MASTER THESIS

A Methodology to Determine the Most Cost-Efficient Capacity of CoolblueBikes

Machine learning models to support capacity decisions at the tactical level in delivery routing problems.

Susanne Heesterman
March 30, 2023

UNIVERSITY OF TWENTE.

UNIVERSITY OF TWENTE.

This report is written as part of the master graduation assignment, performed at Coolblue B.V., of the Industrial Engineering & Management educational program at the University of Twente.

A Methodology to Determine the Most Cost-Efficient Capacity of CoolblueBikes

Machine learning models to support capacity decisions at the tactical level in delivery routing problems.

Author:

S. L. Heesterman

Research institute: University of Twente
Department: Industrial Engineering & Business Information Systems
Programme: Industrial Engineering & Management
Specialization: Production & Logistics Management

University of Twente

Dr. Ir. J.M.J. Schutten
Faculty of Behavioural, Management and Social Sciences

Dr. Ir. M.R.K. Mes
Faculty of Behavioural, Management and Social Sciences

Coolblue B.V.

T. Oord, MSc
Owner Capacity CoolblueBikes

Publication date: March 30, 2023
Number of pages excluding appendices: 80
Number of pages including appendices: 114
Number of appendices: 11

Preface

This thesis marks the end of my Master Industrial Engineering & Management at the University of Twente, and thus my time as a student. I am very grateful for all the opportunities I have encountered to develop myself both professionally and personally, all the wonderful experiences I have gained, and all the amazing people I have met during my student years.

I would like to thank Coolblue for giving me the opportunity to perform my thesis under their guidance. I appreciate all the support and trust I gained since the very first day. Especially, I want to thank Thijs Oord for his enthusiasm, engagement, and guidance during my research. He consistently made time to think along with any challenge I was facing. Also, I would like to thank Anouk Verhagen for all the data-related support and brainstorming sessions, and the CoolblueBikes team for all the fun times we had and for making me feel part of their team. Furthermore, I would like to express my gratitude to Marco Schutten and Martijn Mes. Their critical feedback allowed me to continuously improve the quality of my research. Our meetings and substantive discussions always resulted in new insights and inspirations, and their interest in the subject and confidence in the research process helped me to stay on the right track. Finally, I would like to thank my friends and family for their continuous support and genuine interest. I appreciate your involvement in any form.

I hope you enjoy reading my thesis.

Susanne Heesterman
Rotterdam, March 2023

Management Summary

We conduct this research at CoolblueBikes (CBB), the bike delivery network of Coolblue, as part of the master graduation assignment of the Industrial Engineering and Management educational program at the University of Twente.

Problem description

CBB is an overflow carrier. Therefore, they can select which potential orders they want to deliver, and decide which remaining orders they want to outsource to delivery partners (DP). Accordingly, they select which potential orders they are going to deliver until all deployed capacity is utilized on the operational level. Beforehand, on the tactical level, they mimic this operational order selection. Accordingly, they determine how much capacity they need to deliver the orders that they are going to select. The capacity is based on the order selection and translated to the number of delivery men hours and bike routes. The current order selection equals all forecasted potential orders from a subarea of the delivery area. However, this subarea is static, i.e., independent of the forecast and the bike route efficiency.

Currently, the costs per order of CBB delivery are higher than the costs per order of DP delivery. The CBB delivery costs consist of fixed costs, and variable costs per order, per hour, and per kilometre. Straightforwardly, CBB desires to have minimal costs per order. Hence, to determine the most cost-efficient capacity to deploy per hub at the tactical level to decrease the costs per order of CBB delivery, we answer the main research question:

How should the optimal required capacity of CBB's hubs be determined such that the costs per order of CBB delivery are minimized?

To answer this question, we develop a solution methodology that supports capacity deployment decisions at the tactical level in routing problems. Our main goal is to find out what number of selected orders defines the capacity optimum given the forecasted potential orders per hub, and how robust this optimum is against order uncertainties. Subsequently, we focus on finding methods that estimate the delivery costs of a subset of orders, because (exactly) evaluating all order combinations is too computationally demanding. Also, we focus on finding methods that evaluate the impact of uncertainty on the most cost-efficient order selection.

Solution methodology

After a current system analysis and a literature review, we propose 1) to develop machine learning model(s) that estimate(s) the costs of an order selection, and 2) to analyse the operational routing plan, using Monte Carlo simulation, in which machine learning model(s) are utilized to support decisions about the most cost-efficient order selection.

1) Development of machine learning models

We develop the Random Forest Regression (RFR) and Lightweight Gradient Boosting Machines (LGBM) models because they are the most suitable machine learning models for our problem. We encounter uncertainty in all types of order information from the to-be-evaluated days. However, we only need to know the number of orders to deliver and not which. Therefore, we decide to develop a tactical level machine learning model, for which we do not have to assume all order information to make predictions. To obtain (spatial) features that describe the solution value, we decide to partition the delivery area into PC4s (the 4 numbers of the Dutch postal code) and describe the potential orders on PC4 level. We train, validate and test both models on two types of data: generated and historical data. Hence, we simulate the order selection of the hub in Hilversum to generate instances. Here, we use the historical daily order divisions per PC4, select every day 4 order subsets of different sizes based on the nearest neighbour heuristic, solve the vehicle routing problem (VRP) of these orders selections, and store the respective features on which we will train the machine learning

models. Because there is insufficient historical data per hub to develop both machine learning models, we group the historical data based on hub size (small or large).

2) Monte Carlo simulation

We simulate the operational routing plan to 1) assess the uncertainty in how the potential orders are distributed over the PC4s, and 2) validate the solution methodology. The simulation consists of 4 phases, which we repeat until both the most cost-efficient number of orders selected and the costs per order of the order selection are statistically significant. Hence, we can evaluate the robustness of the most cost-efficient order selection.

1. In Phase 1, we distribute the forecasted potential orders over the PC4s of the delivery area based on a PC4 order probability distribution.
2. In Phase 2, we select the most cost-efficient order subset with the cheapest insertion heuristic based on estimated insertion costs. We estimate these costs with a machine learning model trained on generated data (1), and a machine learning model trained on grouped historical data (2). Additionally, we estimate the insertion costs with the literature estimation of Daganzo to benchmark from. To validate our solution methodology, we also select orders the same as in the current system, i.e., with the Method 'Optimal levels'.
3. In Phase 3, we solve the VRP of each order selection. Hence, we can compare the cost-efficiency of the retrieved order selections objectively and validate our solution methodology. To solve the VRP, we estimate the order volumes and locations.
4. In Phase 4 we store the metrics belonging to the order selections to evaluate the robustness and to learn what method can best be utilized to support CBB with capacity deployment decisions.

Performance

First, we train, validate and test the machine learning models RFR and LGBM on the generated and historical data with 6 tactical-level features. We find that the RFR models have the best (out-of-sample) predictive performance and that the model trained on generated data (MAPE: 8.28%, Adjusted R^2: 0.984, rRMSE: 8.71%, rMAE: 6.17%) outperforms the model trained on grouped historical data (MAPE: 9.77%, Adjusted R^2: 0.873, rRMSE: 10.45%, rMAE: 8.64%) on all performance metrics. Hence, we decide to utilize these two RFR models in the simulation.

Second, we find that the PC4 order distribution can best be estimated with a negative binomial distribution.

Third, we test our proposed solution methodology on the hub in Hilversum. We study and compare the performance of the 4 order selection methods in 3 experiments: 1) the minimum observed potential orders (40), 2) the average observed potential orders (140) and 3) the maximal observed potential orders (460). We observe that the three order selection methods select significantly more orders than the Method 'Optimal levels' with 40 and 140 potential orders and that the difference in the number of orders selected of all methods is insignificant with 460 orders. Strikingly, the differences in costs per order are never significant. Furthermore, we observe that under the current circumstances, all four methods always select a number of orders close to the maximal number of orders.

We conduct a sensitivity analysis to validate that our solution methodology selects the most cost-efficient number of orders. We learn that if we put more weight on the chosen orders (e.g. higher variable costs, increase in distance or lower fixed costs), the most cost-efficient number of orders to select is less obvious. Both the difference with current circumstances and the mutual difference between Methods Daganzo, RFR (1) and RFR (2) is not significant. However, the Method 'Optimal levels' is insensitive to cost changes i.e., the order selection is independent of the costs, and distance increases. Hence, it is not capable of choosing the most cost-efficient number of orders. Furthermore, we confirmed the value of our solution methodology during an experiment with zero fixed costs: it selects the subset of orders based on the current costs of delivering orders based on route efficiency, and decreases the costs per order between 3.9 % – 18.1 %.

We perform an in-depth analysis to understand the comparable performance of the Methods Daganzo, RFR (1) and RFR (2) in the solution methodology. Additionally, we develop an operational level RFR to show the difference in performance with all order information (operational level) and with uncertainty in all types of order information (tactical level), and thus to validate the development and use of the tactical level RFR. We learn that the operational level RFR has a very high predictive performance both in the simulation and on the test set. Also, we learn that RFR (1) makes similar estimations as the operational level RFR, and thus that both models make a similar decision regarding the most cost-efficient order selection. Hence, we conclude that the tactical level model is capable of selecting the most cost-efficient number of orders to deliver. Furthermore, we learn that the fixed costs per order converge to zero when the number of orders selected converges to infinity and that the difference in variable costs per order shows an oscillating behaviour converging to zero when the orders selected approach the maximal number of potential orders. If we want to make an order selection decision based on the travelled distance, the ratio of fixed costs, variable costs, travel distance and travel time should be such that the reduction in fixed costs per order is overshadowed by the in- or decrease in variable costs per order. Under the current circumstances in Hilversum, although the methods underestimate the distances, i.e., they evaluate with a haversine distance, the decrease in fixed costs per order overshadows the change in variable costs per order. Hence, the order selection decision of the three methods is straightforward, and thus comparable: select a number of orders skewed to the maximal number of potential orders.

Conclusions and evaluations

In conclusion, we provide CBB with a valid solution methodology to determine the required capacity aligned with the bike route efficiency. Although the difference in costs retrieved from these methodologies is not statistically significant with 95% confidence compared to the current method, the method is valid, dynamic, robust, easily implementable, and generalizable to other hubs, unlike the current method. Additionally, the method indicates the efficiency and improvement potential of a hub, and can also be used for order selection and route planning at the operational level with 4 small adjustments. Furthermore, we contribute to the literature with a methodology that selects the most cost-efficient order subset to determine the delivery capacity in routing problems at the tactical level.

We recommend Coolblue the following based on the obtained results:

1. We recommend implementing the solution methodology for all hubs based on the distance estimation of RFR (1).
2. We recommend running the tool once for every number of forecasted potential orders per hub under the current circumstances, and only rerunning when the input parameters or current circumstances change.
3. We recommend storing the input parameters and files in Coolblue’s data warehouse BiqQuery and connecting these to the tool such that they are automatically updated.
4. We recommend implementing the distance estimation and route planning API of CBB in the tool to obtain representative distances and bike routes of the CBB planning.
5. We recommend assigning ownership to the model to ensure its continuity.
6. We recommend storing the historical potential order information on order level instead of PC4 level.
7. We recommend documenting the current working methods of CBB and storing this information in one central place.

Table of Contents

Figures and tables

Figures

Tables

Abbreviations

Table 1 Abbreviations used in this report

Abbreviation	Description
CBB	CoolblueBikes
CVRP	Capacitated vehicle routing problem
DP	Delivery partners
EBITDA	Earnings before interests, taxes, depreciation, and amortization
EFB	Exclusive feature bundling
FSCVRP	Fleet size and composition vehicle routing problem
GBDT	Gradient boosting decision tree
GOSS	Gradient-based one-side sampling
ID	Identifier
IRP	Inventory routing problem
Km	Kilometres
KPI	Key performance indicator
LGBM	Lightweight gradient boosting machines
MAE	Mean absolute error
MAPE	Mean absolute percentage error
Min	Minutes
NPS	Net promoter score
PC4	The four numeric values of the Dutch postal code
PC6	The four numeric and two letters of the Dutch postal code
PRP	Production routing problem
RCCP	Rough-cut capacity planning
RFR	Random forest regression
rMAE	Relative mean absolute error
RMSE	Root mean squared error
rRMSE	Relative root mean squared error
SQL	Structured query language
SVRP	Stochastic vehicle routing problem
TSP	Travelling salesman problem
VRP	Vehicle routing problem

1 Introduction

This chapter introduces the context of the research and the background of this research. Section 1.1 introduces the organization we conduct the research for. Section 1.2 explains the research and Section 1.3 describes the research design.

1.1 About Coolblue

Coolblue is a fast-growing e-commerce company that was founded in Rotterdam in 1999 by Pieter Zwart, Paul de Jong, and Bart Kuijpers. They aim to be an exemplary company in the field of customer-centric entrepreneurship and to be profitable. Coolblue measures these goals with the net promoter score (NPS), i.e., a customer loyalty metric (Fisher & Kordupleski, 2018), and the earnings before interests, taxes, depreciation, and amortization (EBITDA) respectively. To maintain a company with a leading customer journey, they have a constant goal of becoming “a little bit better every day” and “to do anything for a smile”. Coolblue opened 20 physical stores and expanded their service by launching their delivery services Coolbluedelivers in 2016 and Coolbluebikes in 2018 (Coolblue, 2022).

Currently, Coolblue operates both physically and online in the Netherlands, Belgium, and Germany. Here, it offers a wide range of consumer electronics. Additionally, it sells solar panels, charging stations, and home office stores, and it offers renewable energy solutions.

1.1.1 CoolblueBikes

Coolbluebikes (CBB) is the bike delivery network of Coolblue. It services Coolblue’s package deliveries in metropolitan areas and has mainly the goal to contribute to the leading customer journey. Currently, CBB has 22 hubs with a predefined delivery area. They deliver packages to roughly 1.3 million customers every year. The packages that can be delivered by CBB have to meet size, weight, and location restrictions (Coolblue, 2022). CBB refers to the packages that meet the restrictions as potential orders.

CBB is an overflow carrier. This means that they do not have to deliver all potential orders. Therefore, CBB can select the potential orders they want to deliver and which they want to outsource. Delivery partners (DP) like PostNL deliver the remaining (potential) orders. Figure 1 illustrates this logic. Coolblue uses the key performance indicator (KPI) **costs per order** to distinguish how beneficial delivery with CBB and delivery with DP is. The lowest costs per order are the most attractive. Furthermore, CBB measures customer satisfaction with the NPS.

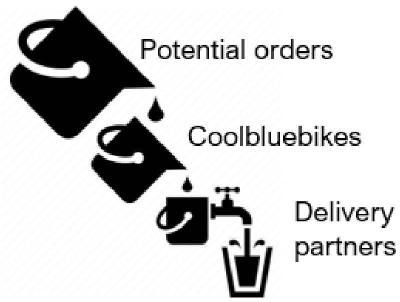

Figure 1 Overflow carrier (Oord, 2022).

1.1.2 Order selection

CBB selects the potential orders that they want to deliver by filtering these orders in two sequential processes, which we explain in Section 1.1.2.1 and Section 1.1.2.2. On the operational level, they select which potential orders they are going to deliver. Beforehand, they determine how much capacity they need to deliver the orders that they are going to select. Thus, they mimic the order selection to plan capacity.

1.1.2.1 Capacity planning

CBB selects the potential orders they want to deliver per hub. Each hub has a predefined delivery area. The delivery area is geographically divided into four fixed subareas, known as delivery levels, shown in Figure 2. Every CBB hub has four delivery levels. CBB delivers a selection of all the potential orders, and thus not necessarily all the potential orders in the delivery area. Therefore, CBB determines per hub which (aggregation of) level(s), and thus in which parts of the delivery area, they always plan to deliver in (Figure 3, Step 1). For example, in Rotterdam, they always plan to deliver in Levels 1, 2, and 3. CBB refers to this aggregation of levels as the ‘optimal’ delivery levels. Note that these delivery levels are not necessarily optimal in mathematical terms.

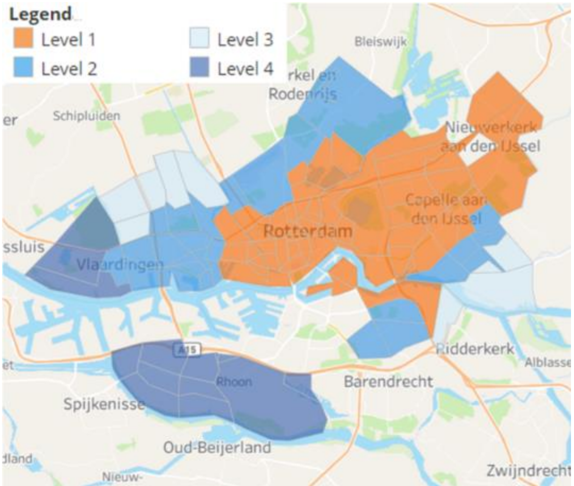

Figure 2 Four levels in the delivery area of the hub in Rotterdam

CBB forecasts the demand only for the ‘optimal’ delivery levels (Figure 3, Step 2). Hence, they determine the required capacity (Figure 3, Step 3) based on the demand in the ‘optimal’ delivery levels. Next, they adjust the hub capacity to this required capacity (Figure 3, Step 4) by hiring a number of delivery men such that in theory, they can deliver all the demand in the ‘optimal’ delivery levels.

Two days before the delivery date, they try to align the capacity with the demand forecast (Figure 3, Step 5). They do this to maximize their capacity utilization. When the forecast is higher than expected, they can decide to decrease the ‘optimal’ delivery levels and vice versa. In the case of Rotterdam, they can anticipate on a higher forecast by decreasing the ‘optimal’ delivery levels to Levels 1 and 2 and on a lower forecast by increasing to Levels 1, 2, 3, and 4. This results in the final aggregation of level(s) CBB is going to deliver in, known as the active delivery area. Figure 3 summarizes this capacity planning process. We refer to Section 2.1 for a more exhaustive explanation of the (steps of the) capacity planning.

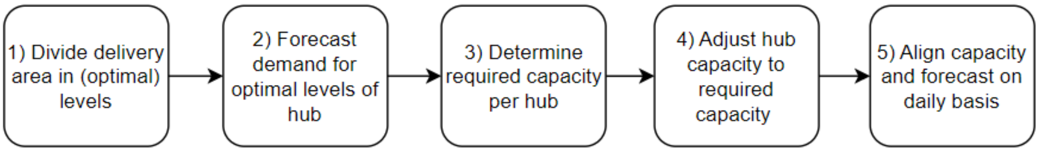

Figure 3 General capacity planning process

1.1.2.2 Route planning

The day before the delivery day, CBB fills the bike routes chronologically, i.e., on the ordering time of the customer, with orders in the active delivery area until the routes are full based on capacity and maximal time per route. The route planning is subject to the capacity determined in the capacity planning process. DP deliver the remaining orders and all the orders outside the final delivery area.

1.2 Research motivation

Although CBB focuses on high customer satisfaction (NPS), they should also be cost-efficient (EBITDA). Therefore, the orders selected to be delivered should obtain high customer satisfaction and have low costs per order. Figure 4 shows that currently, the NPS of CBB is higher than DP as is desirable. Undesirably, Figure 4 also shows that the relative[1] average costs per order of CBB are higher than the relative costs per order of DP. This stresses that it is currently financially more attractive to outsource delivery to DP. Therefore, the current method to select orders results in too high costs per order.

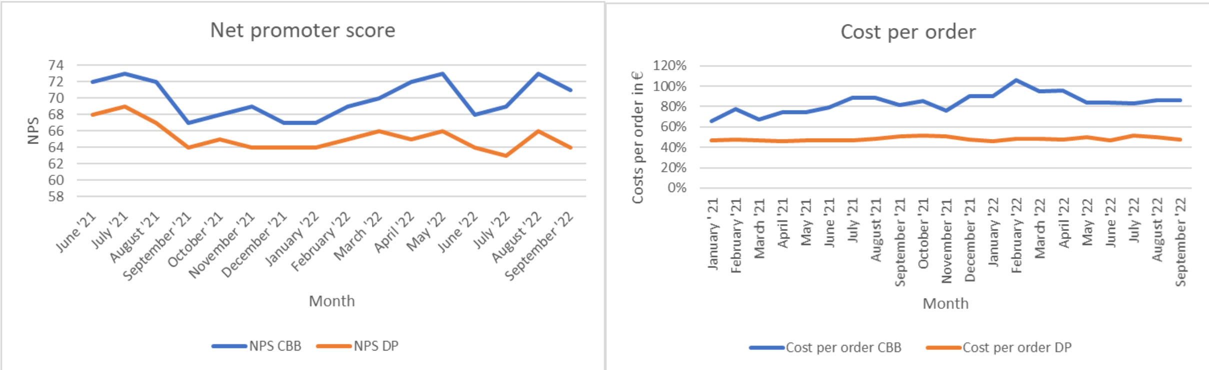

Figure 4 Net promoter score (left) and cost per order (right) of CBB and DP

1.2.1 Problem identification

To investigate how we can decrease the costs per order, we create a problem cluster to identify the cause-and-effect relationships that lead to the core problem(s) (Heerkens & van Winden, 2017). Figure 5 shows the problem cluster.

