
Replication 1	€1 .61	€21.54	**-€1.93**	€1 .61	€1 .	**-€0.28**	€1 .61	€1 .74	**-€0.13**
Replication 2	€22.27	€21.77	**€0.50**	€22.27	€22.1	**€0.09**	€22.27	€22.0	**€0.19**
Replication 3	€20.26	€20. 0	**-€0.64**	€20.26	€22.75	**-€2.49**	€20.26	€21.4	**-€1.23**
Replication 4	€1 . 5	€1 . 0	**-€1.05**	€1 . 5	€20.20	**-€1.35**	€1 . 5	€1 .16	**-€0.31**
Replication 5	€1 . 1	€1 . 7	**-€1.16**	€1 . 1	€20.02	**-€1.21**	€1 . 1	€1 .25	**-€0.44**
Replication 6	€1 .40	€17. 0	**€0.60**	€1 .40	€17.77	**€0.63**	€1 .40	€17.71	**€0.69**
Replication 7	€21.25	€20.7	**€0.46**	€21.25	€22.0	**-€0.84**	€21.25	€20.32	**€0.93**
Replication 8	€20.31	€20.57	**-€0.26**	€20.31	€20.0	**€0.22**	€20.31	€1 .56	**€0.75**
Replication 9	€1 . 3	€21.02	**-€1.09**	€1 . 3	€22.06	**-€2.13**	€19.93	€22.2	**-€2.36**
Replication 10	€21.2	€22.15	**-€0.86**	€21.2	€21.47	**-€0.18**	€21.2	€21.53	**-€0.24**
Mean	€20.10	€20.64	**-€0.54**	€20.10	€20. 5	**-€0.75**	€20.10	€20.31	**-€0.22**
Variance	€1.54	€1.52	**€0.72**	€1.54	€2.32	**€1.07**	€1.54	€2.23	**€0.99**
n	10	10	**10**	10	10	**10**	10	10	**10**
α	0.05	0.05	**0.05**	0.05	0.05	**0.05**	0.05	0.05	**0.05**
t-value	2.26	2.26	**2.26**	2.26	2.26	**2.26**	2.26	2.26	**2.26**
Lower bound CI	€1 .21	€1 .76	**-€1.15**	€1 .21	€1 .76	**-€1.49**	€1 .21	€1 .25	**-€0.93**
Upper bound CI	€20.	€21.52	**€0.06**	€20.	€21. 4	**-€0.01**	€20.	€21.3	**€0.50**

B.4.3. Impact of Fluctuating Forecast Errors

Table B.25 presents the results of the comparisons we did for Hypothesis 4 with regard to the impact of fluctuating forecast errors. In all cases the strategy we consider is BAS. Whenever we distinguish a better performance for one of the forecast error values on a certain KPI, we show the corresponding result in bold. For the KPIs where it is not straightforward whether it is better to have a higher or a lower score, we leave it to the reader to judge which result is best. Note that the experiment in which we forecast a uniform distribution of customer arrival times whereas the real distribution is an empirical one, is abbreviated with *BAS(R=E,F=U)*. The experiment where the distributions are exchanged is abbreviated with *BAS(R=U,F=E)*. The base case, where we do not have a forecast error, is abbreviated with *BAS(NoError)*.

Table B.25. *Results Hypothesis 4 for different fluctuating forecast error experiments (BAS, Scenario 1b)*

	Scenario 1b		
KPI	*BAS(R=E,F=U)*	*BAS(R=U,F=E)*	*BAS(NoError)*
Customers served (% of total)	0.8944	0.9225	**0.9253**
Costs per customer (€)	20.66	**20.03**	20.10
Total driving time (seconds)	76780	**74942**	76147
Total duration (seconds)	137128	**136802**	137989
Total distance (km)	1119.58	**1065.19**	1086.24
Total costs (€)	**2020.72**	2022.43	2034.67
Number of delivery routes	7.8	8	8
Number vehicles used of type 1	2.7	2.9	2.9
Number vehicles used of type 2	1.1	1.1	1.1
Number vehicles used of type 3	4	4	4
Average utilization (%)	0.9248	0.9537	0.9436
Number of customers served	98.1	101.3	**101.6**
Time windows offered	3.48	3.63	**3.64**
Response time (seconds)	3.4355	2.4648	**2.3487**
Running time (seconds)	**646**	714	668
Schedule update time (seconds)	**0.1397**	0.1687	0.1628
Quick optimize time (seconds)	**40**	62	58
Final optimize time (seconds)	**45**	103	95
Change fleet time (seconds)	23.3162	**11.4593**	12.4940

Table B.26 presents the results of the comparison of the percentage of customers served using a paired-t approach. We compare the performance for each experiment with fluctuating forecast error values to the base case where we have no forecast error. Table B.27 presents the results of the comparison in terms of the average delivery costs per customer served. The differences in performance between two experiments are computed with Equation (B.2), in which $j \in \{(R{=}E,F{=}U), (R{=}U,F{=}E), (NoError)\}$. The 95% confidence intervals are computed with Equation (6.2).

$$Z_j = X_{BAS(NoError)} - X_{BASj},\ j \neq (NoError) \tag{B.2}$$

Table B.26. *Comparison percentage customers served fluctuating forecast error experiments (BAS, Scenario 1b)*

	Reality Empirical, Forecast Uniform			**Reality Uniform, Forecast Empirical**		
	BAS(NoError)	*BAS(R=E,F=U)*	***Diff***	*BAS(NoError)*	*BAS(R=U,F=E)*	***Diff***
Replication 1	88.79%	89.66%	**-0.86%**	88.79%	88.79%	**0.00%**
Replication 2	89.22%	89.22%	**0.00%**	89.22%	89.22%	**0.00%**
Replication 3	84.13%	78.57%	**5.56%**	84.13%	84.13%	**0.00%**
Replication 4	95.61%	92.11%	**3.51%**	95.61%	95.61%	**0.00%**
Replication 5	92.50%	85.83%	**6.67%**	92.50%	92.50%	**0.00%**
Replication 6	99.11%	94.64%	**4.46%**	99.11%	98.21%	**0.89%**
Replication 7	94.23%	92.31%	**1.92%**	94.23%	94.23%	**0.00%**
Replication 8	95.92%	91.84%	**4.08%**	95.92%	94.90%	**1.02%**
Replication 9	89.81%	84.26%	**5.56%**	89.81%	88.89%	**0.93%**
Replication 10	96.00%	96.00%	**0.00%**	96.00%	96.00%	**0.00%**
Mean	92.53%	89.44%	**3.09%**	92.53%	92.25%	**0.28%**
Variance	0.20%	0.28%	**0.07%**	0.20%	0.19%	**0.00%**
n	10	10	**10**	10	10	**10**
α	0.05	0.05	**0.05**	0.05	0.05	**0.05**
t-value	2.26	2.26	**2.26**	2.26	2.26	**2.26**
Lower bound CI	89.31%	85.68%	**1.18%**	89.31%	89.14%	**-0.04%**
Upper bound CI	95.75%	93.21%	**5.00%**	95.75%	95.36%	**0.61%**

Table B.27. *Comparison costs per customer served fluctuating forecast error experiments (BAS, Scenario 1b)*