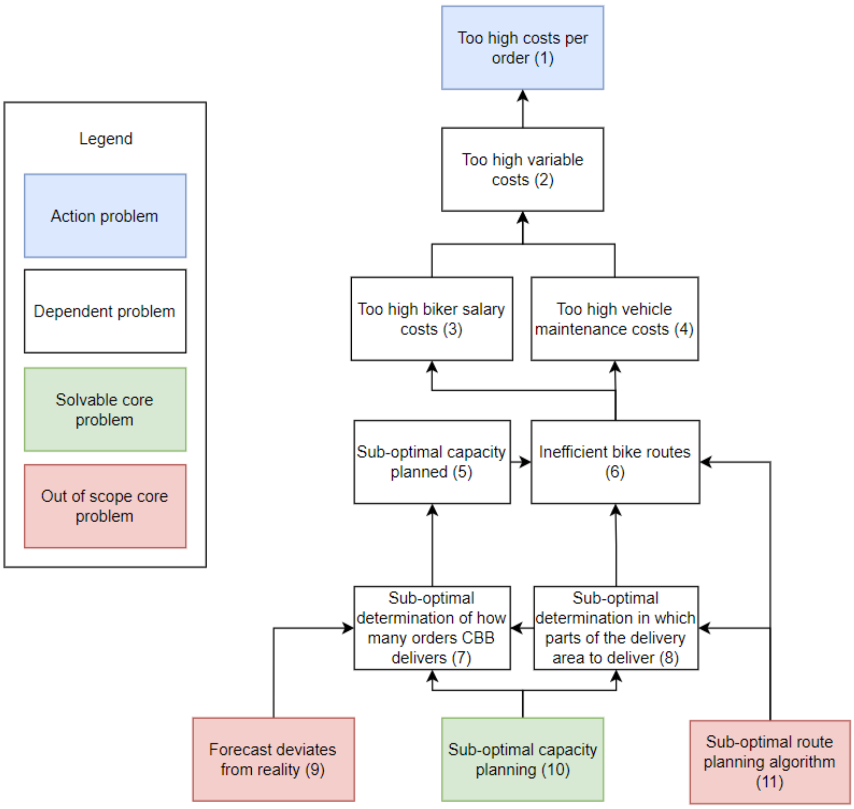

Figure 5 Problem cluster

[1] We display relative (%) costs in this report due to confidentiality. We refer to Appendix K for the costs in € to which all costs in this report are compared.

High costs per order (1)
The costs per order currently consist of more than **70%** of variable costs. The variable costs (2) are expenses that change in proportion to how many orders are delivered. They increase if CBB delivers more and decrease when CBB delivers less (Kenton, 2022). The high variable costs are caused by a high biker salary (3) and high vehicle maintenance costs (4). The biker salary is the salary of the delivery men of CBB. These costs are too high because the bike routes are inefficient (6) in terms of orders delivered per hour. The biker salary costs per hour and the maintenance frequency of the vehicles stay the same when CBB delivers fewer orders per hour. Therefore, the same costs are divided over fewer orders, which increases the variable costs per order.

The following three core problems are the causes of the inefficient bike routes (6).

Sub-optimal capacity planning (10)
The current capacity planning (10) uses delivery levels, i.e., subareas of the delivery area, to determine how many orders CBB plans to deliver (7) and in which parts of the delivery area to deliver (8). The number of orders equals the entire forecast of the ‘optimal’ delivery levels, which CBB translates to capacity (5). Therefore, CBB determines its capacity based on a multitude of full delivery levels and assumes that it is always better to deliver in an entire delivery level. Also, the delivery levels are static, i.e., CBB always plans the capacity with the same ‘optimal’ delivery levels. However, capacity determination based on the optimal orders and area to deliver in depends on many variables, such as the number of orders, location of the orders, etc. Therefore, the (order selection in the) capacity planning is not aligned with bike route efficiency (6).

Sub-optimal route planning algorithm (11)
CBB uses a sub-optimal route planning algorithm (11). CBB fills the bike routes chronologically with orders from the active delivery area until the routes are full based on time or capacity. Therefore, it happens that orders that make less efficient tours are delivered by CBB (6), simply because they were ordered at an earlier time by the customer. To minimize this inefficiency, i.e., such that they exclude orders that are further away, CBB works with delivery levels. CBB can only deliver orders within the levels of the active delivery area (8). Contradictory, this excludes orders located in the inactive delivery area, which possibly could make more efficient routes because orders in the inactive delivery area might give smaller detours than orders in the active delivery area.

To improve this, CBB started with a project where they can cherry-pick the orders. The orders are chosen from the entire delivery area, i.e., independent of the levels. This project shows already improvement potential in route efficiency and is currently in the implementation phase. Therefore, we leave it out of scope for this research.

Forecast deviates from reality (9)
The required capacity (5) is determined based on the number of orders CBB plans to deliver (7). These are equal to the entire forecast of the optimal delivery levels. Therefore, if the forecast deviates from reality, the required capacity planned is not representative of the actual order demand.

The forecasting team tries to minimize the forecasting error. Together with Coolblue, we decide to leave the order forecast out of scope because we expect to gain more profit from an improved capacity planning.

1.2.2 Research problem
The problem identification of Section 1.2.1 brings us to the research problem (10) according to the rules of Heerkens & van Winden (2017):

The current capacity planning of CBB results in too high costs per order.

A tool or model that dynamically determines the most cost-efficient capacity helps to overcome this problem. This research investigates the current capacity planning and advices on how to adjust the capacity such that the costs per order are minimal.

1.3 Research design

This section formulates the research goal and the corresponding (sub)-research questions.

1.3.1 Research goal

We formulate the research objective as follows:

Develop a model or tool that determines the most cost-efficient capacity to deploy per hub based on the forecasted potential orders.

We visualize this research objective in Figure 6. It shows that in the desired situation we dynamically determine the most cost-efficient capacity to deploy aligned with route efficiency and based on the forecasted potential orders instead of the use of static, predetermined 'optimal' delivery levels. Note that CBB refers to these levels as optimal. The required capacity is based on the order selection and translated to the number of delivery men hours and bike routes. This research objective results in the following main research question:

How should the optimal required capacity of CBB's hubs be determined such that the costs per order of CBB delivery are minimized?

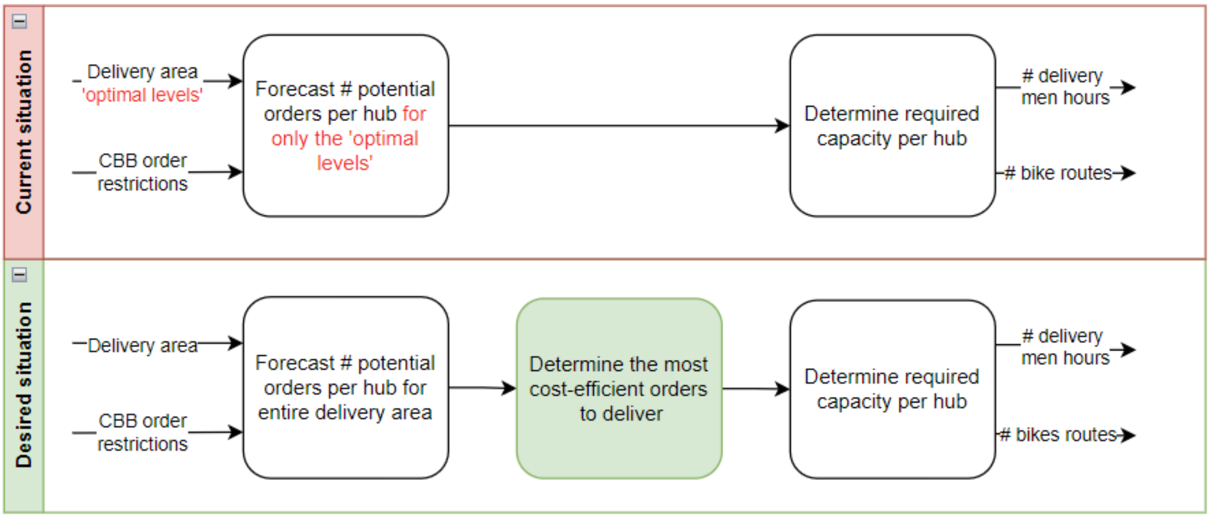

Figure 6 Current and desired situation

This research aims to achieve the research objective by answering the research question. We answer the research question by providing insight into the following aspects:

- Analyse the current determination of the chosen capacity (number of delivery men hours and bike routes) and understand which variables impact the costs per order.
- Provide insight into the similarities and gaps between the current situation at CBB and the available literature.
- Show how these gaps can be covered for CBB with the proposed model or tool.
- Describe how the model or tool determines the most cost-efficient capacity per hub, how robust this capacity decision is, and how this can be used in the capacity planning process.

Note that this research focuses only on capacity planning and not on the operational route planning or forecasting, although they do interact. The capacity planning is subject to the forecast. The route planning is subject to the capacity planned.

1.3.2 Research questions

To answer the main research question, we define research questions and corresponding sub-research questions. We assign each research question to a chapter, such that they structure this research.

Chapter 2 – Current situation

1. *What is the current situation of CBB?*
   a) *How is the current capacity planning process organized?*
   b) *Which costs and variables impact the costs per order?*
   c) *What are the constraints and requirements for the model or tool?*

We describe the current situation in Chapter 2. We identify the main activities to plan capacity and how CBB determines how many orders they plan to deliver. Also, we analyse on which variables the costs per order are dependent. This information gives an understanding of the current system of CBB, such that we can identify improvement opportunities. Lastly, we research what constraints and requirements the model or tool that determines the deployable capacity should suffice. We obtain answers to these questions via interviews with employees and data analysis.

Chapter 3 – Literature review

2. *Which methods that decide upon the most cost-efficient capacity in routing problems are proposed in the literature?*
   a) *What type of routing problem do we encounter?*
   b) *How can a subset of orders be selected to determine capacity in routing problems?*
   c) *What solution methods can be applied to estimate the costs of our routing problem?*
   d) *How can we validate the results?*

We review the literature in Chapter 3. To determine the most cost-efficient capacity to deploy, we should learn how we can estimate the costs of a routing problem given a selection of orders and how we can create cost-efficient order selections. Hence, we translate the studied problem to theoretical problems and identify the similarities and gaps. Furthermore, we review the problem-solving approaches and discuss which methods are applicable to our studied problem. Finally, we determine how we can validate the results of the model or tool that advises on the capacity to deploy.

Chapter 4 – Solution design

3. *How can we develop a capacity model or tool for CBB that dynamically determines the most cost-efficient delivery capacity to deploy?*
   a) *What steps of the methodology need to be taken?*
   b) *What is the order selection logic of the capacity model or tool?*
   c) *How do we determine the costs of an order selection?*

In Chapter 4, we design a methodology that supports capacity deployment decisions at the tactical level in routing problems. Hence, the capacity model determines the order selection with minimal costs per order. We elaborate on the steps of the methodology that need to be taken, the order selection logic of the capacity tool and the determination of the costs of the orders selected.

Chapter 5 – Performance

4. *What is the performance of the proposed model or tool and the respective improvement?*
   a) *How does the capacity decision from the solution methodology perform?*

b) *How does the performance of the solution method differ from the current situation?*
c) *How sensitive is the capacity decision?*

We test the performance of the proposed model in Chapter 5. We validate our results with a case study, by comparing our proposed models based on historical numbers of forecasted potential orders with the current situation. We conduct a sensitivity analysis to analyze the effect of changes in input variables on the capacity decision.

Chapter 6 – Conclusions and recommendations
Lastly, we present our conclusions, theoretical and practical contributions, recommendations and limitations and suggestions for further research in Chapter 6.

1.3.3 Scope

This research focuses on the tactical capacity planning of CBB in the Netherlands. Hence, we focus on how to determine the most cost-efficient delivery capacity for CBB. Note that this excludes the creation of the forecast. Therefore, we consider the potential order forecast method and the respective forecasting inputs as fixed. The same holds for the consecutive steps once the required capacity is determined, like the final capacity scheduling. Also note that this research does not focus on the operational route planning, although this does interact with capacity planning. The planned capacity is input for the route planning, as it plans the routes subject to the capacity planned.

2 Current situation

This chapter describes the current situation at CBB. Section 2.1 elaborates on the current capacity planning process and identifies the improvement opportunities. Next, Section 2.2 introduces the costs per order and explains what variables the costs per order depends. Section 2.3 lists all constraints and requirements for CBB delivery relevant to a capacity model. Lastly, Section 2.4 concludes the chapter.

2.1 Capacity planning process

We elaborate on the capacity planning process of CBB to explain how the capacity is currently determined. We divide the current capacity planning process into three planning levels: strategical, tactical and operational. Figure 7 shows the main activities of the capacity planning process per level.

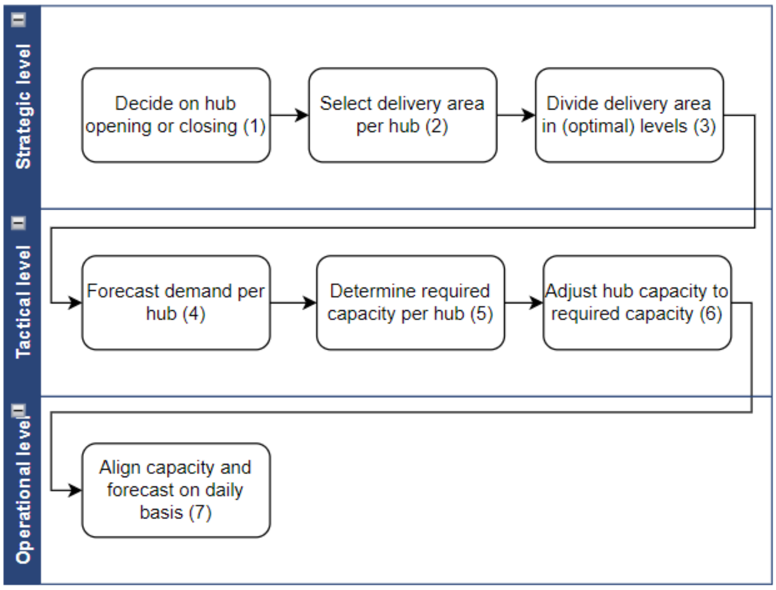

Figure 7 Capacity planning process

2.1.1 Strategical level

The strategical level addresses structural, long-term decision-making on a typical planning horizon of 1 to 5 years. It considers capacities that take a long time to change, either to acquire new capacity or to reduce capacity levels (Olhager, Rudberg, & Wikner, 1999). Hence, CBB addresses these decisions by determination of hub opening or closing, the selection of the delivery area per hub and their respective levels.

Hub opening/closing (1)

CBB bases the decision of hub opening or closing on a business case. They use historical data to determine the expected number of orders and the associated costs. Hence, they decide to open a hub based on the following criteria:

1. The expected costs per order of the CBB hub are at most the costs per order of DP. Note that the expected costs take into account the possible increase in customer satisfaction.
2. The hub location is attractive enough to find delivery men.

The business case includes the expected growth in demand for the coming years. Therefore, all the current 22 hubs have sufficient capacity to handle all orders, even if the demand increases. The current expectation is that the demand for the hubs in the Netherlands will not grow significantly in the coming years.

Selection delivery area (2)

When CBB decides to open a new hub, they select the delivery area in postal codes based on the distance to the hub, order density ($orders/km^2$), safety and accessibility. The hub is located roughly in the centre of the delivery area. Figure 8 shows the delivery areas of all 22 hubs. Each hub is visualized with a different colour.

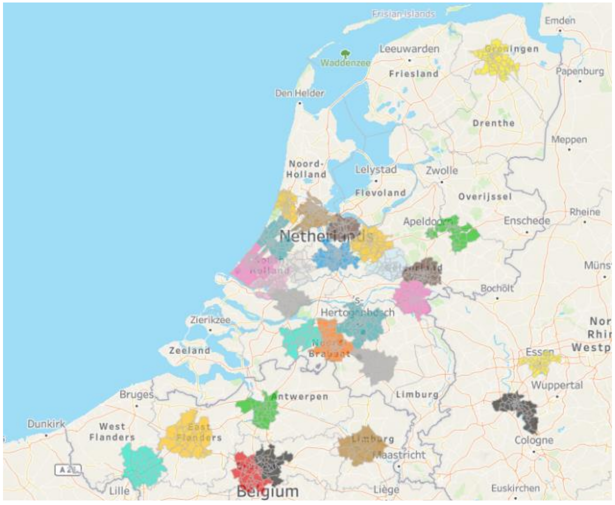

Figure 8 Delivery area of all hubs

Levels (3)

Next, CBB divides the found delivery area into 4 levels. They order the postal codes manually based on the distance to the hub, order density and on practical experience. This is mostly done by logical thinking and not with mathematical logic. Level 1 includes roughly the best 50% of the postal codes in terms of the shortest distance to the hub, highest order density and high practicability, Level 2 the following 20%, Level 3 the next 20% and Level 4 the final 10%.

Afterwards, CBB determines what aggregation of levels are the 'optimal levels' to deliver in. This is done with an analysis, where they estimate the costs per order with the historical route efficiency (number of orders delivered per hour) per level and expected volume per level based on the current costs. The aggregated levels with the lowest costs per order are the optimal levels to deliver in with CBB. Based on experience, CBB can decide to deviate from these optimal delivery levels. The sum of the forecasted orders from the postal codes belonging to the 'optimal levels' is equal to the total forecast of the hub. Based on this forecast, CBB plans the required capacity.

In Rotterdam, the 'optimal levels' are Levels 1, 2 and 3. Therefore, Level 4 is not included in the forecast as explained in Section 1.1.2.1. We analyze the data to find if it is indeed straightforward based on the order density to exclude Level 4. Figure 9 shows the order density

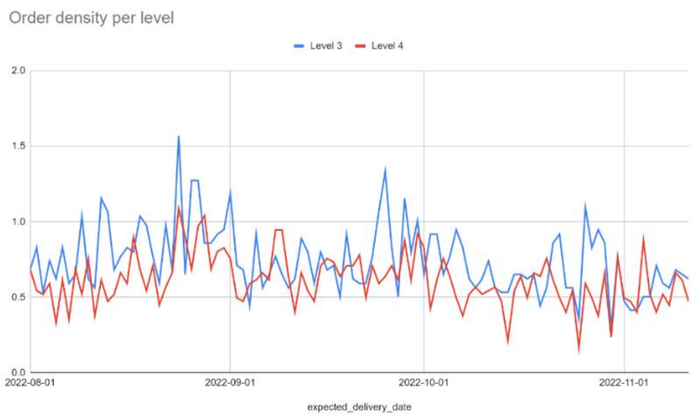

Figure 9 Order density level 3 & 4

of Levels 3 and 4 of the hub in Rotterdam. On several days, for example in September and November, the order density is higher in Level 4 than in Level 3. The orders in Level 4 are excluded from CBB delivery. However, it could be preferable to determine if it is more cost-efficient to visit areas with a high order density outside the 'optimal levels' instead of cycling to orders within the 'optimal levels' in a low order density area. Therefore, it might be interesting to determine where and how much to deliver based on the dynamic forecast instead of the static 'optimal levels'.

2.1.2 Tactical level

The tactical level connects the strategical and operational levels. It addresses decisions regarding optimal resource allocation for a few weeks or months ahead. This middle-term planning horizon provides flexibility to anticipate uncertainties and changes (Hans, Herroelen, Leus, & Wullink, 2005). Accordingly, CBB determines a forecast, decides on their required capacity and adjusts their hub capacity on this.

Forecast demand (4)

Given the 'optimal levels' per hub, CBB determines the number of expected potential orders located in these levels. Based on historical data, they generate a list of orders sufficing the size, weight, volume, delivery moment and level restrictions. Next, the forecasting department modifies the number of potential CBB orders from the last 14 days based on trends, season, advertising and manual adjustments. This gives a forecast of potential orders located in the 'optimal levels' of each hub. For example, in Rotterdam, these are all forecasted potential orders located in Levels 1, 2 & 3.