	Reality Empirical, Forecast Uniform			**Reality Uniform, Forecast Empirical**		
	BAS(NoError)	*BAS(R=E,F=U)*	***Diff***	*BAS(NoError)*	*BAS(R=U,F=E)*	***Diff***
Replication 1	€1 .61	€20.02	**-€0.41**	€1 .61	€1 .61	**€0.00**
Replication 2	€22.27	€22.30	**-€0.03**	€22.27	€22.27	**€0.00**
Replication 3	€20.26	€21. 1	**-€1.65**	€20.26	€20.26	**€0.00**
Replication 4	€1 . 5	€1 .30	**-€0.45**	€1 . 5	€1 . 5	**€0.00**
Replication 5	€1 . 1	€1 .64	**-€0.83**	€1 . 1	€1 . 1	**€0.00**
Replication 6	€1 .40	€17. 2	**€0.48**	€1 .40	€1 .31	**€0.09**
Replication 7	€21.25	€21.	**-€0.63**	€21.25	€20.	**€0.37**
Replication 8	€20.31	€20.57	**-€0.26**	€20.31	€20.5	**-€0.27**
Replication 9	€1 . 3	€21.70	**-€1.77**	€1 . 3	€20.05	**-€0.12**
Replication 10	€21.2	€21.37	**-€0.08**	€21.2	€20.72	**€0.57**
Mean	€20.10	€20.66	**-€0.56**	€20.10	€20.03	**€0.06**
Variance	€1.54	€2.02	**€0.49**	€1.54	€1.40	**€0.06**
n	10	10	**10**	10	10	**10**
α	0.05	0.05	**0.05**	0.05	0.05	**0.05**
t-value	2.26	2.26	**2.26**	2.26	2.26	**2.26**
Lower bound CI	€1 .21	€1 .65	**-€1.07**	€1 .21	€1 .1	**-€0.11**
Upper bound CI	€20.	€21.6	**-€0.06**	€20.	€20.	**€0.24**

B.5. Prophet Benchmark

In this section we present the results used for the benchmark of our strategies against prophet strategies. In Section B.5.1 we do so for Use Case 2, and in Section B.5.2 for Use Case 4.

B.5.1. Use Case 2

Table B.28, Table B.29, Table B.30 and Table B.31 present the results of the comparisons we did for the prophet benchmark for Use Case 2. A better performance for one of the strategies on a certain KPI, is shown in bold.

Table B.28. *Results prophet benchmark per scenario and strategy for Use Case 2*

	Scenario 1a				
KPI	*Prophet (NTW)*	*Prophet (FCTW)*	*BAS*	*MYS*	*OBS*
Customers served (% of total)	**0.7843**	0.7797	0.6876	0.5964	0.6072
Costs per customer (€)	**16.12**	17.38	19.34	21.67	21.36
Total driving time (seconds)	**52751**	63408	61218	57655	58390
Total duration (seconds)	100283	110958	105774	**98515**	99736
Total distance (km)	**732.18**	934.66	921.27	901.92	907.88
Total costs (€)	**1383.56**	1483.37	1451.89	1407.69	1415.66
Number of delivery routes	8	8	8	8	8
Number vehicles used of type 1	3.1	3.4	3.8	3.6	4
Number vehicles used of type 2	5.1	4.6	4.2	4.4	4
Average utilization (%)	0.9830	0.9813	0.9778	0.9868	0.9872
Number of customers served	**86.0**	85.5	75.2	65.2	66.4

Table B.29. *Results prophet benchmark per scenario and strategy for Use Case 2 (continued)*

	Scenario 1b				
KPI	*Prophet (NTW)*	*Prophet (FCTW)*	*BAS*	*MYS*	*OBS*
Customers served (% of total)	**0.8093**	0.8065	0.6964	0.6333	0.6062
Costs per customer (€)	**15.92**	17.04	19.37	21.06	21.50
Total driving time (seconds)	**55397**	65357	64253	62870	61263
Total duration (seconds)	104069	113969	108449	104726	**101625**
Total distance (km)	**753.67**	946.82	952.31	957.70	945.16
Total costs (€)	**1408.90**	1502.53	1472.96	1453.35	1425.12
Number of delivery routes	8	8	8	8	7.9
Number vehicles used of type 1	5.4	5.5	4.8	5	4
Number vehicles used of type 2	2.6	2.5	3.2	3	3.9
Average utilization (%)	0.9822	0.9797	0.9672	0.9828	0.9580
Number of customers served	**88.8**	88.5	76.3	69.4	66.6

Table B.30. *Results prophet benchmark per scenario and strategy for Use Case 2 (continued)*