Note that this forecast is only a number of all the expected orders in the 'optimal' delivery area, i.e., it does not provide any specific information on order location. We analyze the historical data to find out if a more detailed forecast, e.g. on PC4 level (the area belonging to the 4 numbers of the Dutch postal code), might be interesting. We list the PC4s and order this list by the daily number of potential orders observed in the PC4. We find that for example in Hilversum in February 2023, 31 of the 45 PC4s were in the top 10 of PC4s with the most potential orders, from which 8 PC4s were only observed once or twice in this top 10. Even in the top 5, we observe 22 PC4s. This indicates that it is not straightforward to predict in which PC4s the most potential orders will be. Furthermore, we analyze if the PC4s always have the same percentage of orders relative to the total number of potential orders. Figure 10 shows that these percentages fluctuate a lot, and thus that it might not be representative to distribute the forecast of potential orders over the PC4s with a fixed proportion. However, it does show that the percentages of all PC4s are roughly in the same range (0.01% – 0.11%), meaning that although there is no clear daily pattern per PC4, the orders are approximately proportionally spread over the PC4s.

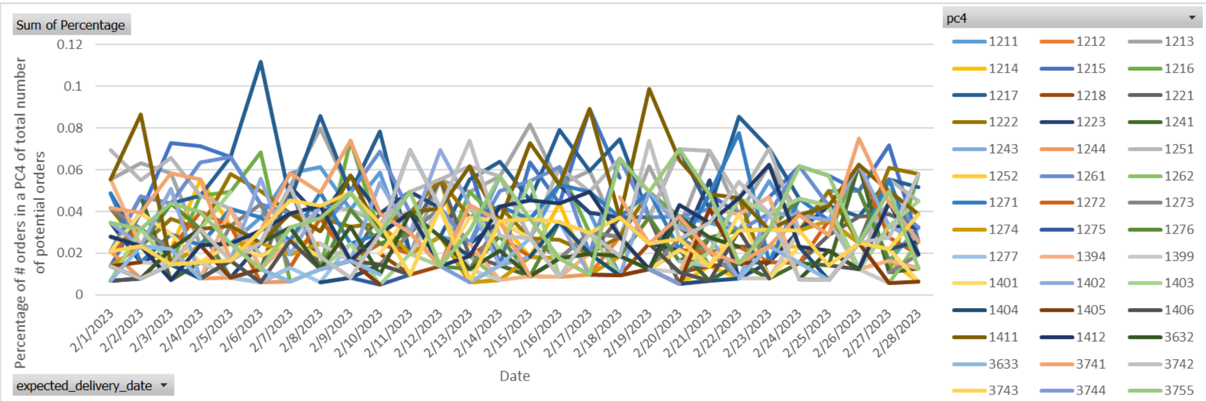

Figure 10 Percentage of potential orders observed in the PC4s in Hilversum

CBB uses a rolling forecast that is updated every day. Every week, they revise the forecast for the coming 4 to 16 weeks to determine the hub capacity. Hence, they include the latest information which comes closer to actual requirements as the moment of ordering approaches (Huang, Hsieh, & Farn, 2011).

Required capacity (5)
CBB plans to deliver all the orders located in the 'optimal levels' and thus the forecast of the potential orders located in the 'optimal levels'. They translate this number of orders into the required capacity. Hence, they divide the number of orders by the historical average number of orders CBB delivers per hour. Subsequently, CBB determines how many delivery men hours are necessary weekly on a bike. Also, they determine the total delivery men hours, which include the depot hours and expected shrinkage hours such as those related to illness and training. Subsequently, for the total biking hours, CBB determines the total number of bike routes they need on the peak day, i.e., the day with the highest number of orders, to ensure these routes can be cycled. The historical averages are static and not updated with new routing information.

We analyze the long-term forecast to find out if CBB is subject to forecast patterns. Figure 11 shows the long-term forecast of bikes for the hubs in The Netherlands per day and per week. It suggests that the weekly capacity necessary for the optimal levels does not fluctuate a lot and that there is a strong daily pattern. Although there is a clear daily deviation, CBB uses the same routing information for all days to determine the daily (and weekly) capacity forecast.

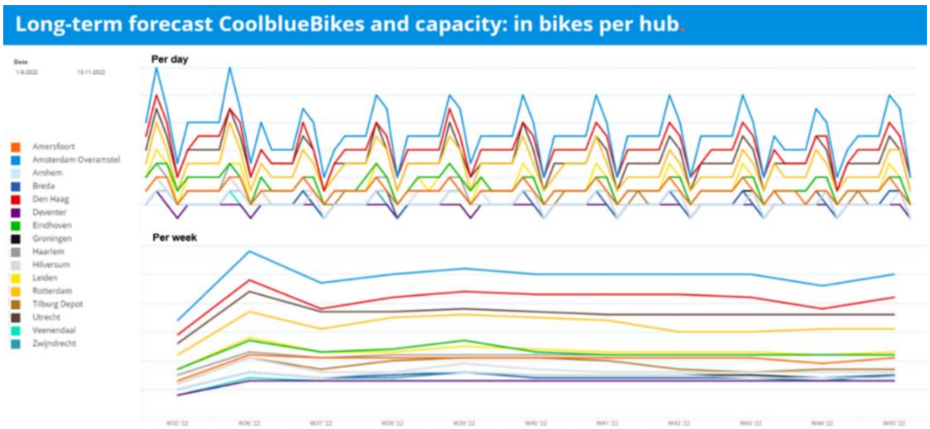

Figure 11 Capacity forecast per hub in the Netherlands per day and per week

Adjust hub capacity (6)
CBB adjusts their capacity of delivery men hours based on min-max contracts and the recruitment targets are adjusted to that. Weekly, they revise the forecast for the coming 4 to 16 weeks and adjust their recruitment target accordingly. More specifically, they align the cumulative minimum contract hours with the lowest demand forecast to prevent overcapacity. This gives the flexibility to plan the delivery men on their minimum or maximum hours. When they are still in shortage of delivery men, they recruit new people. With an average outflow of 10% per month, CBB never experienced a significant excess of delivery men.

We analyze the capacity availability in delivery men hours to learn if currently sufficient capacity is planned. Figure 12 shows the total hours needed by all hubs in the past months compared to the total minimum and maximum contract hours of all hubs. It shows that the minimum contract hours did not exceed the hours needed to deliver in the 'optimal levels'. Therefore, CBB should not have had overcapacity if they have scheduled their capacity right. Contrary, the hours needed exceeded the maximal contract hours in multiple months. This indicates that CBB did not have sufficient capacity to deliver all orders in the optimal levels, which they believe would have been most cost-efficient.

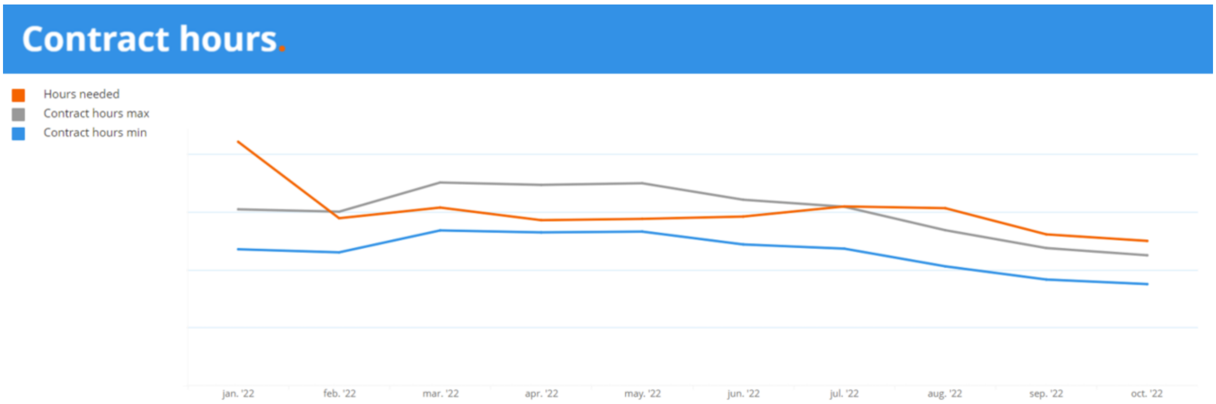

Figure 12 Contract hours of all hubs

The current bike capacity ranges from 2 bikes in small hubs such as Essen in Germany to 16 bikes in larger hubs like Amsterdam in The Netherlands as shown in Figure 13. This capacity is enough to cover the highest demand peak. Therefore, CBB sets the non-necessary bikes to inactive, such that they have no maintenance costs for them. The inactive bikes can be used if bikes are broken down or if there is high demand. They have 0 to 4 inactive bikes per hub. In the exceptional case that the current bike capacity is insufficient, CBB can rent extra bikes.

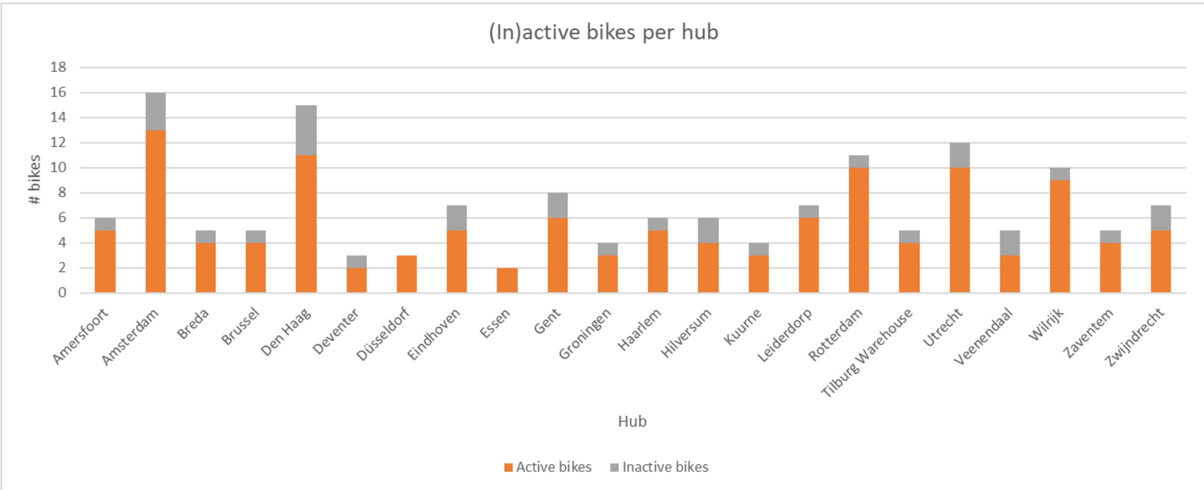

Figure 13 (In)active bikes per hub

2.1.3 Operational level

The operational level schedules detailed activities and their respective timing (Hans, Herroelen, Leus, & Wullink, 2005). It controls the execution and optimal timing of the operations within a planning horizon ranging from even less than one day to a week. Accordingly, CBB aligns their capacity with the forecast and schedules the activities.

Align capacity and forecast (7)

Every week, the planning team of CBB tries to schedule their capacity as close to the forecast as possible. This is dependent on the required capacity, expected shrinkage and availability of delivery men and deployable bikes.

Based on the schedule, CBB compares the scheduled daily capacity with the expected forecast per aggregated level(s). When the forecast is higher than expected, they can decrease the optimal levels. This can be done until two days before the delivery day, due to the current carrier assignment and packaging system in the warehouse. If the forecast is lower than expected, they can increase the 'optimal levels'. This can be done until one day before the delivery day. This results in the final aggregation of level(s) CBB is going to deliver in, known as the active delivery area. CBB in- or decreases the 'optimal levels' to prevent overcapacity because the route planning algorithm only receives orders from the levels included in the active delivery area. CBB does not deal with the consequences of undercapacity, since they outsource all remaining orders to DP.

We analyse the historical data to find out how well the capacity and forecast is aligned on the operational level. Figure 14 shows the number of unused routes, i.e., the available and planned capacity that was not utilized, per week for all 22 hubs. This means that CBB had delivery men and bikes scheduled to deliver orders, but did not deploy them. Hence, CBB had overcapacity in terms of delivery men. CBB must pay for all scheduled delivery men, even if they were not utilized.

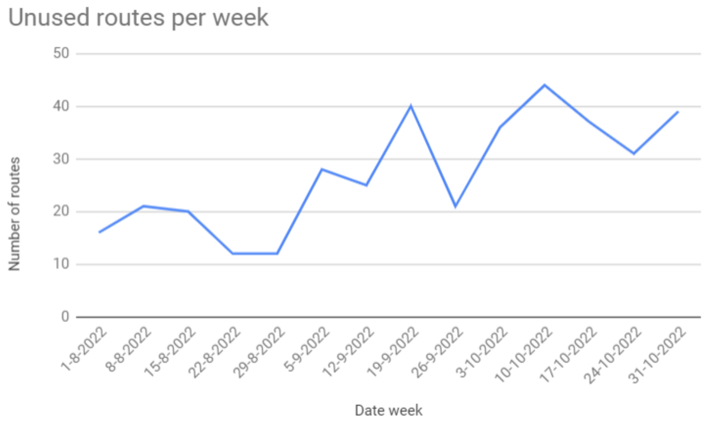

Figure 14 Unused routes per week

2.1.4 Operational route planning

The day before the delivery day, CBB fills the available bike routes given the number of delivery men hours and bikes. When an order is placed, both eligible for CBB delivery and located in the active delivery area, their route planning algorithm determines 1) if the order still can be delivered based on bike capacity and route time, and if so, 2) in which route the order will be delivered. The planned orders in the routes can still switch from a route if this results in a more efficient route. This process repeats until all capacity, i.e., time and volume, is utilized. The travel time of a route is determined based on the actual distance, i.e., taking road works and actual bike roads into account, and the actual speed that can be cycled at specific parts of the route. Note that the time at which an order is placed is important in this process. If an order relatively far away fits in a route, such as Orders 13 & 14 shown left in Figure 15, the order is planned. However, if later orders are placed that are relatively close to, e.g., the already planned orders in the route, they are cancelled because all capacity is already utilized. Hence, once an order is planned, it cannot be cancelled anymore in the current route planning. DP deliver the cancelled orders and all the orders outside the active delivery area.

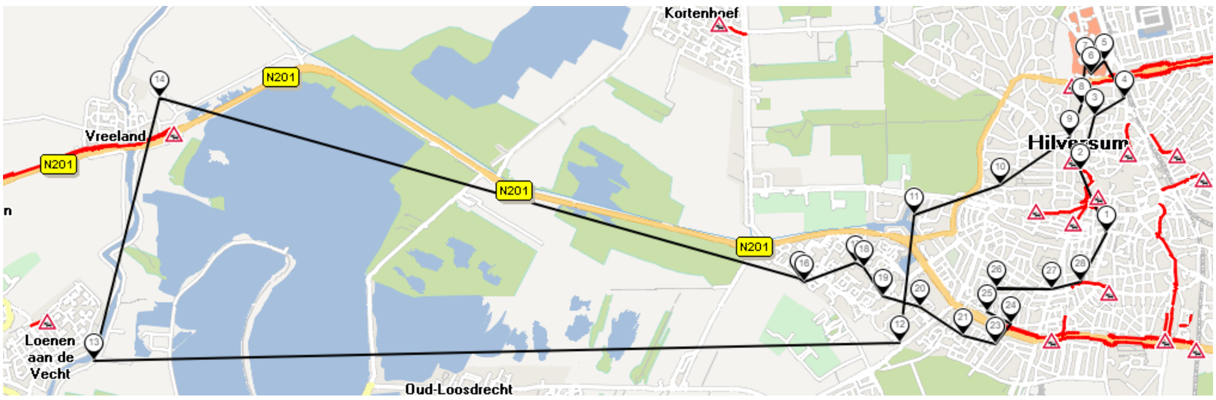

Figure 15 An example of a planned bike route in Hilversum

2.1.5 Process flow diagram

We create a process flow diagram to visualize the capacity planning process and to understand the relations between steps. Figure 16 shows the diagram. We also include the route planning process in the process flow diagram, to show how it interacts with capacity planning. Note that CBB determines the routing data statically, so they do not update it with new routing data. We include the routing data in the diagram to show the relation with capacity planning.

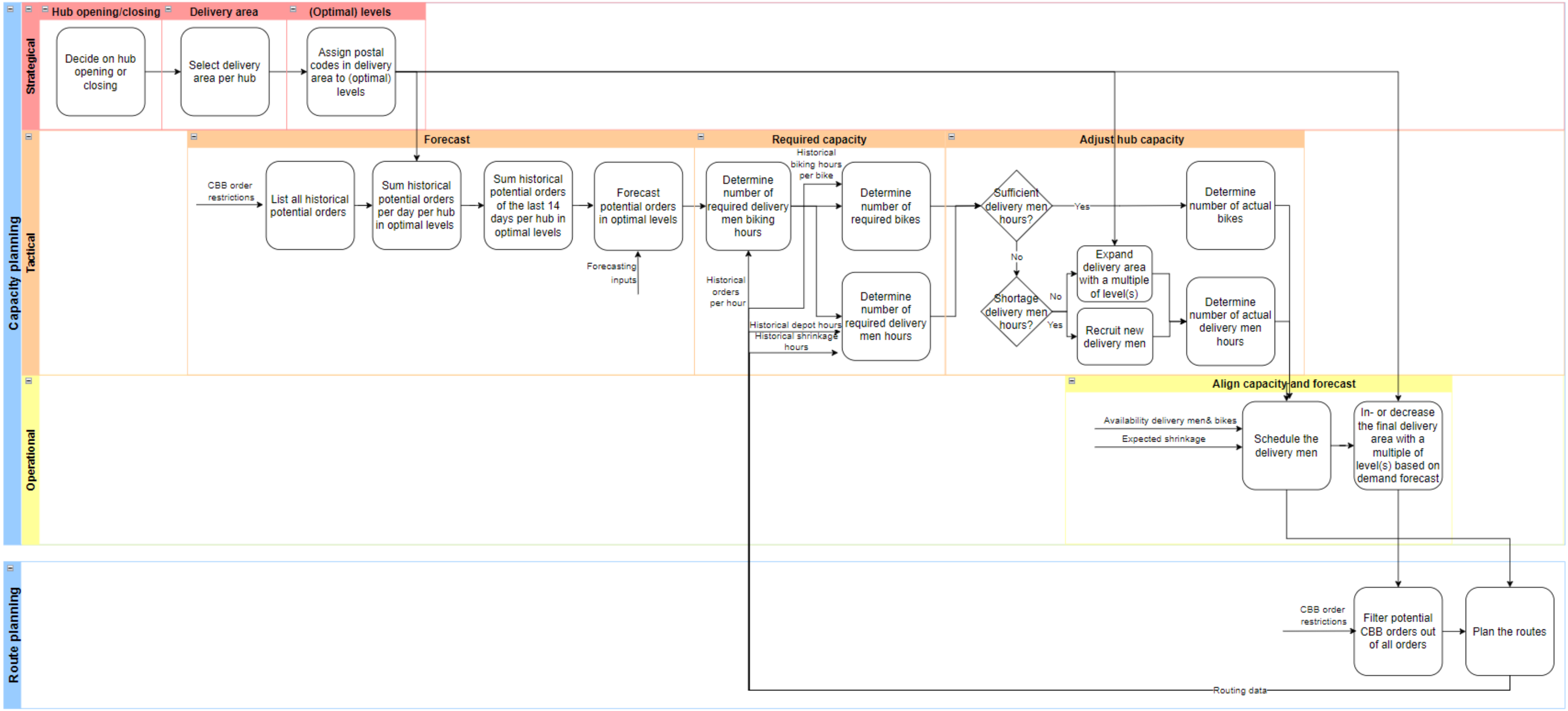

Figure 16 Process flow diagram

2.2 Costs per order

To determine how we can minimize the costs per order, we first need to understand what variables it depends on. The costs per order consist of two types of cost; fixed costs and variable costs per order. For each type, we explain how the costs are made up, what the relation with the number of orders is to define the costs per order and on what variables the costs are dependent. We only include costs that are not subject to interests, taxes, depreciation and amortization, to be in line with the profit measure (EBITDA) of Coolblue.

Fixed costs

Fixed costs are independent of the number of orders. They must be paid by a company, independent of any specific business activity (Hayes, 2022). The fixed costs per order are dependent on the order volume. A volume increase usually means that we can divide the same fixed costs over more orders. This results in a decrease in the fixed costs per order. Hence, when the number of orders converges to infinity, the fixed costs per order converge to zero. Note that the price agreements for these costs are fixed and therefore we do not consider these as influenceable variables.