	Scenario 3a				
KPI	*Prophet (NTW)*	*Prophet (FCTW)*	*BAS*	*MYS*	*OBS*
Customers served (% of total)	**0.8368**	0.8169	0.7241	0.6772	0.6927
Costs per customer (€)	**20.52**	23.12	24.51	25.88	25.31
Total driving time (seconds)	**100615**	119880	108665	106184	106201
Total duration (seconds)	146149	164424	149369	**145490**	146179
Total distance (km)	**1657.18**	2088.07	1876.21	1850.01	1842.07
Total costs (€)	**1879.38**	2067.08	1941.07	1914.29	1916.52
Number of delivery routes	8	8	8	8	8
Number vehicles used of type 1	4.1	3.9	4	3.6	4
Number vehicles used of type 2	3.9	4.1	4	4.4	4
Average utilization (%)	0.9849	0.9754	0.9271	0.9584	0.9612
Number of customers served	**91.7**	89.5	79.4	74.2	75.9

Table B.31. *Results prophet benchmark per scenario and strategy for Use Case 2 (continued)*

	Scenario 3b				
KPI	*Prophet (NTW)*	*Prophet (FCTW)*	*BAS*	*MYS*	*OBS*
Customers served (% of total)	**0.8280**	0.8106	0.7022	0.6479	0.5896
Costs per customer (€)	**20.38**	22.81	24.92	26.71	28.23
Total driving time (seconds)	**96771**	114462	105229	102458	98284
Total duration (seconds)	142017	159228	145255	140684	**133462**
Total distance (km)	**1610.11**	2014.11	1837.58	1817.14	1738.38
Total costs (€)	1847.00	2023.42	1910.49	1881.01	**1815.93**
Number of delivery routes	8	8	8	8	7.9
Number vehicles used of type 1	5.9	6	5.3	5.6	4
Number vehicles used of type 2	2.1	2	2.7	2.4	3.9
Average utilization (%)	0.9837	0.9784	0.9283	0.9588	0.8836
Number of customers served	**90.7**	88.8	76.8	70.7	64.6

B.5.2. Use Case 4

Table B.32, Table B.33, Table B.34 and Table B.35 present the results of the comparisons for the prophet benchmark for Use Case 4. We show a better performance for a strategy on a certain KPI in bold.

Table B.32. *Results prophet benchmark per scenario and strategy for Use Case 4*

	Scenario 1a				
KPI	*Prophet (NTW)*	*Prophet (FCTW)*	*BAS*	*MYS*	*OBS*
Customers served (% of total)	**0.9491**	**0.9491**	0.9068	0.8568	0.6072
Costs per customer (€)	**17.60**	19.25	19.92	21.19	21.36
Total driving time (seconds)	**54067**	71571	69903	71031	58390
Total duration (seconds)	117145	134511	131919	131127	**99736**
Total distance (km)	**695.47**	1011.70	994.44	1039.20	907.88
Total costs (€)	1828.99	2001.43	1977.66	1984.76	**1415.66**
Number of delivery routes	8	8	8	8	8
Number vehicles used of type 1	1.7	1.5	1.7	2.1	4
Number vehicles used of type 2	2.3	2.5	2.3	1.9	4
Number vehicles used of type 3	4	4	4	4	0
Average utilization (%)	0.9752	0.9747	0.9607	0.9723	0.9872
Number of customers served	**104.2**	**104.2**	99.5	93.8	66.4

Table B.33. *Results prophet benchmark per scenario and strategy for Use Case 4 (continued)*

	Scenario 1b				
KPI	*Prophet (NTW)*	*Prophet (FCTW)*	*BAS*	*MYS*	*OBS*
Customers served (% of total)	**0.9651**	0.9643	0.9253	0.8868	0.6062
Costs per customer (€)	**17.50**	19.20	20.10	21.12	21.50
Total driving time (seconds)	**56635**	74683	76147	76855	61263
Total duration (seconds)	119995	137959	137989	137347	**101625**
Total distance (km)	**716.38**	1046.14	1086.24	1134.02	945.16
Total costs (€)	1849.75	2027.86	2034.67	2040.94	**1425.12**
Number of delivery routes	8	8	8	8	7.9
Number vehicles used of type 1	3.2	3.1	2.9	2.7	4
Number vehicles used of type 2	0.8	0.9	1.1	1.3	3.9
Number vehicles used of type 3	4	4	4	4	0
Average utilization (%)	0.9732	0.9730	0.9436	0.9666	0.9580
Number of customers served	**106.0**	105.9	101.6	97.0	66.6