Variable costs

The variable costs change in proportion to the volume. CBB has three types of variable costs: variable costs per order, variable costs per hour and variable costs per kilometre. The total variable costs increase when the number of orders increase (more travel time, more travel distance and more orders). We cannot influence the price agreements of these variable costs per order, per hour and per kilometre. For example, the hourly salary costs of the bikers are fixed. However, the respective variable costs per order depend on how efficiently and effectively the capacity is utilized (Yu-Lee, 2002). The more efficiently and effectively CBB works, the more they obtain from the price investment and thus the less variable costs per order they have. For example, if CBB delivers more orders with the same number of delivery men, fewer salary costs will be passed on towards the orders. However, the relation between the total variable costs per order and the number of orders to deliver is currently unknown.

Table 2 summarizes our findings of all the costs per order per hub in the Netherlands. We make a distinction between the type of cost, the respective sub-costs (per unit if applicable), the influenceable variables, the month(s) from which we retrieve the cost value and the cost division among the hubs. We refer to Appendix A for the definitions and explanations of the sub-costs. The table shows what variables we need to consider when determining the most cost-efficient capacity. Also, it shows from which period we retrieve the cost value to use in our analysis. We distinguish between the average cost value of the most representative month, December 2022, and the average cost value of 2022. We assume 30 working days per month, i.e., CBB delivers 7 days a week, to account for public holidays. Furthermore, it presents how the costs are divided among the hubs. Costs might be hub specific, averaged over all hubs, divided based on the hub size or a combination of the latter two.

Table 2 Costs per order information

Type of costs	Sub-costs	Influenceable variable(s)	Month(s)	Cost division among the hubs
Fixed costs	Housing	Order volume	2022	Hub specific
	Office salary	Order volume	Dec. 2022	Hub size (50%) and average of all hubs (50%)
	Hub lead	Order volume	Dec. 2022	Hub specific
	Linehaul	Order volume	2022	Hub specific
	Other operating expenses	Order volume	2022	Hub size
Variable costs	Biker salary (hour)	Route efficiency	Dec. 2022	Average of all hubs
	Vehicle maintenance (km)	Route efficiency	2022	Average of all hubs

	Fraud/theft (order)	-	2022	Average of all hubs
	Recruitment (order)	-	2022	Average of all hubs
	Allocated customer service costs (order)	-	2022	Average of all hubs
	Allocated costs process returns (order)	-	2022	Average of all hubs
	Other staffing costs (hour)	Route efficiency	Dec. 2022	Average of all hubs

2.3 Constraints and requirements

CBB cannot deliver every order at every moment at every location. Therefore, we list all constraints and requirements of CBB delivery and how we should use this in our research.

- **Size, weight, volume**
  The orders CBB delivers are subject to size, weight and volume restrictions. This influences the order forecast and route efficiency. Therefore, we should filter the orders that do not suffice these restrictions when determining the total number of potential orders.

- **Delivery safety**
  Some addresses are safe during the day, but unsafe during the evening. Likewise, some addresses are less reachable, or considered unsafe during winter, whilst not in summer. CBB does not want to risk the safety of their delivery men. Therefore, they include only safe and reachable postal codes in the evening and/or winter routes, which influences the order forecast and route efficiency. Therefore, we should include these adjustments when determining the total number of potential orders.

- **Time span**
  CBB aligns the hub and required capacity for a 16-week period. They check this alignment, which includes the peak day, every week. Therefore, we should determine the most cost-efficient capacity on a daily and weekly level for a time span of 16 weeks.

- **Hub**
  Every hub has its predefined delivery area. This area is fixed. The area characteristics like the number of inhabitants differ. Therefore, the bottleneck of the hub can be different at every hub. Consequently, we should analyze each hub individually to determine what the most cost-efficient capacity level is.

- **Dynamism**
  Coolblue is a dynamic company, changing its work routines often to improve or to adapt to growth. The determination of the most cost-efficient capacity should comply with this dynamism. It must be able to determine the optimal capacity also with changing parameters, for example, obtained from changing costs.

- **Model output**
  CBB defines the required capacity as the number of delivery men hours and the number of bikes necessary to deliver the desired orders. The goal of the model is to find the weekly required capacity at the tactical level, which is evaluated on daily basis to consider peak days. Although on the operational level, CBB deploys and pays for full tours, i.e., the maximal time of a tour (see Section 2.1.2), they do not define the required capacity as the number of bike tours. This would imply rounding the daily delivery hours, which gives a distorted view of the weekly available work. Accordingly, the chance of overfitting is high when deciding at the tactical level the exact number of tours necessary at the operational level, i.e., the point forecast is still likely to deviate. Furthermore, CBB wants to be able to express the capacity relative to the potential

order forecast. Hence, CBB wishes the daily number of orders to deliver as model output.

- **Postal codes**
  Currently, the system is designed such that CBB can only in- or decrease the delivery area with a multiple of subareas. CBB defines these subareas as the area belonging to the PC4. Therefore, we should design a tool that is at least able to translate the capacity advice to PC4s.

2.4 Conclusions

This chapter introduced the current situation of CBB. We explained the current capacity planning process, and how the deployable capacity depends on the number of forecasted potential orders and the static ‘optimal’ delivery levels. We showed how the operational route planning depends on the deployed capacity. Next, we explained that the costs per order consist of fixed costs, and variable costs per order, per hour, and per kilometre. Lastly, we introduced the constraints and requirements of the capacity tool. Hence, we should ensure that the tool is dynamic, that the deepest level of detail is on PC4 level, and that it finds the optimal daily number of orders to deliver.

3 Literature review

This chapter reviews the literature in the area of vehicle routing problems, order selection and capacity planning, and simulation. Section 3.1 introduces the vehicle routing problem and the fleet size planning problem. Next, Section 3.2 elaborates on the integration of order selection and capacity planning, to learn how a subset of orders can be selected to decide upon capacity deployment under routing considerations. Section 3.3 describes methods to obtain the solution value of a vehicle routing problem and Section 3.4 introduces Monte Carlo simulation. Finally, Section 3.5 concludes this chapter and states the contribution to the literature.

3.1 Vehicle routing problem

The vehicle routing problem (VRP) is concerned with finding the optimal route(s) to visit all customers with a restricted number of vehicles starting and ending at the depot (Caric & Gold, 2008). Caric & Gold state that the most common objective is to minimize the transportation costs related to these routes. Furthermore, they argue that the transportation cost can be improved by reducing the total travelled distance and by reducing the number of required vehicles.

We formulate the capacitated VRP (CVRP) with maximum trip duration constraint based on Bodin, Golden, Assad & Ball (1983) to understand the problem structure, i.e., the optimal capacity to deploy depends on the routes and the respective transportation costs. The binary decision value x_{ij}^{v} (see Constraint 3-7) assumes a value of 1 if there is a route going from customer i to customer j with vehicle v, for $i, j \in N$ and $v \in V$.

$$\min \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{v=1}^{V} d_{ij} * x_{ij}^{v} \qquad \text{3-1}$$

Subject to:

$$\sum_{j=1, j \neq i}^{N} \sum_{v=1}^{V} x_{ij}^{v} = 1 \quad (i = 2, \dots, N) \qquad \text{3-2}$$

$$\sum_{i=1}^{N} x_{ip}^{v} - \sum_{j=1}^{N} x_{pj}^{v} = 0 \quad (\forall\, v \in V, p = 1, \dots, N) \qquad \text{3-3}$$

$$\sum_{i=1}^{N} \sum_{j=1}^{N} k_i * x_{ij}^{v} \leq K \ (\forall\, v \in V) \qquad \text{3-4}$$

$$\sum_{i=1}^{N} \sum_{j=1}^{N} s_i * x_{ij}^{v} + \sum_{i=1}^{N} \sum_{j=1}^{N} t_{ij}{}^{v} * x_{ij}^{v} \leq T \quad (\forall\, v \in V) \qquad \text{3-5}$$

$$y_i^{v} - y_j^{v} + N * x_{ij}^{v} \leq N - 1 \quad (\forall\, v \in V, (i,j) \in N : i \neq j) \qquad \text{3-6}$$

$$x_{ij}^{v} \in \{0,1\}, y_i^{v} \in \mathbb{R}_{\geq 0},\ \ y_j^{v} \in \mathbb{R}_{\geq 0}, (\forall\, (i,j) \in N, v \in V) \qquad \text{3-7}$$

The objective function 3-1 minimizes the distance travelled by all vehicles. Constraint 3-2 ensures that every customer is visited exactly once. Constraint 3-3 guarantees route continuity, i.e., if a vehicle arrives at a customer, it should also depart from this customer. Constraint 3-4 ensures that every vehicle carries at most their capacity K and Constraint 3-5 is the maximum trip duration T constraint. This considers the travel time $t_{ij}{}^{v}$ and stop time s_i of all the customers. Constraint 3-6 is a subtour elimination constraint. Finally, Constraint 3-7 displays the nonnegativity constraints.

VRP fleet size

The total number of vehicles in a VRP is known as the VRP fleet size. The solution of the VRP shows the number of vehicles required: for every route a vehicle is necessary. The traditional VRP assumes that a fixed number of similar vehicles are available. Contrary, there are variations of the VRP that choose the vehicles to deploy. For example, the fleet size and composition vehicle routing problem (FSCVRP) chooses a number of vehicles in the fleet from a pool of different types of vehicles (Gheysens, Golden, & Assad, 1986). Gheysens, Golden, & Assad explain that this problem decides upon the composition of a fleet of vehicles and constructs an associated set of routes for these vehicles that deliver goods to a set of customers with known demands. Here, the objective is to minimize the sum of the fixed costs arising from vehicle acquisition, and variable costs originating from the delivery routing. However, in many real-world situations, the set of customers, their location and their demand are uncertain (Gendreau, Laporte, & Seguin, 1996). Hence, the VRP becomes a stochastic VRP (SVRP) (Berhan, Beshah, Kitaw, & Abraham, 2014). Accordingly, List et al. (2003) propose a solution to the fleet sizing problem when there is uncertainty in future order information. They trade-off the fleet ownership costs, fleet operating costs and penalty costs when demand is not met at the requested time. They focus on a robust solution to the fleet sizing problem in terms of good performance under a wide variety of order uncertainty. Couillard (1993) designs a decision support system to plan fleet size and mix under uncertainty. This system generates fleet size and mix alternatives based on the order forecast, evaluates them and selects the best alternative. Furthermore, Shyshou, Gribkovskaia and Barcelo (2010) generate real-world scenarios to solve the fleet size planning problem. Here, they model possible scenarios of the VRP concerning the uncertainty, i.e., they evaluate multiple possible scenarios with different customer sets, locations and demands. Next, they assess the optimal number of vehicles to deploy based on these scenarios. A more long-term fleet size planning decision in delivery routing problems is whether to maintain a private delivery fleet, employ external carrier services, or combine both options (Pilcher, 1990). Pilcher argues that this decision differs from the decision on how to serve a particular day's deliveries, because for this the order information, such as the exact location, is known with greater certainty. To solve this problem, she proposes to divide the delivery area into sectors with a random number of delivery locations and determine the costs of assigning a sector to a private or external fleet.

3.2 Order selection and capacity planning

We review the literature regarding order selection and capacity planning to understand how orders can be selected, how capacity can be planned, and how order selection is integrated into capacity planning. A selection of orders can be made for several reasons at several decision levels. Hence, various hierarchical planning models distinguish the strategic, tactical and operational decision levels to break down capacity planning into more manageable parts (De Boer, 1998).

This section introduces the three hierarchical planning levels. Furthermore, we describe several order selection and capacity planning methods for each decision level, stemming from VRP literature and other research fields to benchmark from. We focus mainly on the methods at the tactical and the operational level since order selection is hardly discussed at the strategic level. This literature review will provide insight into the similarities and gaps between our research problem and the research problems discussed in the literature.

3.2.1 Strategic level

According to the framework of Hans et al. (2005), strategic capacity management addresses the strategic resource planning. A long-term plan is made that guides an organization towards its business goals. The objective is to determine the organization's global resource capacity levels (De Boer, 1998). Often, planners attempt to devise a master plan that represents an optimal solution for the forecasted conditions (Mierzejewski, 1998). Based on this optimal solution, organizations determine when and by how much the capacity levels should change to fulfil the forecasted demand (Olhager, Rudberg, & Wikner, 1999). Generally, these capacities take a long time to change, such as modifying a transportation network or choosing a (new) hub location (SteadieSeifi, Dellaert, Nuijten, Van Woensel, & Raoufi, 2014).

3.2.2 Tactical level

Tactical planning is a middle-level activity connecting strategic planning and operations control (Bushuev, 2014). The main goal is to allocate resources such as capacity over a medium-range planning horizon. Order selection given the capacity determined on the strategical level is a typical tactical level activity in traditional manufacturing and project environments (Hans, Herroelen, Leus, & Wullink, 2005) (Brahimi, Aouam, & Aghezzaf, 2015). Here, a rough-cut capacity planning (RCCP) can help to make such order acceptance decisions by simulating an initial plan (De Boer, 1998). Boer explains that the method determines if the orders can be completed in time and with the available capacity. Usually, the goal is to make a plan that fits all orders with the fixed capacity. If impossible, organizations can decide to temporarily extend capacity with for example overtime or postpone the due dates of customers. This flexibility in time and capacity supports planning in making a trade-off between expected delivery performance and the expected costs of exploiting flexibility by using non-regular capacity (De Boer, 1998). However, we do not determine the minimal (extra) capacity to fit all orders within a time frame, but we select a subset of orders to determine capacity. Additionally, we cannot postpone due dates, because every rejected order is immediately outsourced to DP.

Lin & Chang (2008) present an order selection method with limited production capacity. They prioritize the orders and maximize the total production quantity when the order demand exceeds capacity. However, like much literature, this model has the underlying assumption that more orders are preferable. Even when there is sufficient capacity to fulfil the entire set of orders, Brahimi, Aouam & Aghezzaf (2015) state that it is not always optimal to accept all the orders. They argue that the revenue from an additional order should at least offset the variable production cost plus the shadow prices of the capacity constraints that take into account workload. Geunes, Romeijn & Taaffe (2002) identify this order selection flexibility. They state that the organization must determine the most profitable set of orders to fulfil, given a set of orders for production along with fixed plus linear production costs and variable holding costs in every period. Their solution model, however, assumes that orders can be held in inventory, while we outsource (the delivery of) all unselected orders. Also, the production costs are linear, while we should take the efficiency of delivery/production into account, i.e., we do not assume a linear relation between the number of orders and the variable costs. In addition, order selection is executed to optimally use the fixed capacity instead of determining the optimal capacity based on the most profitable set of orders.

A simulated initial plan is also used in delivery time slot management literature. For instance, Herandez, Gendrau & Potvin (2017) use estimates of demand and stop time in each postal code zone to construct a tactical routing plan, i.e., an approximation of the operational routing plan. Here, a heuristic selects the most profitable time slots for each zone. Klein et al. (2018) approximate the opportunity costs of selecting a customer with a mixed integer linear programming model that maximizes the expected profit. These applications use historical data to predict customer information like demand and geographical location. They consider multiple combinations of routes by approximating the costs of adding a customer to a route of a time slot. However, all customers must be served in the given time slots, while we have to select a subset of customers to serve. Cleophas & Ehmke (2014) do select a subset of customers. They use vehicle routing procedures to decide whether to accept or reject sequentially processed fictitious requests in delivery tours for a given fleet of vehicles. However, they use the DYN algorithm as proposed by Ehmke & Campbell (2014), which aims to maximize the number of orders delivered instead of the profit obtained from a subset of delivered orders.

3.2.3 Operational level

Decisions on the operational level deal with short-term operational and scheduling problems (Bushuev, 2014). Within a given time frame, the goal is to schedule all the work given the assigned workload and available (capacity) resources (Hans, Schutten, & Maan-Leeftink, 2021). SteadieSeifi et al. (2014) distinguish this activity into itinerary planning problems, i.e., the online, real-time optimization of routes and/or schedules and offline resource allocation. The scheduling of the delivery men and the routes they will travel, i.e., a VRP solution, are examples of resource allocation. Some VRP variants explicitly require some form of customer (order) selection or prioritization (Akkerman & Mes, 2022). These variants integrate order selection and route planning.

The inventory routing problem (IRP) combines the decision on when to replenish the customers' inventories, how much product to deliver, and in which way to route the vehicles that execute the delivery (Mes, Schutten, & Perez-Rivera, 2014). Hence, the customers to replenish are selected ensuring that all customers do not experience a stock-out (Bertazzi, Savelsbergh, & Grazia-Speranza, 2008). Customers are selected based on a heuristic like cheapest insertion (Mes, Schutten, & Perez-Rivera, 2014), or by prioritizing big orders or lowest storage first (Roldán, Basagoiti, & Coelho, 2016). Although capacity decisions are made, i.e., replenishing can be advanced if it is advantageous, the main objective of the problem is the routing and not capacity determination.

Likewise, the production routing problem (PRP) jointly optimizes production, inventory and transport routing planning decisions by integrating a lot-sizing problem to determine production amounts and a VRP to determine delivery routes (Díaz-Madroñero, Peidro, & Mula, 2015). Díaz-Madroñero, Peidro, & Mula (2015) stress that such problems should correspond with and be solved at the tactical decision level as well, to more accurately and better synchronize the two planning processes (Amarim, Belo-Filho, Toledo, Almeder, & Almada-Lobo, 2013). However, in most of these cases, transportation is considered a product distribution resource (i.e., supplement to production planning) (Lagemann & Meier, 2014) instead of the source to determine the (production) capacity on.

Another variant that selects a subset of customers to serve is the vehicle routing problem with profits (VRPP). Here, the decision should be taken which customers to serve and how to cluster these customers in different routes and order the visits in each route (Archetti, Grazia-Speranza, & Vigo, 2014). This method may leave cost-unattractive customers unvisited (Akkerman & Mes, 2022) and can for example be applied when unprofitable customers are outsourced if the total demand is greater than the whole capacity (Chu, 2005). In this case, Chu (2005) selects the customers with the highest outsourcing costs until all capacity is satisfied. Kim, Li & Johnson (2012) propose an augmented large neighbourhood search method to solve the VRPP. They construct an initial solution based on a greedy heuristic, where they select the next feasible customer with the smallest distance divided by reward. Afterwards, three improvement algorithms perturb the initial solution to find the solution with the highest profit. In both solution approaches, customer selection is executed to optimally use the fixed capacity instead of determining the optimal capacity based on the most profitable set of customers.

3.2.4 Discussion

In summary, most literature focuses on scheduling as many orders as possible given the fixed capacity, instead of determining the most profitable subset of orders to determine capacity on. However, the need for optimal order selection independent of the capacity is recognized by, e.g., Brahimi, Aouam & Aghezzaf (2015) and Geunes, Romeijn & Taaffe (2002).

Although order selection is discussed for a different purpose, i.e., to optimally use the predetermined capacity instead of determining the optimal capacity based on the most profitable set of orders, the logic can still be used as inspiration. At the tactical level, an initial plan is established to determine the feasibility and costs of the selected orders. This plan is an approximation of the operational plan. It is determined based on estimates and assumptions of order information, usually based on historical information. Next, it selects the subset of orders with the best solution value, e.g., the cost of production in manufacturing or the tour in routing. At the operational level, several VRP variants select orders based on the costs of the solution value belonging to the subset of orders. Most of the time, a heuristic constructs a solution. Often, all order information is known at this decision level and therefore, the (route) costs result from explicit routing decisions.

We further discuss the approximation of the VRP distance in Section 3.3. Also, we discuss the simulation of the operational routing plan, i.e., a tactical routing plan, in Section 3.4.

3.3 VRP distance estimations

VRPs are NP-hard (Konstantakopoulos, Gayialis, & Kechagias, 2020), and can therefore not be solved to optimality in polynomial times (Dantzig & Ramser, 1959). The algorithms to solve the VRP are categorized into exact and approximate methods as presented in Figure 17. Here, the approximate methods are further distinguished into heuristics and metaheuristics. To limit computational complexity, several options for exact and approximate methods exist. Options are to decompose the problem into smaller sub-problems (Lalla-Ruiz & Voß, 2020), to balance intensification and diversification such that one finds a feasible solution in an acceptable timescale with metaheuristics (Yang, Deb, & Fonf, 2013), or to reduce the decision space with priority rules (Doan, Bostel, & Hà, 2021). However, when many combinations of order subsets must be computed, these methods are still quite time-consuming.