Table B.34. *Results prophet benchmark per scenario and strategy for Use Case 4 (continued)*

	Scenario 3a				
KPI	*Prophet (NTW)*	*Prophet (FCTW)*	*BAS*	*MYS*	*OBS*
Customers served (% of total)	**0.9622**	0.9395	0.8740	0.8560	0.6927
Costs per customer (€)	**20.90**	24.52	25.35	26.28	25.31
Total driving time (seconds)	**92873**	125793	116955	119377	106201
Total duration (seconds)	148661	179841	167925	170347	**146179**
Total distance (km)	**1418.38**	2135.56	1954.42	2027.42	1842.07
Total costs (€)	2203.01	2523.16	2424.53	2460.47	**1916.52**
Number of delivery routes	8	8	8	8	8
Number vehicles used of type 1	2.4	1.6	2.1	1.6	4
Number vehicles used of type 2	1.6	2.5	1.9	2.4	4
Number vehicles used of type 3	4	3.9	4	4	0
Average utilization (%)	0.9697	0.9471	0.8997	0.9313	0.9612
Number of customers served	**105.6**	103.1	95.9	93.7	75.9

Table B.35. *Results prophet benchmark per scenario and strategy for Use Case 4 (continued)*

	Scenario 3b				
KPI	*Prophet (NTW)*	*Prophet (FCTW)*	*BAS*	*MYS*	*OBS*
Customers served (% of total)	**0.9564**	0.9330	0.8678	0.8612	0.5906
Costs per customer (€)	**20.88**	24.24	24.55	25.52	28.22
Total driving time (seconds)	**90150**	119202	106304	113068	98475
Total duration (seconds)	145902	173526	157304	164356	**133707**
Total distance (km)	**1404.56**	2026.84	1768.16	1916.89	1741.61
Total costs (€)	2186.83	2476.32	2328.85	2403.37	**1817.93**
Number of delivery routes	8	8	8	8	7.9
Number vehicles used of type 1	3	3.4	3.3	3.2	4
Number vehicles used of type 2	1	0.6	0.7	0.8	3.9
Number vehicles used of type 3	4	4	4	4	0
Average utilization (%)	0.9734	0.9517	0.8913	0.9290	0.8852
Number of customers served	**104.9**	102.3	95.1	94.2	64.7

B.6. Impact of Intermediate Optimization Calls

Table B.36 presents the results of the comparisons we did between simulations with and without intermediate optimization calls for Use Case 2. Whenever we distinguish a better performance for one of the strategies on a certain KPI, we show the result for that strategy and KPI in bold. For the KPIs where it is not straightforward whether it is better to have a higher or a lower score, we leave it to the reader to judge which result is best.

Table B.36. *Results impact of intermediate optimization calls per scenario and strategy*

	Scenario 1a		**Scenario 1b**		**Scenario 2a**		**Scenario 2b**	
KPI	*OBS (NoOpt)*	*OBS*	*OBS (NoOpt)*	*OBS*	*OBS (NoOpt)*	*OBS*	*OBS (NoOpt)*	*OBS*
Customers served (% of total)	0.5905	**0.6072**	0.6045	**0.6062**	0.6247	**0.6927**	0.5512	**0.5896**
Costs per customer (€)	22.26	**21.36**	21.92	**21.50**	26.92	**25.31**	29.25	**28.23**
Total driving time (seconds)	59994	**58390**	63359	**61263**	**98970**	106201	**93400**	98284
Total duration (seconds)	100662	**99736**	103637	**101625**	**135558**	146179	**126292**	133462
Total distance (km)	955.89	**907.88**	990.37	**945.16**	**1733.65**	1842.07	**1676.70**	1738.38
Total costs (€)	1430.41	**1415.66**	1445.34	**1425.12**	**1835.83**	1916.52	**1754.56**	1815.93
Number of delivery routes	8	8	7.9	7.9	8	8	7.8	7.9
Number vehicles used of type 1	4	4	4	4	4	4	4	4
Number vehicles used of type 2	4	4	3.9	3.9	4	4	3.8	3.9
Average utilization (%)	0.9874	0.9872	0.9566	0.9580	0.9020	0.9612	0.8281	0.8836
Number of customers served	64.5	**66.4**	66.5	**66.6**	68.6	**75.9**	60.4	**64.6**
Time windows offered	2.34	**2.41**	2.38	**2.40**	2.25	**2.55**	1.99	**2.19**
Response time (seconds)	0.2063	**0.1364**	0.1805	**0.1397**	0.2247	**0.1947**	0.2077	**0.1783**
Running time (seconds)	**120**	320	**111**	324	**115**	358	**103**	309
Schedule update time (seconds)	0.1546	**0.1390**	**0.1169**	0.1274	**0.1123**	0.1733	**0.1054**	0.1373
Quick optimize time (seconds)	**0**	45	**0**	46	**0**	49	**0**	42
Final optimize time (seconds)	79	**56**	75	**59**	75	**67**	66	**56**
Change fleet time (seconds)	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

Table B.37 presents the data that we used for the paired-t approach to compare the percentage of customers that are served both in cases with and without intermediate optimization calls.

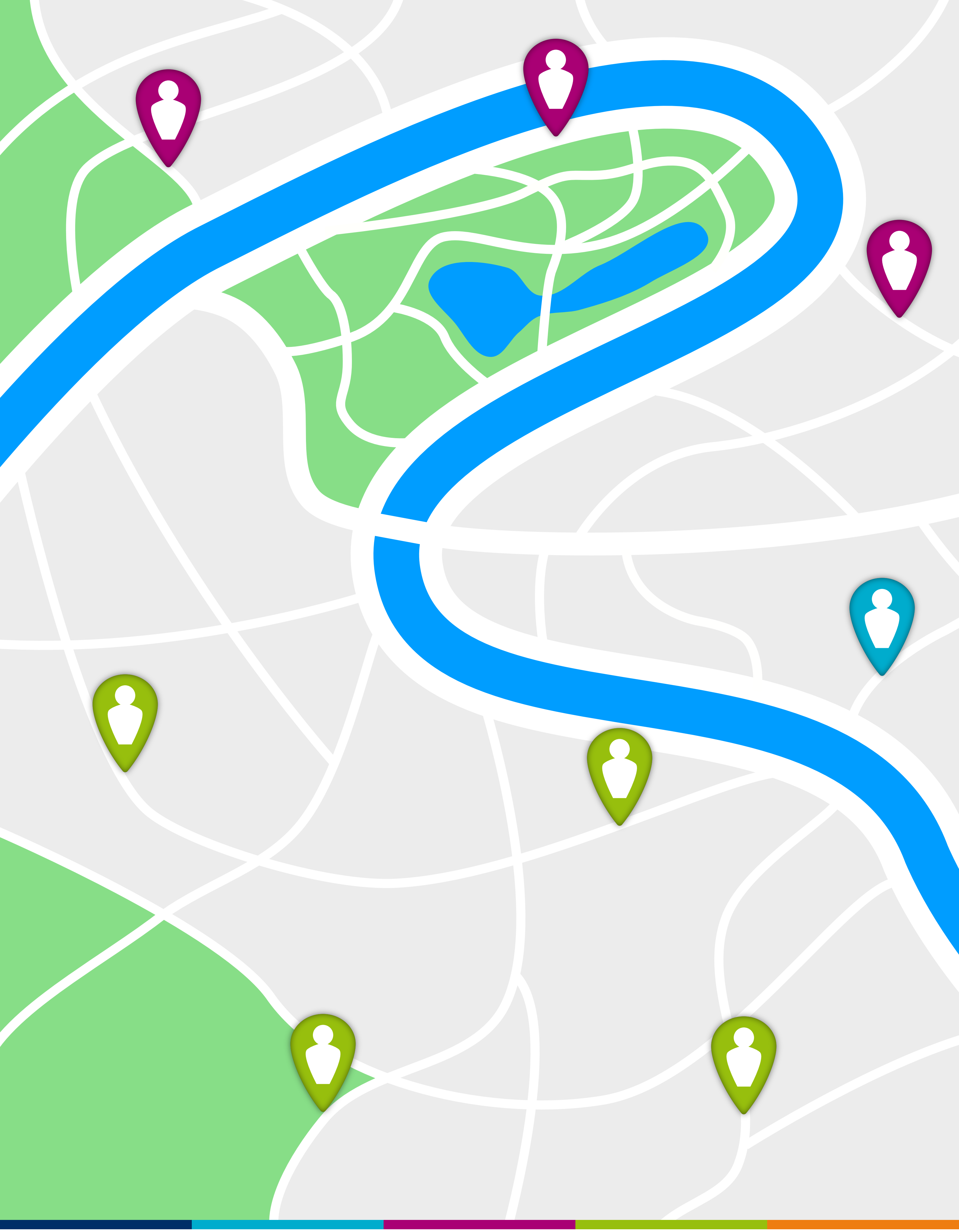
Table B.37. *Data used for comparing the percentage of customers served for OBS (NoOpt) and OBS*