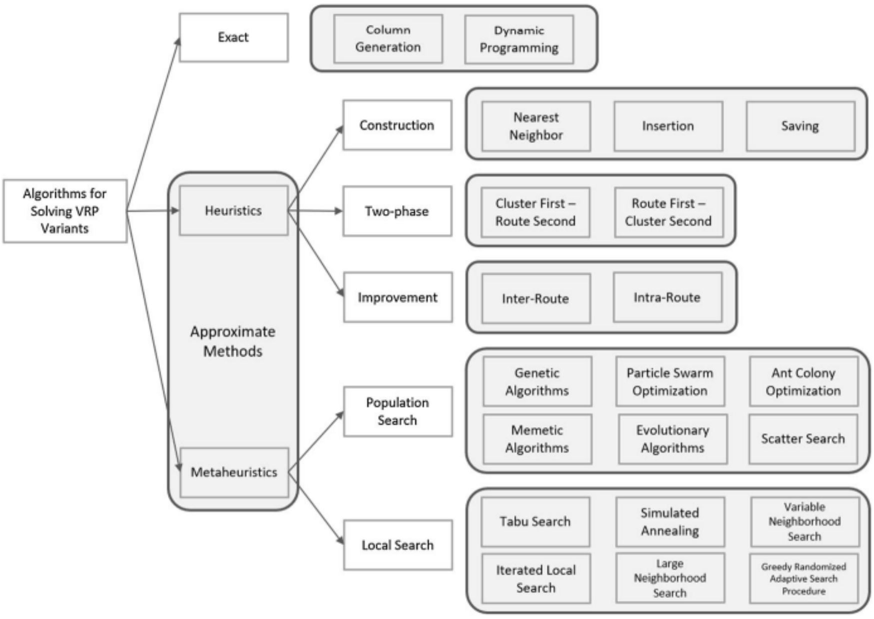

Figure 17 Classification of algorithms for the VRP. (Konstantakopoulos, Gayialis, & Kechagias, 2020)

Alternatively, estimating the unknown solution value of the problem, prior to solving, can help to reduce the decision space by excluding unattractive customer subsets according to Akkersman & Mes (2022). They argue that it can help solve logistic problems where it is necessary to select a subset of customers to serve, e.g., when there is insufficient capacity or when it is allowed to leave cost-unattractive locations unvisited. They state that the number of possible subsets, with subset size r, from a set of customers with size n, equals $n!/r!(n-r)!$. Hence, approximation models can help make customer selection decisions when there is high demand on computational times.

Much of the existing literature considers the (variable) costs of a VRP tour to be roughly equivalent to the total travelled distance (Nicola, Vetschera, & Dragomir, 2018). The variable costs mostly originate from the travelling time of each route (Konstantakopoulos, Gayialis, & Kechagias, 2020). As a result, variable costs are affected by the length and duration of the route. Therefore, estimating the total travelled distance of a subset of orders can be used to estimate the respective (variable) costs.

We distinguish between two types of VRP estimation methods prior to solving. Section 3.3.1 elaborates on mathematical approaches and Section 3.3.2 on machine learning estimations.

3.3.1 Mathematical approaches

Various studies have been done towards the mathematical approaches of the optimal solution value, i.e., total travelled distance, of a VRP. Beardwoord, Halton & Hammersley (1959) propose one of the first approximation models. They show that for the most basic VRP version, the travelling salesman problem (TSP), the total travelled distance asymptotically converges to $c\sqrt{NA}$ when $N \to \infty$ for a set of N customer in the convex area with surface area A, where c is a constant. Afterwards, many authors identify the need to include the shape of the area or the distance between customers. Chien (1992) modifies the area into the smallest area covering all customers. Also, he includes distance measures between customers and the depot. Hindle & Worthington (2004) consider the customer distribution. Contrary, Çavdar & Sokol (2015) develop a customer distribution-free approximation that can make predictions when the distribution of the customers' coordinates is unknown. Basel and Willemain (2001) demonstrate that the optimal tour length of a set of TSP instances can be predicted with the standard deviation of random tour lengths. Here, they start with an empty tour and randomly select an eligible stop until the tour is complete. They repeat this random tour generation, and once sufficient random tours are generated, they define the relation between the standard deviation of the length of these random tours with the (known) optimal tour length. Kou, Golden & Poikonen (2022) extend this relationship and reveal the asymptotic linear relationship between the standard deviation and the $\sqrt{NA}$ predictor discovered by Beardwoord, Halton & Hammersley (1959). In further research, they enhance the estimation ability of their model by also considering the mean of random feasible solutions (Kou, Golden, & Poikonen, 2023).

Webb (1968) was the first to consider the CVRP. He studied the correlation between total travel distance and the distance between the customers and the depot. Daganzo (1987) developed a well-known estimation of the CVRP, which takes the capacity of the vehicles into account. Given *N* customers, an area of *A*, an average distance *r* between customers and the depot and maximal *Q* customers that can be served by a vehicle, he estimates the tour length with Equation 3-8:

$$CVRP\ (N) = \frac{2rN}{Q} + 0.57\sqrt{NA} \qquad \textit{3-8}$$

Based on this estimation, Robusté, Estrada & López-Pita (2004) propose adjustments to include the shape of the area, i.e., square, rectangular, circular and elliptic zones. Furthermore, Figliozzi (2008) studied six approximations of the average VRP tour length when there is variability in the number, level and locations of customer demands. The best-proposed approximation is $VRP = k_l * \frac{N-M}{N} * \sqrt{AN} + k_b * \sqrt{\frac{A}{N}} + k_m * M$, where *N* is the number customers, *A* the area, *M* the number of vehicles, and where the coefficients of these independent variables are estimated by linear regression.

3.3.2 Machine learning estimations

Instead of using mathematical approaches, also machine learning models have been utilized to approximate the VRP distance. Here, the literature refers to features instead of independent variables that describe the dependent variable, i.e., the VRP distance. Arnold and Sörensen (2019) use data mining techniques to classify the relation of VRP features with the solution value. Kwon, Golden & Wasil (1995) define estimators of the optimal TSP tour length using linear regression and neural networks, which produce reasonably good estimates. Nicola, Vetschera & Dragomir (2018) describe the VRP with features such as distances, capacities and demands. Hence, with the use of forward stepwise selection and backward stepwise regression, they select the features that estimate the total travelled distance. They achieve good approximations and outperform previous models. Akkerman (2021) uses random forests and neural networks to predict the final routing distance. In further research, Akkerman & Mes (2022) use more distinct methods of linear regression, random forest regression, and neural networks to approximate the VRP distance. The use of these methods and the inclusion of additional features improve the distance predictions compared to Nicola, Vetschera & Dragomir (2018) and Figliozzi (2008).

We elaborate on the machine learning models random forests regression in Section 3.3.2.1 and lightweight gradient boosting machines in Section 3.3.2.2, as we find in Section 4.3.2 that these models have the best performance on our data. Section 3.3.2.3 explains how to assess the model performance of the models. Both models are regression models that estimate the relationships between one or more independent variables and a quantitative dependent variable with a set of statistical techniques. The machine learning literature calls the independent variables, the features and the dependent variable, the response.

3.3.2.1 Random forests regression

Random forests regression (RFR) is a supervised machine learning algorithm that combines multiple decision trees to predict quantitative responses. A decision trees algorithm builds a tree-like model of decisions and their possible consequences based on the training data (James, Witten, Hastie, & TIbshirani, 2021). According to James, Witten, Hastie, & TIbshirani (2021), the process of building a regression decision tree consists of roughly two steps.

1. Divide the feature space, which consists of all possible feature values, into distinct and non-overlapping regions. To do so, the algorithm first selects a feature from the dataset that best separates the data into distinct groups based on their response. This feature is used as the root node of the tree, i.e., the node that starts the decision tree. The algorithm then splits the datasets into data subsets based on the values of the selected features, creating intermediate nodes and more specific regions. Each final region represents a branch or path in the tree, and each path leads to a leaf node representing the response. The algorithm recursively applies the same process to each subset of data until a stopping criterion is met. This criterion can be based on a pre-specified maximum depth of the tree, a minimum number of data points per leaf node, or other criteria.
2. To make a prediction for a new data point, the algorithm traverses the tree from the root node to a leaf node, following the path based on the values of the input features. The prediction is then the mean of the response values in the region.

The goal of a decision tree algorithm is to find regions that minimize the difference between the predictions and the response values of the training data. Decision trees tend to overfit the data when the tree is too deep or complex and possibly, poorly generalize on unseen data.

RFR builds multiple decision trees to create a predictive model. Contrary to the traditional decision tree, a random sample of m features from the total set of p features is chosen as model features. Usually $\sqrt{p}$ features are used per split (James, Witten, Hastie, & TIbshirani, 2021). The splits are repeated until the decision tree reaches the leaf node, i.e., the response. The random forest model trains multiple random decision trees to create a forest of decision trees. To make a prediction for a new data point, the algorithm feeds the data point through each decision tree and determines the average of each tree to obtain strong, final predictions (Breiman, 2001). The construction of the (trees in the) final random forest depends on the hyperparameters. These are selected before training an RFR model, and determine the behaviour of the model. Common RFR hyperparameters are the maximal number of features considered for splitting in a decision tree, the minimum number of data points per leaf in a decision tree, the number of decision trees in the forest and the maximal depth of each decision tree. Figure 18 shows an example of a Random Forest with three decision trees and a maximal depth of three nodes from the root node to the leaf nodes in the tree.

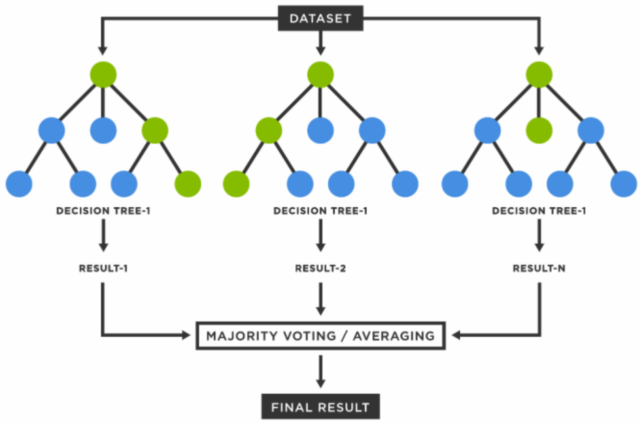

Figure 18 Example of Random Forests with 3 decision trees (TIBC, 2023)

The main advantage of RFR is that it can handle high-dimensional data with many features and can capture complex nonlinear relationships between features and the response variable. Additionally, RFR is robust to overfitting, meaning it can generalize well to unseen data (James, Witten, Hastie, & TIbshirani, 2021).

3.3.2.2 Lightweight gradient boosting machines

Lightweight gradient boosting machines (LGBM) is a highly efficient gradient boosting decision tree algorithm proposed by Ke et al. (2017). A gradient-boosting decision tree (GBDT) algorithm combines the predictions of multiple weaker models based on training data, to create a stronger overall model. The trees grow sequentially: each tree is grown by utilizing information obtained from previously grown trees (James, Witten, Hastie, & TIbshirani, 2021). Again, James, Witten, Hastie, & TIbshirani (2021) distinguish roughly two steps to build a gradient-boosting decision tree.

1. Fit a decision tree on the training data. Subsequently, evaluate the errors and train another decision tree on these errors. Then, in every iteration, the algorithm fits the gradient of the loss function, i.e., the error measure, to update the model such that the loss function is minimized (Ke, et al., 2017). The gradient is the derivative of the loss function concerning the predicted response value. This means that the algorithm focuses on the examples that were incorrectly predicted by the previous models and tries to improve the predictions for those examples in the next iteration. To control the rate at which the boosting learns, the algorithm adds the gradient multiplied with a learning rate of typically 0.01 or 0.001 to the current parameters of the model. The algorithm repeats the error evaluation and training of decision trees on these errors.
2. To make a prediction for a new data point, the algorithm combines the predictions of all weak models by taking their weighted sum. Each tree's weight is determined by its contribution to the overall performance of the model.

The goal is to minimize the loss function by adjusting the parameters of the decision tree. The iterative algorithm can be prone to overfitting if the number of trees or iterations is too large, or if the models are too complex. Also, Ke et al. (2017) state that, for every feature, all data instances need to be scanned to estimate the information gain of all the possible split points. Therefore, their computational complexities will be proportional to both the number of features and the number of instances, which can be very time-consuming.

LGBM reduces the number of data instances and the number of features, and thus the training speed and memory consumption of GBDT. For data reduction, Ke et al. (2017) use the technique Gradient-based One-Side Sampling (GOSS) to select only the most important data points. The larger the gradient of the loss function, the more the data points contribute to the learning process. Hence, GOSS keeps the data instances with larger gradients and randomly samples what data instances with smaller gradients to keep, to retain the accuracy of the estimation. For the reduction in the number of features, they use Exclusive Feature Bundling (EFB) which bundles the mutually exclusive features into a single feature. According to Ke et al. (2017), mutually exclusive features never have a nonzero value at the same time.

LGBM uses a histogram-based algorithm to identify the optimal split points, i.e., the regions, and the number of features that are considered per split in the decision tree. Hence, it creates a histogram of each feature and divides it into discrete bins. A higher number of bins allows a more detailed data splitting and can improve the model accuracy, but also increases the computational time. However, instead of splitting all the data, Ke et al. (2017) use GOSS to select only the data points with the highest gradients for splitting. When the decision tree is fit, the next steps of the gradient boosting decision tree algorithm as described above are executed. However, the gradient of the loss function is only computed for the data points with the highest gradients obtained from GOSS. To make a prediction for a new data point, LGBM traverses the decision tree and sums up the predicted values of the leaf nodes.

3.3.2.3 Model performance

Various performance metrics exist to assess the predictive performance of a machine learning model. These metrics quantitatively evaluate how well the predications match the observed data (James, Witten, Hastie, & TIbshirani, 2021). The three most popular error measures are the root mean squared error (RMSE), the mean absolute error (MAE) and the mean absolute percentage error (MAPE) (Botchkarev, 2019). The RMSE assesses the error magnitude and penalizes large errors through the square. The MAE measures the average error without under- or overprediction consideration. The MAPE represents the average absolute error relative to the observed data. The disadvantage of the RMSE and MAE is that interpretation is harder when the magnitude of the response can vary. Relative errors can provide a better interpretation of how well the evaluated forecasting method performs compared to another method (Chen, Twycross, & Garibaldi, 2017). The three relative error performance metrics are given by:

$$rRMSE\ (\%) = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(Actual_i - Prediction_i)^2}}{\overline{Actual}} * 100\% \qquad \text{3-9}$$

$$rMAE(\%) = \frac{\frac{1}{n}\sum_{i=1}^{n}|Actual_i - Prediction_i|}{\overline{Actual}} * 100\% \qquad \text{3-10}$$

$$MAPE\ (\%) = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{Actual_i - Prediction_i}{Actual_i}\right| * 100\% \qquad \text{3-11}$$

Additionally, R^2 is a well-known statistical measure of fit that indicates the proportion of variance of a response that is explained by predictors in a regression model (Ferando, 2021). Thus, an R^2 of 1 means that all observed variation can be explained by the predictors. The R^2 increases when more predictors are added to the model. The adjusted R^2 modifies for this. Hence, it prevents overfitting and ensures that models with a different number of predictors can be compared (James, Witten, Hastie, & TIbshirani, 2021). The R^2 and the adjusted R^2 are calculated by Equation 3-12 and Equation 3-13 respectively:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(Actual_i - Prediction_i)^2}{\sum_{i=1}^{n}(Actual_i - \overline{Prediction})^2} \qquad \text{3-12}$$

$$Adjusted\ R^2 = 1 - \frac{(1 - R^2)(Sample\ size - 1)}{Sample\ size - Nr. predictors - 1} \tag{3-13}$$

3.4 Monte Carlo simulation

Simulation is a technique that uses computers to imitate real-world processes (Law, 2015). Subsequently, simulation is used to evaluate a process, e.g., order delivery, numerically and to gather data to estimate the desired true characteristics of the process.

A Monte Carlo simulation is a sampling-based simulation, in which a stochastic problem or calculation is solved multiple times, to estimate the probability distribution of the result (Heijungs, 2020). The simulation is suitable for systems that are static and do not involve time evolution (Lawson & Leemis, 2008). Hence, the passage of time during a simulation run does not play a significant role. The entities, i.e., the objects or actors that are simulated, in the Monte Carlo simulation do not influence each other and are thus independent of each other. Typically, the inputs of the simulation are unknown and therefore based on random number generators and probability distributions (Law, 2015). The output distribution is then used to inform decision-makers about the characteristics of the result, such as the mean value, the standard deviation and quantiles (Heijungs, 2020). Hence, a Monte Carlo simulation is used to understand the impact of uncertainty (Kenton, 2022) and to assess the robustness of the output.

3.5 Conclusions and contributions to the literature

This chapter reviewed the literature regarding vehicle routing problems and ways to find the optimal solution value, order selection and capacity planning, and simulation. First, we introduced the capacitated vehicle routing problem with maximal trip duration constraint. Second, we found that little to no research has been done towards tactical capacity determination based on the delivery outsourcing of unprofitable orders. Order selection, if done, usually has the goal of optimally using the predetermined capacity instead of determining the optimal capacity based on the most profitable set of orders. Additionally, many methods assume that all order information is known when order selection decisions are made. Third, we elaborated on exact and approximate methods to solve VRPs, as well as estimations of the solution value prior to solving. The total distance travelled is roughly equivalent to, and thus a good estimate of, the variable VRP costs. Furthermore, we showed the potential of machine learning to support order selection under routing considerations. Lastly, we introduced the Monte Carlo simulation. This method could provide insight into the variety and robustness of the most cost-efficient order selection because the order selection with minimal costs per order given the total number of forecasted potential orders is dependent on uncertain order characteristics (e.g., order division).

In summary, we contribute to the existing literature with a methodology that selects the most cost-efficient order subset to determine delivery capacity at the tactical level. This contribution includes:

1. An application of distance estimation, prior to solving, to support order selection decisions to determine the capacity to deploy.
2. An application of machine learning model(s) to estimate the variable costs of a VRP.
3. The provision of insight into the dependencies of several features concerning the VRP distance.
4. A general method based on information known at the tactical level instead of estimated or assumed information such as exact order locations.
5. A method that takes variability and stochasticity into account and assesses the robustness of the optimal order selection given stochastic input sources.

4 Solution design

This chapter proposes a solution design that supports capacity deployment decisions at the tactical level in routing problems. We illustrate our solution methodology on the hub in Hilversum. Section 4.1 describes the problem context and Section 4.2 proposes the solution methodology to address the problem. Next, Section 4.3 describes the development of machine learning models to support order selection decisions. Section 4.4 elaborates on the Monte Carlo simulation that evaluates the robustness of the order selection with minimal costs per order given the total number of forecasted potential orders. Finally, Section 4.5 concludes the chapter.

4.1 Problem description

We aim to find the required capacity of CBB's hubs that gives minimal costs per order based on the forecasted potential orders of the hub's delivery area. We decide to determine the required capacity based on the daily number of orders to deliver as required by CBB, to ensure that our solution is easily implementable in the current system (see Section 2.1). Hence, the problem consists of finding the hub's most cost-efficient order selection per day based on the total number of forecasted potential orders. This implicates making two decisions: choosing how many orders to select and how to estimate the costs of an order selection.

Order selection

Order information such as location and volume is unknown at this stage, i.e., we consider a tactical level problem where we encounter uncertainties. To be more precise, we determine the required capacity for 4 to 16 weeks ahead (see Section 2.1), whilst CBB strives to deliver the placed order the next working day. Hence, there is uncertainty in all types of order information from the to-be-evaluated days.

The forecasted potential orders are the orders of the entire delivery area of a CBB hub that CBB predicts to be eligible for CBB delivery. The delivery area consists of a set of subareas indicated by A, partitioned by the numeric values of the postal codes (PC4). Every delivery area has one hub from which each $v \in V$ delivery bikes and each $m \in M$ delivery men depart, where the number of v and m that depart is equal, to deliver each $n \in N$ orders. We consider an infinite capacity of the number of delivery bikes and delivery men hours, such that we can determine the most cost-efficient number to deploy independent of the current available capacity. Each bike tour is constrained by the bike capacity K and the maximal tour time T.

Costs per order

The costs consist of fixed costs independent of the order selection and variable costs that are dependent on the order selection as discussed in Section 2.2. The total fixed costs are always the same, and the fixed costs per order decrease when the number of orders increases. Hence, these costs converge to zero when the numbers of orders converge to infinity.