	Scenario 1a			**Scenario 1b**			**Scenario 2a**			**Scenario 2b**		
	OBS (NoOpt)	*OBS*	***Diff***	*OBS (NoOpt)*	*OBS*	***Diff***	*OBS (NoOpt)*	*OBS*	***Diff***	*OBS (NoOpt)*	*OBS*	***Diff***
Replication 1	54.31%	54.31%	**0.00%**	61.21%	60.34%	**0.86%**	58.62%	63.79%	**-5.17%**	47.41%	54.31%	**-6.90%**
Replication 2	57.84%	59.80%	**-1.96%**	61.76%	62.75%	**-0.98%**	60.78%	76.47%	**-15.69%**	57.84%	66.67%	**-8.82%**
Replication 3	44.44%	50.79%	**-6.35%**	52.38%	52.38%	**0.00%**	61.90%	65.08%	**-3.17%**	53.17%	56.35%	**-3.17%**
Replication 4	63.16%	64.91%	**-1.75%**	64.04%	63.16%	**0.88%**	58.77%	64.04%	**-5.26%**	45.61%	49.12%	**-3.51%**
Replication 5	53.33%	53.33%	**0.00%**	62.50%	61.67%	**0.83%**	61.67%	65.00%	**-3.33%**	53.33%	57.50%	**-4.17%**
Replication 6	62.50%	64.29%	**-1.79%**	74.11%	70.54%	**3.57%**	59.82%	69.64%	**-9.82%**	59.82%	58.04%	**1.79%**
Replication 7	63.46%	64.42%	**-0.96%**	56.73%	57.69%	**-0.96%**	70.19%	73.08%	**-2.88%**	61.54%	58.65%	**2.88%**
Replication 8	69.39%	70.41%	**-1.02%**	57.14%	57.14%	**0.00%**	60.20%	69.39%	**-9.18%**	51.02%	56.12%	**-5.10%**
Replication 9	61.11%	62.96%	**-1.85%**	54.63%	55.56%	**-0.93%**	65.74%	72.22%	**-6.48%**	57.41%	63.89%	**-6.48%**
Replication 10	61.00%	62.00%	**-1.00%**	60.00%	65.00%	**-5.00%**	67.00%	74.00%	**-7.00%**	64.00%	69.00%	**-5.00%**
Mean	59.05%	60.72%	**-1.67%**	60.45%	60.62%	**-0.17%**	62.47%	69.27%	**-6.80%**	55.12%	58.96%	**-3.85%**
Variance	0.48%	0.38%	**0.03%**	0.37%	0.27%	**0.05%**	0.15%	0.21%	**0.15%**	0.36%	0.36%	**0.14%**
n	10	10	**10**	10	10	**10**	10	10	**10**	10	10	**10**
α	0.05	0.05	**0.05**	0.05	0.05	**0.05**	0.05	0.05	**0.05**	0.05	0.05	**0.05**
t-value	2.26	2.26	**2.26**	2.26	2.26	**2.26**	2.26	2.26	**2.26**	2.26	2.26	**2.26**
Lower bound CI	54.09%	56.33%	**-2.95%**	56.12%	56.90%	**-1.73%**	59.70%	65.98%	**-9.62%**	50.80%	54.69%	**-6.48%**
Upper bound CI	64.02%	65.12%	**-0.38%**	64.78%	64.35%	**1.38%**	65.24%	72.56%	**-3.98%**	59.44%	63.24%	**-1.22%**

Table B.38 presents the data that we used for the paired-t approach to compare the average delivery costs per customer served both in cases with and without intermediate optimization calls.