The total variable costs consist of a cost per hour, per kilometre and per order. They increase when the number of orders increases (more travel time, more travel distance and more orders). However, the relation between the variable costs per order and the number of orders is still unknown. These costs might increase or decrease when the number of orders increases, depending on the total cost increase of the travel time and the travel distance relative to the number of orders. We obtain a decrease in costs per order when the decrease in the fixed costs per order is greater than the increase in the variable costs per order, or when both the fixed and the variable costs per order decrease. Therefore, we obtain a (local) costs per order minimum when after a decrease, an increase is observed.

4.2 Solution methodology

We develop a solution methodology that supports capacity deployment decisions at the tactical level in routing problems. We choose to analyze the operational routing plan, using Monte Carlo simulation, to make decisions about the most cost-efficient capacity to deploy. Here, we mimic a delivery day, choose which orders to deliver this day based on the lowest estimated costs per order, and determine how much capacity is necessary to deliver these orders. We propose to develop a machine learning model that supports the order selection decision by estimating the costs per order of the selected orders. Accordingly, the solution approach consists of two main steps:

1) Develop machine learning model(s) that estimates the costs of an order selection.
2) Build a Monte Carlo simulation, in which the machine learning model(s) are utilized to support order selection decisions.

Figure 19 shows the full, high-over solution approach and the general steps that should be taken. It displays the relation of the machine learning models with the Monte Carlo simulation. We decide to develop two types of machine learning models: trained on generated data (1) and trained on historical data (2), on which we further elaborate in Section 4.3.3. These developed machine learning models are input for the Monte Carlo simulation.

The goals of the Monte Carlo simulation are 1) to understand the impact of uncertainty in the order distribution over the delivery area on the most cost-efficient order selection, and 2) to validate and compare the most cost-efficient order selection chosen by the machine learning models with the orders selected by the current system and a literature approach to benchmark from. The simulation consists of four phases. In Phase 1, we distribute the forecasted potential orders over the delivery area. Next, we select the most cost-efficient order subset with the machine learning models in Phase 2. Additionally, in Phase 2, we select orders based on a literature approach, and we select all orders from the PC4s located in the 'optimal levels' as done in the current system (see Section 2.1). Note that the latter two methods do not need to be developed because we can simply retrieve them from the literature and CBB. In Phase 3, we solve the VRP of the order selections for comparison and validation purposes. In Phase 4,

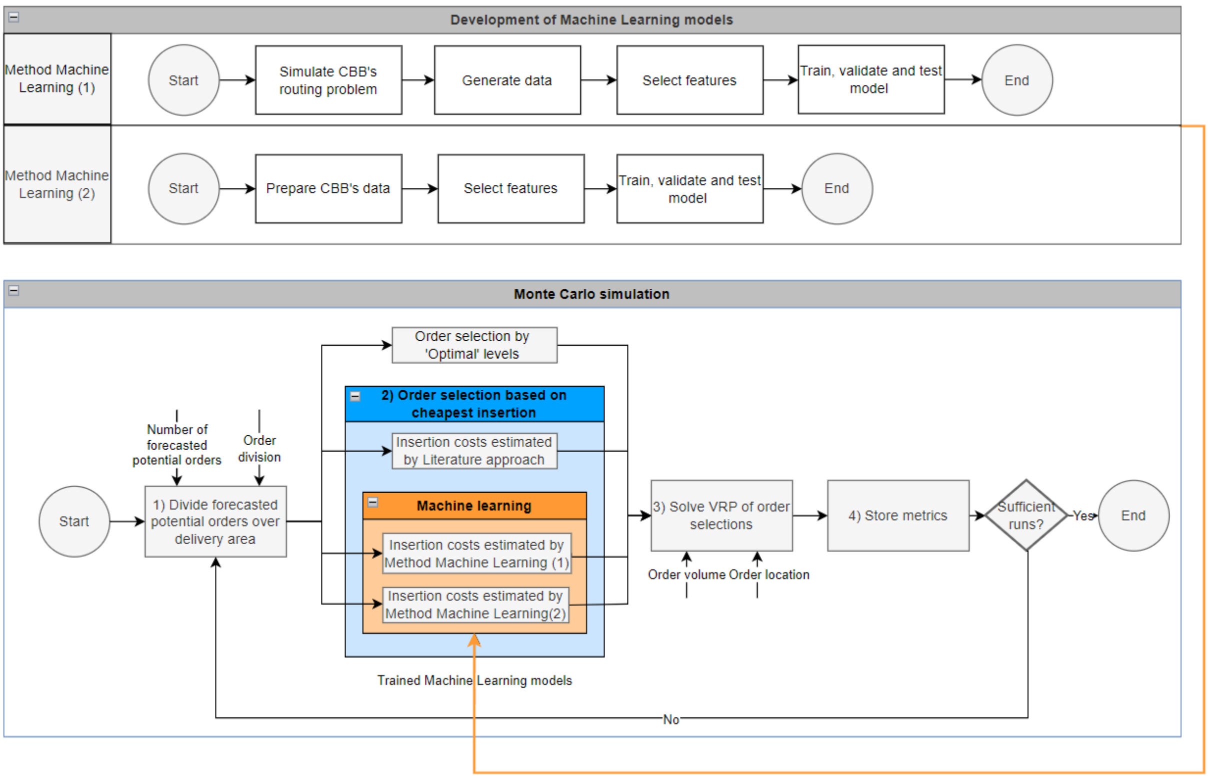

Figure 19 Solution methodology

we store the performance metrics to be able to evaluate the robustness of the most cost-efficient order selection and to learn what method can best be utilized to support CBB with capacity deployment decisions. We repeat these four phases until both the most cost-efficient number of orders and the costs per order of this order selection are statistically significant. Multiple simulation runs account for the uncertainty observed in for example the division of orders over the delivery area. Hence, if we run the simulation sufficient times, we retrieve an average value of the order selection and costs per order with a specified confidence level (see Section 5.2.1.2). Note that Phases 3 and 4 are for validation and comparison purposes because the most cost-efficient order selections are the output of Phase 2.

The motivation for this solution methodology is nine-folded:

Machine learning

1) The use of machine learning estimations prevents solving the VRP for all possible combinations to find the subset with the lowest costs per order. This is very computationally demanding compared to estimating with a machine learning model, and might even be considered an infeasible option.
2) When we exploit a machine learning model, we do not necessarily need all detailed order information to determine the distance contrary to solving a VRP. Hence, we can utilize the available tactical-level information to describe the VRP.
3) According to the literature, there is a high potential in exploiting machine learning models to predict a VRP distance and/or costs.

Monte Carlo simulation

4) A Monte Carlo simulation provides the possibility to mimic the route planning. Therefore, we can establish an initial routing plan, on which we can determine the costs of an order selection as suggested in the literature. These costs are necessary to determine what number of orders selected is most cost-efficient.
5) The simulation of the route planning also allows the alignment of the order selection decision with the bike route efficiency as required (see Section 1.2), because we choose the orders in the initial plan with the lowest costs per order, i.e., with the least distance per order and thus the highest route efficiency.
6) When we simulate the route planning, we can desirably base the order selected in the initial plan on the total number of forecasted potential orders (see Section 1.2). This number can be the input of the simulation and indicates how many orders can be maximally selected.
7) The output of the Monte Carlo simulation is the number of orders to deliver, which is directly implementable in the current capacity planning and desired by CBB.
8) The Monte Carlo simulation allows evaluation of the most cost-efficient order selection based on unknown and/or stochastic inputs that are common at the tactical level (e.g., order distribution over the delivery area). Hence, we can assess the robustness of the most cost-efficient order selection.
9) A Monte Carlo simulation is suitable for our routing problem because the passage of time does not play a significant role. The evaluated delivery day is independent of any other delivery day because the orders and capacity of the evaluated day cannot be shifted.

According to the two main steps of the solution methodology, Section 4.3 elaborates on the consecutive steps to develop the machine learning models, and Section 4.4 motivates the approach and choices of each phase of the Monte Carlo simulation.

4.3 Machine learning to support order selection

The goal of the order selection is to select the most cost-efficient subset of orders. To determine the costs of an order selection, we should know (an estimation of) the travel time in hours, the travel distance in kilometres and the number of orders selected as explained in

Section 2.2 and Section 4.1. Hence, we propose to develop a machine learning model that tries to predict the VRP distance based on features.

Figure 20 depicts the general process of developing a machine learning model. This section explains this process, and thus the steps we are going to take to develop the machine learning models to support order selection. Accordingly, Section 4.3.1 elaborates on the choice of the features. Next, Section 4.3.2 motivates the choice of two machine learning models that we are going to develop and their respective hyperparameters. Afterwards, Section 4.3.3 explains how we obtain two types of data sets. Hence, the combination of two machine learning models and two types of data sets results in four machine learning models to evaluate.

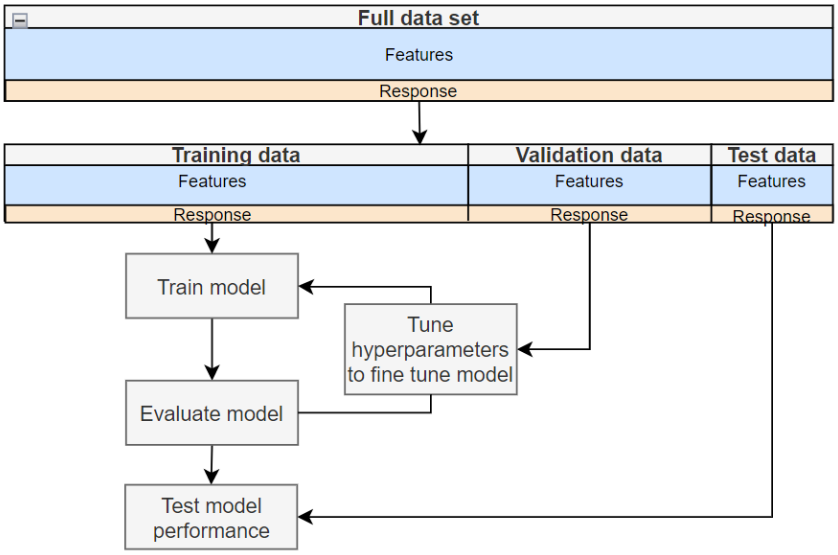

Figure 20 General process of developing a machine learning model

4.3.1 Features

Features are independent variables that describe the dependent variable in a machine learning model. Hence, we use features to estimate the distance of the VRP, i.e., the response. We decide to only consider features of which the values are known when selecting the orders to determine capacity at the tactical level. This may give a general tool applicable at the tactical level for capacity decision-making without assuming all order information. The known features are the total number of forecasted potential orders, the total number of selected orders and the spatial features of the delivery area.

Including only these known features, means excluding many features considered in the literature, such as inter-customer driving time and characteristics of the customer locations. However, these literature features were often considered for operational models, i.e., where these feature values were known or could be determined. Also, we do not assume like Daganzo (1987), Robusté et al. (2004) and Figliozzi (2008) that the number of necessary routes or vehicles is known a priori, since we want this information as the output of the model. The inclusion of unknown features would imply assuming their values. In that case, the model would be trained on these values, and therefore more prone to overfitting and to create bias. Also, there are way more operational than tactical level features to utilize, which could make the model unnecessarily complex. Therefore, we decide to exclude unknown features from our model.

In the end, we are interested in predicting the costs of delivering an extra order independent of order location, to determine how much required capacity, i.e., delivery men hours and bike routes, to deploy. We do not have to determine the sequence in which we are going to serve the orders, and we can only speculate about their location and volume. Therefore, we group orders together into zones like Hernandez, Gendreau & Potvin (2017) to obtain spatial features that might describe the solution value. Based on historical data and the experience of Coolblue, we use the numeric values of the postal codes (PC4) to partition the delivery area.

Table 3 shows the features used in the machine learning model. We have relatively few features (6) compared to the data (8798 generated data points, and 10160 (small hubs) / 5579 (large hubs) historical data points), on which we further elaborate in Section 4.3.3). Therefore, the chance of overfitting and redundant data, and the computational time to train the model are relatively low. Note that Feature 6 can be interpreted as a term related to the distance between customers (Figliozzi, 2008). Furthermore, we decide to exclude the number of selected orders per PC4 as a feature based on the predictive performance. The inclusion of e.g., 45 features, one for every PC4 in Hilversum, introduces extra noise and decreases the accuracy of the model. The model becomes more complex with 51 instead of 6 features, and tends to overfit the data. We elaborate on this noise in Appendix B. Also, we decide to exclude the feature $\sqrt{\text{Total area} \,/\, \text{Selected orders}}$ introduced by Figliozzi (2008), since it is directly correlated with Feature 6.

Table 3 Features of the machine learning model(s)

Number	Feature	Description
1	Potential orders	The number of potential orders in the entire delivery area
2	Selected orders	The number of selected orders in the entire delivery area
3	Total area	The total area in km^2 of the PC4s from the selected orders
4	Average distance to the hub	The average distance in km from the centroid of the PC4s from the selected orders to the hub
5	Variance distance to the hub	The variance in km from the centroid of the PC4s from the selected orders to the hub
6	$\sqrt{\text{Total area} * \text{Selected orders}}$	The square root of the total area in km^2 of the PC4s from the selected orders multiplied by the number of selected orders in the entire delivery area

4.3.2 Machine learning models

We use the Python Lazy Predict library (Pandala, 2022) to understand which machine learning models fit our dataset. We refer to Section 4.3.3 for more information on the dataset. We want to predict the quantitative distance of the VRP (dependent variable) with features (independent variables), and therefore perform a regression analysis (James, Witten, Hastie, & TIbshirani, 2021). We found that the simplest regression model, linear regression, is not a valid method for the dataset. Specifically, we observe that the residuals do not follow a normal distribution and are heteroscedastic, meaning that the underlying assumptions to utilize a linear regression model are not met.

We find that decision tree-based models, which are capable of handling non-linear relations, are most suitable for our dataset, i.e., have the highest predictive performance. Therefore, we consider the highest scoring models random forest regression (RFR) and lightweight gradient boosting machines (LGBM) for the rest of this study. Other high-scoring models are histogram gradient boosting regression, gradient boosting regression and extra trees regression. The first two work quite similarly to LGBM, i.e., gradient-boosting decision trees, and the latter is a variation of random forest regression. Hence, we expect no significant difference between these models, and we decide to only evaluate RFR and LGBM. We choose to train and test the models in Python with the Scikit-Learn (Scikit-learn, 2023) and the LightGBM (Microsoft Corporation, 2023) libraries respectively. For both models, we use the bootstrapping method as a resampling technique with out-of-bag samples to more accurately fit the data. This means that for each decision tree, we randomly draw data samples with replacement from the training data, on which we then build the decision tree. We test the performance on the data that is never sampled, i.e., the out-of-bag sample.

4.3.2.1 Hyperparameter tuning

Hyperparameters in machine learning models are parameters that are selected before training the algorithm. The machine learning model utilizes these hyperparameters to effectively learn the optimal parameters that accurately predict the dependent variable from the features (independent variables). Hence, they directly control the behaviour of the training algorithm and have a significant effect on the predictive performance of the machine learning models (Wu, et al., 2019).

For RFR, we decide to optimize the maximal number of features for splitting. This hyperparameter influences the generalization error and is often seen as the most important hyperparameter of RFR (Wright & Ziegler, 2017). Furthermore, we choose to tune the stopping criterium: the minimum number of data points per leaf node. To balance model performance with run time, we set the number of trees to 200 for both models as done in comparable research (Akkerman & Mes, 2022). We decide to let the trees grow to full depth because we only have a limited set of features.

For LGBM, we decide to optimize the maximal number of bins to bucket the features for LGBM, which controls the number of features considered per split (see Section 3.3.2.2). Hence, this hyperparameter is important as discussed with RFR. Also, we tune the learning rate, as this controls the step size at each iteration at which the algorithm converges to the final model. The maximal number of leaves controls the complexity of the model and should be tuned together with the maximal tree depth, often is suggested that the maximal number of leaves is equal to $2^{maximal\ tree\ depth}$ (Microsoft Corporation, 2023). Additionally, an iterative learning process can be more prone to overfitting than averaging randomly constructed models such as RFR does, hence we decide to tune both the maximal number of leaves and the maximal tree depth. Again, we consider 200 trees in the forest.

We tune the hyperparameters on the (unseen) validation set to find a set of optimal values. We use Bayesian optimization with 5-fold cross-validation with the Scikit-Learn and Scikit-Optimize Python libraries (Head, Kumar, Nahrstaedt, Louppe, & Shcherbatyi, 2021) and select the hyperparameters belonging to the models with the best R^2. For more information on this optimization and its implementation, we refer to Akkerman & Mes (2022). Table 4 summarizes the hyperparameters and the value decisions.

Table 4 The to-be-tuned hyperparameters of the RFR and LGBM model

Model	Hyperparameter	Description	Value
RFR	Maximal number features for splitting	The number of features to consider when looking for the best split.	*To be tuned*
	Minimum number of data points per leaf node	The minimum number of data required to be at a leaf node.	*To be tuned*
	Number of trees	The number of trees in the forest.	200
	Maximal tree depth	The maximum depth of the tree.	Full depth
LGBM	Maximal number of bins	The maximum number of bins to bucket the feature values.	*To be tuned*
	Learning rate	The rate at which the model is updated.	*To be tuned*
	Maximal tree depth	The maximum depth of the tree.	*To be tuned*
	Number of leaves	Maximal number of leaves in one tree.	*To be tuned*
	Number of trees	The number of trees in the forest.	200

4.3.3 Training data

To develop a regression model, we need to obtain training data. We decide to distinguish between two types of training data; historical data and generated data. We use both types of data to make separate machine learning models, to learn what data can best be utilized to support CBB with capacity deployment decisions. We split the obtained training data into training, test and validation sets, with respectively 80%, 10%, and 10% ratio. We use the training set to fit the parameters to the machine learning model, the validation set to tune the hyperparameters of the model and the test set to evaluate the fitted model on the training set. Hence, the use of the validation set ensures that the hyperparameters are trained on unseen data, which prevents overfitting and a biased evaluation, and increases the generalizing capability.

4.3.3.1 Historical data

CBB collects data of the delivery process since the start of each hub. This data consists of mainly two tables. The first table stores all potential orders, i.e., the primary key is the order identifier (ID). It contains information such as the order- and delivery date and time, the country, the allocated hub, the PC4 in which the order is located, and binary values if the order suffices the constraints for CBB delivery (see Section 2.3). This table should be used to determine the total daily number of potential orders eligible for CBB delivery. The second table stores all tour information, i.e., the primary key is the tour ID. It contains tour information like the hub from which the tour departs, the delivery date and the total tour distance. Also, it includes the stops of the tour, which is described by most importantly the sequence number of the stop in the tour, the planned and realized arrival time at the stop, the planned and realized time to the next stop, and the address of the stop.

Unfortunately, the hubs only have been open for roughly two years. Because one day equals one data point, resulting in +/- 730 data points before data preparation, we have insufficient data to develop an RFR and LGBM model per hub. Therefore, we decide to group hubs together based on hub size in consultation with CBB and make individual hub predictions with the aggregated model. We discriminate between small and large hubs, as we find that the spatial features and bike routes are most comparable with this distinction. We prepare the data in Coolblue's data warehouse BiqQuery (SQL) by deleting observations with missing values, test or fun tours that are not representative (e.g., tests with longer maximal bike time or present delivery at Sinterklaas to employees' children), observations with old volume restrictions (causing, e.g., different inflow and route utilization), and by adjusting the data when the hub has moved (e.g., the distance to the hub should be to the old hub instead of the new hub). Afterwards, we obtain a sufficiently large data set of 10,160 and 5,579 entries respectively.

The main advantage of grouping the data is the generalization and implementation of the solution methodology. Hence, only 2 machine learning models (one for small hubs and one for large hubs) have to be developed to support capacity deployment decisions at all hubs. This is a way less exhaustive process than data generation for every hub as explained in Section 4.3.3.2. The disadvantage is that in a shared model, hub-specific spatial features emerge less, such as the location of, e.g., a forest in the delivery area and customer distributions. However, the purpose of the model is to support capacity deployment decisions. Therefore, deviation of the distance estimation model does not necessarily influence the capacity decision, if the model always proportionally under- or overestimates the actual distance.

Furthermore, historical data can be biased. For example, it might not be representative, i.e., over- or underrepresented observations, or consist of manual adjustments based on human logic that are not readily apparent during data analysis. Therefore, the data should contain sufficient variability to be able to make proper predictions. By comparing the performance of the models with historical and generated data, we can conclude about the practicality of the model with historical data.

4.3.3.2 Generated data

To ensure that the model can decide upon the optimal capacity and does not only mimic the historical decision, we also generate data. This filters out the bias and logic of for example the human planners of Coolblue. Hence, we obtain an additional data set, on which we train, test and validate the RFR and LGBM models.