Table B.38. *Data used for comparing the costs per customer served for OBS (NoOpt) and OBS*

	Scenario 1a			**Scenario 1b**			**Scenario 2a**			**Scenario 2b**		
	OBS (NoOpt)	*OBS*	***Diff***	*OBS (NoOpt)*	*OBS*	***Diff***	*OBS (NoOpt)*	*OBS*	***Diff***	*OBS (NoOpt)*	*OBS*	***Diff***
Replication 1	€23.24	€22.21	**€1.03**	€20.50	€20.73	**-€0.23**	€27.07	€25.41	**€1.66**	€34.01	€29.73	**€4.28**
Replication 2	€24.31	€22.87	**€1.44**	€23.83	€23.08	**€0.75**	€28.33	€24.98	**€3.35**	€27.58	€27.04	**€0.54**
Replication 3	€23.93	€20.90	**€3.03**	€22.60	€22.41	**€0.19**	€23.56	€23.22	**€0.34**	€26.75	€26.17	**€0.58**
Replication 4	€20.12	€20.11	**€0.01**	€20.34	€20.08	**€0.26**	€28.27	€26.76	**€1.51**	€33.74	€31.84	**€1.90**
Replication 5	€22.93	€21.99	**€0.94**	€20.84	€20.72	**€0.12**	€24.71	€24.50	**€0.21**	€29.30	€27.82	**€1.48**
Replication 6	€21.14	€21.19	**-€0.05**	€18.55	€19.10	**-€0.55**	€27.27	€25.27	**€2.00**	€28.94	€29.00	**-€0.06**
Replication 7	€21.39	€20.84	**€0.55**	€22.98	€22.65	**€0.33**	€25.93	€24.98	**€0.95**	€25.47	€28.18	**-€2.71**
Replication 8	€22.49	€21.68	**€0.81**	€21.53	€21.56	**-€0.03**	€31.25	€28.01	**€3.24**	€30.97	€29.44	**€1.53**
Replication 9	€20.77	€19.80	**€0.97**	€23.86	€22.84	**€1.02**	€25.55	€23.81	**€1.74**	€28.22	€27.42	**€0.80**
Replication 10	€22.27	€22.03	**€0.24**	€24.13	€21.84	**€2.29**	€27.22	€26.17	**€1.05**	€27.48	€25.68	**€1.80**
Mean	€22.26	€21.36	**€0.90**	€21.92	€21.50	**€0.42**	€26.92	€25.31	**€1.61**	€29.25	€28.23	**€1.01**
Variance	€1.92	€0.94	**€0.79**	€3.45	€1.73	**€0.63**	€4.65	€1.96	**€1.13**	€8.16	€3.37	**€3.11**
n	10	10	**10**	10	10	**10**	10	10	**10**	10	10	**10**
α	0.05	0.05	**0.05**	0.05	0.05	**0.05**	0.05	0.05	**0.05**	0.05	0.05	**0.05**
t-value	2.26	2.26	**2.26**	2.26	2.26	**2.26**	2.26	2.26	**2.26**	2.26	2.26	**2.26**
Lower bound CI	€21.27	€20.67	**€0.26**	€20.59	€20.56	**-€0.15**	€25.37	€24.31	**€0.84**	€27.20	€26.92	**-€0.25**
Upper bound CI	€23.25	€22.06	**€1.53**	€23.25	€22.44	**€0.98**	€28.46	€26.31	**€2.37**	€31.29	€29.55	**€2.28**



UNIVERSITY OF TWENTE.

ORTEC