We simulate the order selection of the hub in Hilversum to generate instances. We build and run this simulation for data generation in Python 3.9. Figure 21 displays the 4 phases of the data generation based on the historical potential orders per PC4 of Hilversum suitable for bike delivery. In line with those phases, Section 4.3.3.2.1 elaborates on the order division, Section 4.3.3.2.2 on the order selection, Section 4.3.3.2.3 on solving the VRP and Section 4.3.3.2.4 on storing the features. Furthermore, Section 4.3.3.2.5 explains how we ensure sufficient order selection variability.

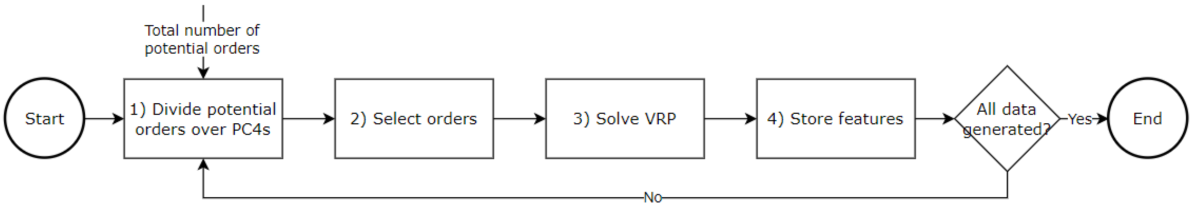

Figure 21 Flowchart data generation simulation

4.3.3.2.1 Order division

Historically, we know the daily number of potential orders per PC4. Therefore, we use almost 5 years of available, historical PC4 order divisions to create instances (1759 days: 24-01-2018 – 12-12-2022). Hence, we simulate the exact same 1759 days with the exact same number of orders per PC4. This way, we automatically filter out all orders that were unsafe to deliver or did not suffice the size restrictions (see Section 2.3). Furthermore, we ensure representative order divisions by using historically observed order divisions. We prepare this datafile in Coolblue’s data warehouse BiqQuery.

Because we cannot match the historical potential orders with the actual addresses, we decide to randomly draw a customer location from a list with an address in the corresponding PC4 inspired by Cleophas & Ehmke (2014). Therefore, we first create a list with 50 addresses per PC4. Each address contains the street, house number, postal code (PC4 and PC6), latitude and longitude. 50 Addresses should suffice because we observed a maximum of 30 potential orders per PC4 per day and a minimal of 96 distinct order addresses per PC4. First, we prepare the historical order data stored in BiqQuery (e.g., remove typing errors of the customers). Afterwards, we randomly filter one random address per full postal code (PC6), i.e., the four numbers and two letters of the postal code. This way the addresses are spread throughout the entire delivery area. We randomly filter 50 addresses of the address list containing unique PC6s. For the 5 PC4s with less than 50 PC6s, we randomly duplicate PC6s in case we select a higher number of orders in a PC4 then the number of PC6s that belong to that PC4.

4.3.3.2.2 Order selection

The generated data set should contain all kinds of order selection decisions to train models that make proper predictions. Hence, we should select a different number of orders per day to ensure enough order selection variability. Therefore, we decide to categorize the selected orders relative to the potential orders every day into 4 categories, i.e., 2.6 - 25%, 25 - 50%, 50 - 75%, and 75 - 100%. Note that the first category starts at 2.6% instead of 0.0% to ensure we never choose only one order and always can solve a VRP. This 2.6% results from the minimum number of 40 potential orders observed, i.e., $\frac{1\ order}{40\ orders} = 2.5\%$, thus with 2.6% the minimum number of orders selected is 2. Every simulated day, we randomly draw a percentage in the range from every category, which results in four percentages. We translate these percentages to a number of orders based on the total number of potential orders, e.g., 40% of 140 potential orders is 56 orders to be selected.

We start with an empty solution and select orders based on the nearest neighbour heuristic. This relatively simple heuristic, i.e., select the next closest customer (here order), is a well-known heuristic in VRP literature. We choose this heuristic because it assumes making an order selection with approximately minimal costs since the (variable) costs in a VRP are assumed to be roughly equal to the distance (Nicola, Vetschera, & Dragomir, 2018) (Daganzo, 1987) (Figliozzi, 2008). To determine which order is the closest, we first have to determine the distance between orders. We use the latitudes and longitudes stored in the historical address list of the simulated day for this and convert this list to a distance matrix with the haversine formula in Python (Rouberol, 2022). This formula returns the distance between two points on a sphere, like the earth, and it is therefore representative. Unfortunately, we do not have access to the real distance as determined by CBB's planning algorithm. Therefore, we base the nearest neighbour decision on the haversine distance instead of the actual distance. We stop selecting orders when we reach the number of orders based on the categories. Therefore, we obtain 4 order selections per simulated day.

4.3.3.2.3 Solve VRP

To obtain the total daily distance travelled to deliver the selected orders, we solve the VRP for the 4 daily order selections. To solve this problem with limited bike capacity, we need the volume of the orders. Because we can also not match the historical potential orders with the historical order volume, we decide to estimate the order volume based on historical data. The orders eligible for CBB delivery suffice a maximal volume constraint. Hence, to include this constraint and because we did not find a significant fit on a theoretical distribution, we use the empirical probability distribution function of the order volume based on historically observed orders in Hilversum.

We use the current state-of-the-art Python algorithm of Vidal (2022) to solve the VRP with the maximal trip duration and bike capacity of each order selection. This is an open-source implementation of the hybrid genetic search specialized to the CVRP.

4.3.3.2.4 Store the respective features

Based on the solution to the VRP, we calculate and store the features belonging to the 4 order selections. Hence, we obtain 1759 days * 4 order selections = 7036 generated instances as input for the training, validation and testing of the machine learning models. Because this number is more than sufficient to develop RFR and LGBM, it is acceptable to let the number of generated instances depend on the amount of historical data.

4.3.3.2.5 Order selection variability

After the data is generated, we observe that indeed the order selection percentages are roughly equally spread. However, we also find that we only observe a limited number of observations with more than 150 orders selected. This is because the average observed number of potential orders is 140, and therefore, on average, less than 140 orders are selected in the simulation. Figure 22 displays both observations.

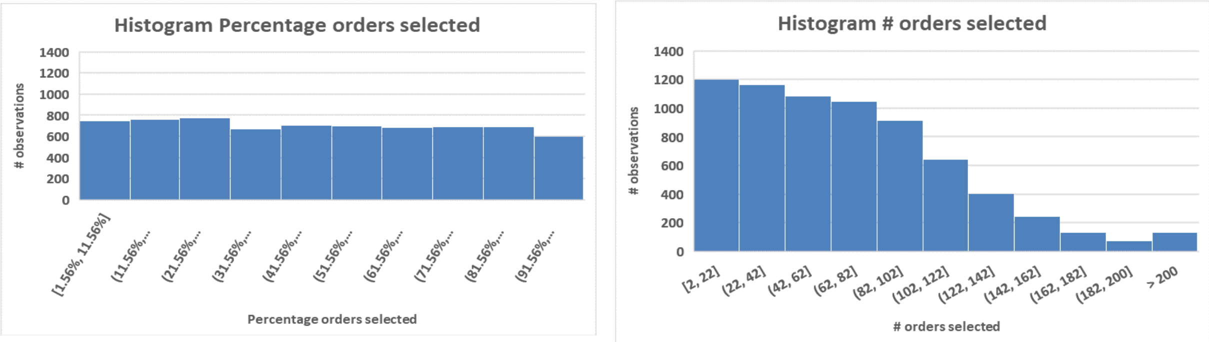

Figure 22 Histogram of the percentage orders selected (left) and number of orders selected (right)

Because we also want that our model makes a proper prediction when more than 150 or even 200 orders are selected, we decide to generate some extra instances. Hence, now we filter the days with more than 150 and 200 potential orders from the 1759 days. We find 581 observations with more than 150 potential orders and 300 observations with more than 200 potential orders. Again, we execute the 4 abovementioned steps. However, we now define two instead of four categories of the selected orders relative to the potential orders every day, i.e., 70 - 85% & 85 - 100%. Hence, we ensure that the number of orders selected is high and that the percentage of orders selected is not completely unbalanced. This results in (581 + 300) * 2 = 1762 extra instances, and 7036 + 1762 = 8798 final instances as input for the training, validation and testing of the machine learning models. Figure 23 shows the histogram of both the percentage of orders selected relative to the total number of potential orders and the histogram of the number of orders selected. We observe more observations with a relatively high percentage, and with more than 150 and 200 orders. Thus, we created a balance between the percentage and the absolute number of orders selected to ensure sufficient order selection variability in the data set.

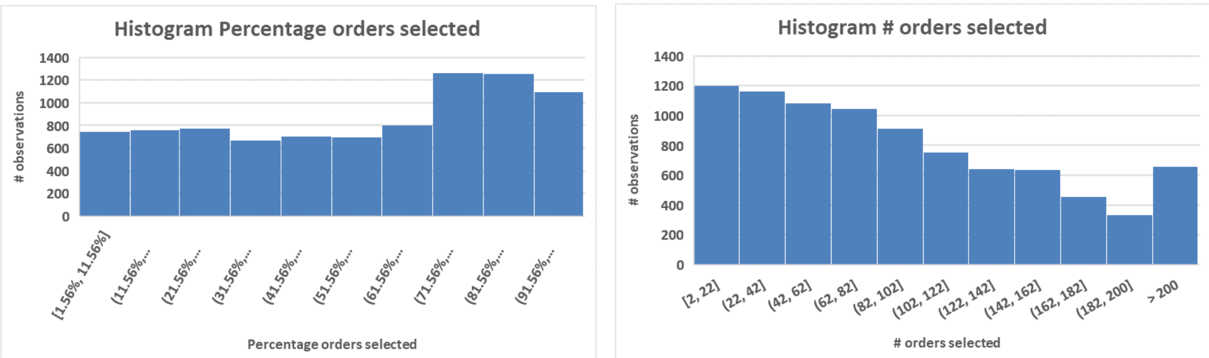

Figure 23 Histogram of the percentage orders selected (left) and number of orders selected (right) with extra generated instances

4.4 Monte Carlo simulation

This section describes the Monte Carlo simulation. We use this simulation model to advise Coolblue on the robustness of the order selection with minimal costs per order given the total number of forecasted potential orders. The order selection might vary because there is uncertainty about how the orders are distributed over the PC4s. Hence, the simulation of multiple scenarios from the PC4 order division stochasticity gives an estimate of the overall performance of the most cost-efficient order selection in realistic settings (Juan, Faulin, Grasman, Rabe, & Figureira, 2015). Furthermore, an additional purpose of this simulation model is to compare the proposed methods' estimated costs per order of an order selection with each other and with the current situation. Hence, we stochastically evaluate the solutions proposed by the different methods. Section 4.4.1, Section 4.4.2 and Section 4.4.3 elaborate on the first three phases of the model, respectively, as shown in Figure 19. Finally, Section 4.5 concludes the chapter.

4.4.1 Order division

One simulation run is equivalent to one day because we want to find the daily required capacity as explained in Section 2.3 and Section 4.1. The input of the simulation run is the daily number of forecasted potential orders. However, the order locations and even the order division over the PC4s are unknown and should be estimated to make a VRP distance prediction. In line with the features (see Section 4.3.1), we decide to estimate the number of potential orders per PC4 because this allows describing the order selection based on spatial PC4 characteristics. Additionally, it is the deepest level of detail CBB works with. The division of orders over the PC4s influences the travel time and distance of an order selection, and thus the most cost-efficient subset of orders.

To determine how the orders are divided over de PC4s, we estimate the distributions based on historically observed orders per PC4. We choose among the possible discrete probability distributions Poisson, binomial, geometric and negative binomial, and compare the goodness-of-fit values to determine which one represents reality (the best). Similarly, as with the data generation, we use almost 5 years of available, historical PC4 order divisions (24-01-2018 – 12-12-2022). We decide to plot the histogram of the N observed data points per PC4 with B bins of the size of 1 order because we want to know the probability belonging to observing a(n) (multitude of) order(s). We estimate the sample parameters of the theoretical distributions per PC4 from the observed data. Next, we make B predictions per PC4 with the density distribution functions *dpois, dbinom, dgeom, dnbinom* in RStudio fitted on their estimated parameters. Finally, we compare the predictions with the observations. We compute the p-value of the Chi-square goodness-of-fit test with $B - 1$ degrees of freedom with *chisq.test* in RStudio, and compare this with a significance level of 0.05.

Once fitted, we randomly draw a number of orders per PC4 based on the discrete probability distributions. To ensure the sum of all drawn orders is equal to the total number of forecasted potential orders, we normalize the drawn orders with the normalizing factor, i.e., the total number of forecasted potential orders divided by the sum of all drawn orders. Hence, we ensure that the orders are randomly, but proportionally spread over the PC4s as observed in Section 2.1.2. Because the orders are most likely no discrete values after normalization, we round them to the nearest integer and randomly add or subtract one order from a PC4 until the sum of the orders is equal to the total forecasted potential orders. Figure 24 depicts this order division process. We run multiple Monte Carlo simulations to account for the stochasticity of the PC4 order division.

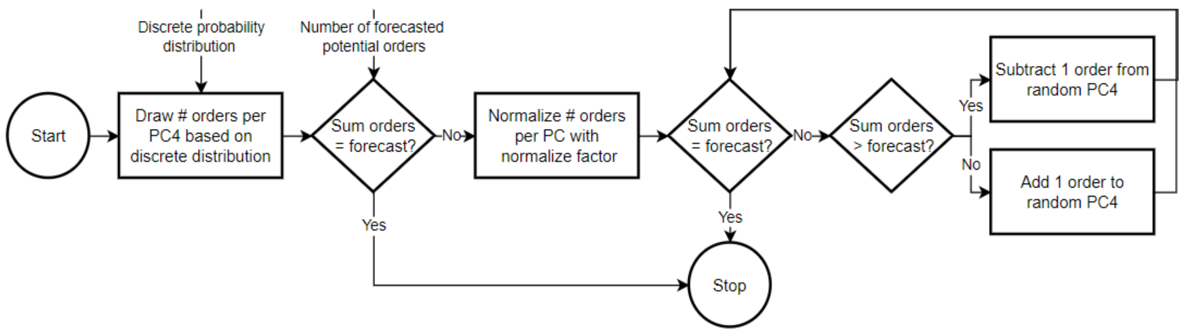

Figure 24 Order division flowchart

4.4.2 Order selection

The second phase is selecting orders from the order division. The goal is to select the most cost-efficient subset of orders, i.e., with the lowest costs per order. Instead of solving the VRP to find the related costs per order of an order subset, whilst assuming necessary information such as order locations, we predict the solution value based on known features as explained in Section 4.3.

Order selection heuristic

We use the cheapest insertion heuristic on PC4 level to decide upon the order selection. We choose this heuristic because 1) it assumes inserting the order with the minimal increase in costs, which is in line with finding the order selection with minimal costs per order, and 2) it is not extremely computationally demanding, which is beneficial when the number of potential orders is large and when order selections have to be made repeatedly during several simulation runs. This heuristic starts with an empty solution and adds the order to the current order selection that minimizes the increase in distance. Since the (variable) costs in a VRP are assumed to be roughly equal to the distance (Nicola, Vetschera, & Dragomir, 2018) (Daganzo, 1987) (Figliozzi, 2008), the heuristic, therefore, assumes to make an order selection with minimal costs increase. Because we only know the orders on PC4 level, see Section 4.1, we evaluate the distance increase on PC4 level. Hence, no additional order information, i.e., order location and volume, is necessary to perform this heuristic. Once a new order is added, we determine the costs per order of the current order selection based on the distance estimation.

Distance estimation methods

We decide to compare multiple distance estimation methods to approximate the distance increase when adding an order to the selection: machine learning model(s) trained on generated data (see Section 4.3.3.2), machine learning model(s) trained on historical data (see Section 4.3.3.1), and the Daganzo estimation (1987) shown in Equation 3-8 to benchmark from. We choose the Daganzo estimation because it is most in line with the features known at the tactical level and it does not include the number of vehicles to deploy, which we desire as output instead of input of the model. Hence, the Daganzo estimation requires the least additional assumptions or uncertainty, compared to e.g., Figliozzi (2008), enabling a more reliable comparison between the distance estimation methods. The only assumption we make is the estimation of the maximal number of *Q* customers a vehicle can serve, which we estimate by dividing the total vehicle capacity by the average order volume of the evaluated day. Furthermore, we define *N* as the number of orders selected, *A* as the total area in km^2 of the PC4s from the selected orders, and *r* as the average distance from the centroid of the PC4s from the selected orders to the hub in the Daganzo estimation.

Order selection algorithm

Figure 25 displays the pseudocode of the order selection algorithm. We use this algorithm for all distance estimation models and explain it based on the lines of Figure 25. The order selection algorithm starts with an empty solution (Line 1) and determines the costs per order of the order selection when adding a new order until there are no forecasted potential orders left (Line 2). Hence, in every iteration we temporarily add an order of a PC4 (Line 7) that still contains potential orders to the current order selection (Line 6), calculate the features belonging to this order selection (Line 8) and estimate the distance increase (Line 9). We store the distance and the order of the PC4, if the distance increase is smaller than the lowest distance increase (Lines 10 – 13). Then, we remove the temporarily selected order from the PC4 (Line 14) and repeat the same steps until all PC4s are evaluated (Line 5). Once all PC4s are evaluated, the order of the PC4 with the shortest distance increase is added to the current order selection (line 18), and removed from the current potential orders (Line 17).

The subset with the lowest costs per order, determined by the distance estimation and the total number of selected orders, is the final order selection (Lines 23 – 24). We compare all subset sizes, i.e., we do not stop adding orders when the costs per order increase, to prevent stopping at a local minimum for the costs per order.

CHEAPEST INSERTION

```
Initialize Features, Selectedorders = 0, minCPOindex = -1, Orderselection = [], CurrentPotentialorders
= list with potential orders per PC4
while (∑_{i=1}^{PC4s} CurrentPotentialorders [i] > 0) do
    Set shortestDistance = ∞, shortestDistancePC4 = -1
    Selectedorders += 1
    for (all PC4's in PC4list) do
        if (CurrentPotentialorders[PC4] > 0) then
            Orderselection[PC4] += 1
            calculate features of temporarily adding 1 order of PC4 to Orderselection
            estimate the respective CVRP distance
            if (distance < shortestDistance) then
                shortestDistance = distance, shortestDistancePC4 = PC4
                store Features
            end if
            Orderselection[PC4] -= 1
        end if
    end for
    CurrentPotentialorders[shortestDistancePC4] -= 1
    Orderselection[shortestDistancePC4] += 1
    calculate costs per order of Orderselection
    store costs per order of Orderselection
    update Features of Orderselection
end while
minCPOindex = min(costs per order).index
select Orderselection[1:minCPOindex,]
return Orderselection
```

Figure 25 Pseudocode order selection heuristic

4.4.3 Solve VRP

To validate and compare our solution method(s), we solve the VRP of the final order selections and determine the respective costs per order. To solve the VRP, we need some additional order information. Like the order division over the PC4s, this additional information is uncertain and influences the distance of the VRP.

First, we need order locations. Again, we use the list of 50 addresses per PC4 including their latitude and longitude based on historical data (see Section 4.3.3.2), draw for every order in the final order selection a random address from this list with the corresponding PC4, and convert this list to a distance matrix with the haversine formula in Python (Rouberol, 2022). Second, we need the order volume. Again, we use the empirical distribution function of the order volume based on historically observed orders in Hilversum, because it takes the maximal volume constraint into account (see Section 4.3.3.2). We run multiple Monte Carlo simulations to account for the randomness of these inputs, and during every run, we use the same location and volume values for each method.

Finally, we use the current state-of-the-art Python algorithm of Vidal (2022) to solve the VRP with the maximal trip duration of each order selection as introduced in Section 4.3.3.2. Although this is not equivalent to the routing planning mechanism CBB utilizes, it gives a good indication of the potential of the model(s) and enables a valid comparison between the order selection methods.

4.5 Conclusions

This chapter introduced the problem context that consists of finding the hub's most cost-efficient order selection per day based on the total number of forecasted potential orders. Also, we proposed the solution methodology to address this tactical routing problem. We decided to simulate the operational routing plan to make decisions about the most cost-efficient order selection with a Monte Carlo simulation. We proposed to utilize machine learning models to support order selection decisions. Accordingly, we introduced the tactical level features, the suitable machine learning models (random forest regression and lightweight gradient boosting machines), the hyperparameters of both models and how we obtain training data to develop the models. We decided to use two types of training data (generated and historical) to learn what data can best be utilized to support CBB with the order selection decision.

Furthermore, we explained the four steps of the simulation. First, we distribute the orders over the PC4s of the delivery based on a discrete probability distribution. Second, we decide upon the most cost-efficient order selection with the cheapest insertion heuristic, where we estimate the insertion costs with the machine learning models trained on generated data and trained on historical data. Additionally, we estimate the insertion costs with the heuristic of Daganzo (1987) to benchmark from, and we select orders with CBB's current order selection method for comparison purposes. Third, we solve the VRP of the final order selections for validation purposes and fourth, we store the metrics.

5 Performance

This chapter implements the solution methodology and assesses its performance. We illustrate the solution methodology on the hub in Hilversum and compare the order selection performance of the machine learning models with the current method, i.e., the use of 'optimal' delivery levels explained in Section 2.1, and the Daganzo estimation. Section 5.1 elaborates on the predictive performance of the machine learning models to support the order selection decision. Next, Section 5.2 elaborates on the experiments of the Monte Carlo simulation in which the machine learning models are utilized to select orders. This simulation indicates the robustness of the order selection. Section 5.3 provides an in-depth analysis of the performance to validate the solution methodology. Finally, Section 5.4 concludes the chapter.

5.1 Machine learning to support order selection

This section shows the findings and performance of the machine learning models that we will utilize to support order selection. First, Section 5.1.1 shows the tuned hyperparameters of the four models. Second, Section 5.1.2 elaborates on the performance of the models.

5.1.1 Hyperparameters

We tune the hyperparameters of RFR and LGBM as chosen and described in Section 4.3.2.1. We distinguish between the machine learning models based on generated data (1) and grouped historical data (2). Table 5 summarizes the tuned hyperparameters and their respective optimal values per model. We use those values to fit the machine learning models.

Table 5 Hyperparameters of RFR and LGBM models

Model	Hyperparameter	Value (1)	Value (2)
RFR	Maximal number features for splitting	2	*2*
	Minimum number of data points per leaf node	1	1
LGBM	Maximal number of bins	255	130
	Learning rate	0.070091	0.105513
	Maximal tree depth	856,269	958,969
	Number of leaves	18	10

5.1.2 Performance

We fit the generated training data (1) and historical grouped training data (2) on the RFR and LGBM model with the Scikit-Learn (Scikit-learn, 2023) and the LightGBM (Microsoft Corporation, 2023) libraries. Table 6 shows the performance on the test dataset of the literature estimation of Daganzo and the two regression models trained and tested on the generated data. Likewise, Table 7 shows the performance of the models trained on the grouped historical data and tested on (unseen) Hilversum data. Note that Table 6 compares the 3 daily distance estimations with the generated daily distance (see Section 4.3.3.2), and Table 7 with the observed distance in the historical data (see Section 4.3.3.1). Appendix C shows the performance of the grouped data, both the aggregated model of the small and large hubs, on all hubs. We compare the models with four performance metrics, the mean absolute percentage error (MAPE), the R^2 adjusted for the number of features, the root mean squared error relative to the mean of the target value (rRMSE) and the mean absolute deviation relative to the mean of the target value (rMAE).

We observe in Table 6 that the two regression models outperform the literature estimation on all performance metrics. The predictions of the Daganzo estimation are reasonable, but not perfect. It explains more than half of the variance by the inputs, but still an adjusted R^2 of 0.603 is not very high. The MAPE and rMAE show that the model deviates with relatively large values from their intended targets. Furthermore, the rRMSE shows that there are relatively large outliers. This performance can be explained by the assumption made for the capacity and the area. We used the expected order volume to determine the number of orders a vehicle could serve. Additionally, multiple vehicles can serve the same PC4, and the area from the served PC4s is not necessarily the smallest possible area containing all served orders as assumed by Daganzo (1987).

The RFR is the best-performing model, with a slightly better performance than LGBM. Both have a high adjusted R^2, and the MAPE, rRMSE and rMAE are small enough for distance prediction. The performance of both models is comparable to the operational level RFR and LGBM of Akkerman & Mes (2022), with a somewhat higher adjusted R^2 and a lower MAPE, rRMSE and rMAE, and slightly worse than Nicola, Vetschera & Dragomir (2018) and the best model (6) of Figliozzi (2008). This indicates that with the tactical level information, a performance comparable with the operational level information can be achieved.

Table 6 Model performances on generated data Hilversum

Generated data (1)	**Daganzo**	**Random Forest**	**Light GBM**
MAPE	42.14 %	8.28 %	10.08 %
Adjusted R^2	0.603	0.984	0.982
rRMSE	44.74 %	8.71 %	10.19 %
rMAE	35.47 %	6.17 %	7.26 %

Similarly, we observe in Table 7 that both regression models on the grouped historical data outperform the Daganzo estimation. Striking is that the adjusted R^2 of the literature estimation is negative, meaning that the model tends to be less accurate than the mean. Furthermore, the difference between the two regression models is even smaller. We observe that the regression models on the generated data set (1) have a better performance than the grouped historical data set (2). We can explain this difference by the fact that the first models are trained on Hilversum-specific instances, while the second models are trained on ten different small hubs, including Hilversum.

Table 7 Model performance grouped historical data (small hubs) Hilversum

Historical data (2)	**Daganzo**	**Random Forest**	**Light GBM**
MAPE	26.16 %	9.77 %	9.54 %
Adjusted R^2	- 0.011	0.873	0.863
rRMSE	28.27 %	10.45 %	11.56 %
rMAE	26.11%	8.64 %	8.55 %

We decide to continue with the literature estimation of Daganzo as a benchmark, and with the best-performing machine learning models RFR trained both on generated data (1) and grouped historical data (2) for the remainder of this research. We will compare these methods with the order selection obtained from CBB's current method, i.e., the use of the 'optimal' delivery levels. From now on, we refer to the order selection methods as Method 'Optimal levels', Method Daganzo, Method RFR (1), and Method RFR (2).

5.2 Monte Carlo simulation

We perform a Monte Carlo simulation to 1) assess the robustness of the order selections retrieved from the different order selection methods, and 2) compare the performance of the current heuristic, i.e., the use of the 'optimal' delivery levels, the Daganzo estimation and the machine learning models. For the first reason, we evaluate the effect of uncertainty in the PC4 order division. For the second, we account for the randomness in the assumed order volumes and order locations that were necessary to solve the VRP, and thus to compare the methods. Section 5.2.1 elaborates on the experimental design, Section 5.2.2 on the experimental result, and Section 5.2.3 on the sensitivity analysis. We built and run this simulation model in Python 3.9 on a laptop with an Intel Core i7-10750H CPU processor of 2.60GHz with 32 GB RAM.

5.2.1 Experimental design

This section elaborates on the experimental design of the Monte Carlo simulation. Section 5.2.1.1 estimates the parameters of the PC4 order division. Section 5.2.1.2 explains the number of replications necessary to obtain statistically significant results and Section 5.2.1.3 elaborates on the experiments to be conducted.

5.2.1.1 PC4 Order division

We start with the estimation of the PC4 order distribution that is the (main) source of stochasticity in the simulation. To find the discrete probability distribution belonging to a PC4, we first plot the observed data (green) against the Poisson, binomial, geometric and negative binomial distributions (errred) to inspect the fit. Figure 26 shows an example of such a plot for PC4 1211 in Hilversum that follows a negative binomial distribution. Likewise, we find that 36 of the 45 PC4s in Hilversum follow a negative binomial distribution based on a significance level of 0.05. The other 9 PC4s have no significant relation with all four distributions, based on the Chi-Square goodness of fit test with a significance level of 0.05. Therefore, we decide to use the empirical distribution for these PC4s in our simulation. We refer to Appendix D for the Chi-Square p-values and the estimated parameters of the PC4s.

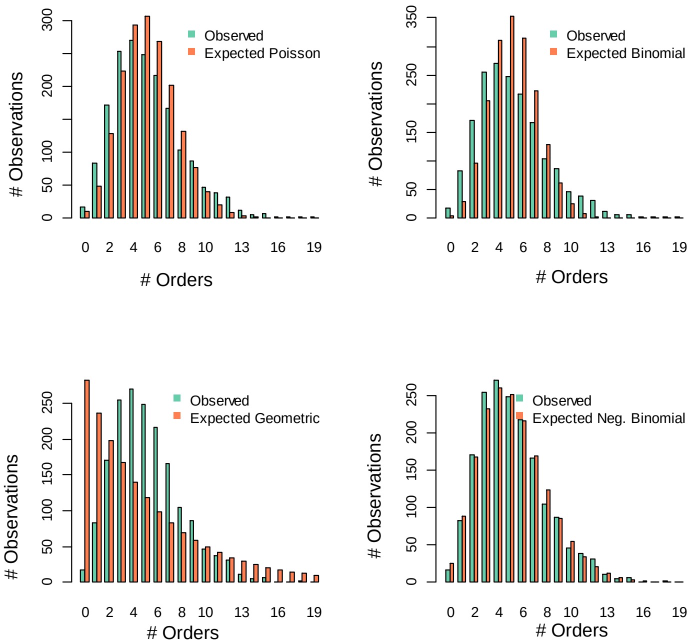

Figure 26 Distribution plot of PC4 1211

5.2.1.2 Number of replications

We determine the number of replications to obtain the required precision based on Law (2015). We decide that the result is statistically significant with an error of 2.5% and a confidence interval of 95% for the number of orders selected. Hence, we first run the typical 1000 replications for a Monte Carlo simulation (Heijungs, 2020) with the instances from Section 5.2.1.3. Afterwards, we execute the sequential procedure and determine per replication if the result is statistically significant.

We find that we need to conduct 7 replications to get statically significant results. However, we decide to perform 10 replications to decrease the influence of random number streams and obtain more statistically significant results.

5.2.1.3 Experiments

CBB wants to know the most cost-efficient capacity to deploy based on the total potential orders. Hence, we should decide how many orders to deliver and the model should be able to cope with different potential order volumes. Therefore, we simulate the order selection with the minimum observed potential orders (40), the average observed potential orders (140) and the maximal observed potential orders (460). We compare the performance with the metrics costs per order and the most cost-efficient number of orders selected.

Table 8 summarizes the instances of the four alternative methods for the Monte Carlo simulation for the hub in Hilversum. For every instance, we distinguish the value, by which method the instance is required (during which simulation phase), and how the data is partitioned. We refer to Method 'Optimal levels' as A, Method Daganzo as B, Method RFR (1) as C and Method RFR (2) as D. Note that we set the number of vehicles to infinity such that it will never be a limiting factor, to ensure the model advises about the optimal capacity to deploy. Furthermore, we determine the number of orders a vehicle can serve for the Daganzo estimation by dividing the vehicle capacity by the average order volume of that day.

Table 8 Hilversum instances for the Monte Carlo simulation

Instances	Value	Method	Data partition
Potential orders	(40,140,460)	A,B,C,D	Hub
Order division	~ Negative Binomial (s, p) ~ Empirical distribution	A,B,C,D	PC4
Order locations	~ Random	A,B,C,D (solve VRP)	Order
Order volume	~ Empirical distribution	A,B,C,D (solve VRP)	Order
Vehicle capacity	160000 cm3	A,B,C,D (solve VRP)	Bike
Number of bikes	∞	A,B,C,D (solve VRP)	Hub
Duration limit	206.08 minutes	A,B,C,D (solve VRP)	Bike
Stop times	Static	B,C,D (order selection), A,B,C,D (solve VRP)	PC4
Bike speed	18.8 km/h	B,C,D (order selection), A,B,C,D (solve VRP)	Hub
Fixed costs	4373.5 %	B,C,D (order selection), A,B,C,D (VRP costs)	Hub
Variable costs/order	9.8 %	B,C,D (order selection), A,B,C,D (VRP costs)	CBB
Variable cost/hour	289.5 %	B,C,D (order selection), A,B,C,D (VRP costs)	CBB
Variable cost/km	2.1%	B,C,D (order selection), A,B,C,D (VRP costs)	CBB
Optimal levels	1, 2, 3	A (order selection)	Hub

5.2.2 Experimental results

We report the order selection and costs per order of Method 'Optimal levels', Method Daganzo, Method RFR (1), and Method RFR (2). Table 9 shows the median (Q2) and the inter quartile range (Q1-Q3) to indicate the spread of the results. We verified the results with CBB's capacity specialist. We report the median and inter quartile range of the distance, travel time, necessary time, number of routes, number of postal codes and the orders deliver per hour belonging to the experiments in Appendix E.

Table 9 Performance Monte Carlo simulation of all four models

Number potential orders	Metric	'Optimal levels'		Daganzo		RFR (1)		RFR (2)	
		Q2	Q1-Q3	Q2	Q1-Q3	Q2	Q1-Q3	Q2	Q1-Q3
40	Order selection	38	38 - 39	40	40 - 40	40	40 - 40	40	40 - 40
	Costs per order (%)	168.2	163.2 - 169.8	165.2	163.6 - 166.7	166.7	165.2 - 167.0	165.2	163.6 - 166.7
140	Order selection	133	132 - 134	140	137 - 140	137	135 - 140	140	140 - 140
	Costs per order (%)	68.6	68.0 – 69.7	68.7	68.5 – 691	69.1	68.7 – 69.3	68.5	68.2 – 69.3
460	Order selection	439	438 - 446	454	439 - 460	451	436 - 460	438	433 - 450
	Costs per order (%)	40.9	40.7 – 41.0	40.8	40.5 – 41.0	40.8	40.6 – 41.2	41.0	40.6 – 41.5

We observe that for 40 potential orders, Methods Daganzo, RFR (1) and RFR (2) select exactly the same number of orders in the interquartile range. This stresses that the maximal number of orders gives the lowest costs per order according to these methods. The costs per order of Method RFR (1) are slightly higher than Method Daganzo and Method RFR (2), indicating that the minimum order selection size differs, which resulted in higher costs per order. The range of the costs per order is the smallest with Method RFR (2) and the largest with Method 'Optimal levels'.

With 140 potential orders, we again see that Methods Daganzo, RFR (1) and RFR (2) select a higher number of orders ((skewed to) 140) than Method 'Optimal levels'. However, where we observe no variation with Method RFR (2), somewhat variation and the same median with Method Daganzo, we observe more variation and a different median with Method RFR (1). Strikingly, there is no intuitive relation between the number of orders selected and the costs per order in this experiment. Method 'Optimal levels' gives the second-best median of costs per order, although it selected the least orders. This shows that the order selection heuristic applied to Methods Daganzo, RFR (1) and RFR (2) does not necessarily lead to the lowest costs per order, although the difference is only a few cents.

With 460 potential orders, we find that the costs per order inter quartile ranges get smaller, the difference between the methods' costs per order gets smaller, and the order selection inter quartile ranges get wider and tend less towards the number of potential orders. Hence, we need relatively more orders to slightly improve the costs per order. Remarkable is that Method RFR (2) selects the least orders, whereas with 40 and 140 orders it selected the maximal number of orders. Furthermore, note that Method 'Optimal levels' in the three experiments always finds the lowest observed costs per order. Again, this might imply that the order selection heuristic applied to Methods Daganzo, RFR (1) and RFR (2) does not necessarily lead to the lowest costs per order

We conduct a paired t-test with 95% confidence and find that with 40 and 140 potential orders the difference in costs per order is for no method combination significant. For the order selection difference, we conduct a Wilcoxon Signed-Rank Test (Rey & Neuhäuser, 2011) with 95% confidence because we cannot assume a normal distribution, see Figure 28. We explain the choice for the Wilcoxon Signed-Rank Test, the assumptions of, and the difference between this test and the paired t-test in Appendix F. The difference in order selection with Method 'Optimal levels' is significant for Methods Daganzo, RFR (1) and RFR (2), but the mutual difference is not significant. With 460 potential orders, the difference between all the order selections and the costs per order is not significant. Figure 27 and Figure 28 visualize the respective boxplots.

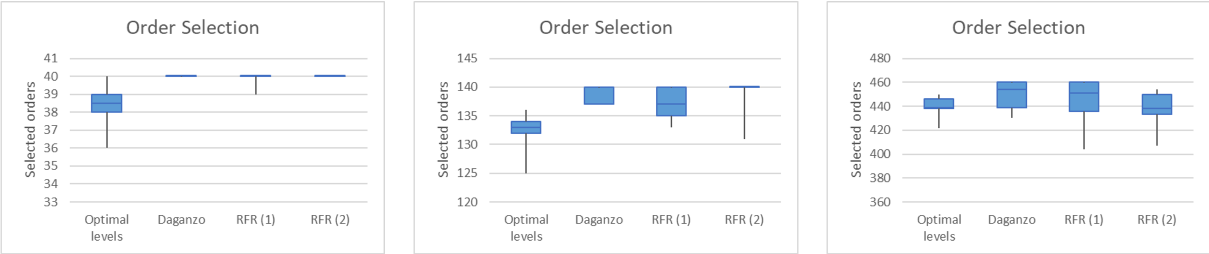

Figure 27 Boxplots order selection with 40, 140 and 460 potential orders

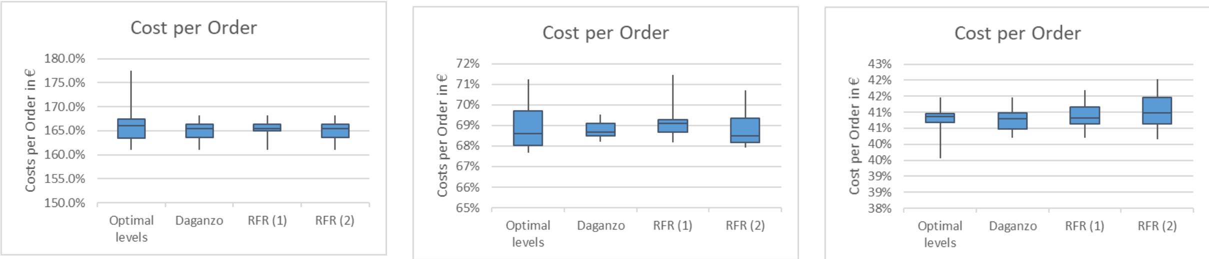

Figure 28 Boxplots costs per order with 40, 140 and 460 potential orders

In general, Method Daganzo tends to select more orders. Surprisingly, the performance of Method Daganzo is comparable to both Method RFR (1) & RFR (2), despite the worse predictive performance observed in Section 5.1.2. This indicates that the predictive importance is not very important under the current costs. Method RFR (2) tends to perform worse, i.e., relatively higher costs per order, when the number of potential orders increases. This is probably due to fewer observations of many potential orders in the training data and overestimation in the distance of the model, i.e., the distance is not based on the haversine distance with which the VRP is solved. However, the difference in costs per order is not significant with methods that select more orders. Method RFR (1) seems to have a wider spread in order selection, possibly because the training data includes more variety in order selection with a similar distance. We can explain the comparable performance of Method 'Optimal levels' by the fact that CBB manually optimized this area for the current costs and that it seems optimal for the hub in Hilversum to select at least 90% (Levels 1, 2 & 3, see Section 2.1.2) of the orders from the delivery area.

5.2.2.1 Analysis per number of potential orders

To obtain a better insight into the behaviour of the methods subject to the number of potential orders, we decide to run the model for a range of 20 to 680 potential orders with steps of 20. We report only one method, Method RFR (1), because the results of the four methods are similar. Figure 29 and Figure 30 show the spread order selection and costs per order for the range of potential orders. We observe that with more potential orders, the costs per order are still decreasing. Apparently, the costs per order converge towards a lower limit. This means that we need more potential orders to obtain a subset of orders with the lowest possible costs per order. Furthermore, we notice that the spread of order selection gets wider when the number of potential orders increases. When we have less than roughly 100 potential orders, for the model it is obvious that the order selection size should be (close to) maximal with the current costs (range of maximal 5 orders difference). This indicates that no matter the order division over the PC4s, it is always interesting to deliver (close to) 100 orders to account for the fixed costs. Afterwards, we observe more order selection variation, which indicates the trade-off between the fixed costs and variable costs, resulting from the randomness of the PC4 order division. Hence, even though the costs per order have not reached a definite limit yet, because it is limited to the number of potential orders, selecting all potential orders does not necessarily lead to the lowest costs per order due to the random order division over the PC4s. Although the number of orders selected is less obvious, the inter quartile range of the costs per order is small and tends to get smaller when the number of potential orders increases. We report the median and inter quartile range of the distance, travel time, necessary time, number of routes, number of postal codes and the orders deliver per hour of all methods, belonging to the experiments, in Appendix E.

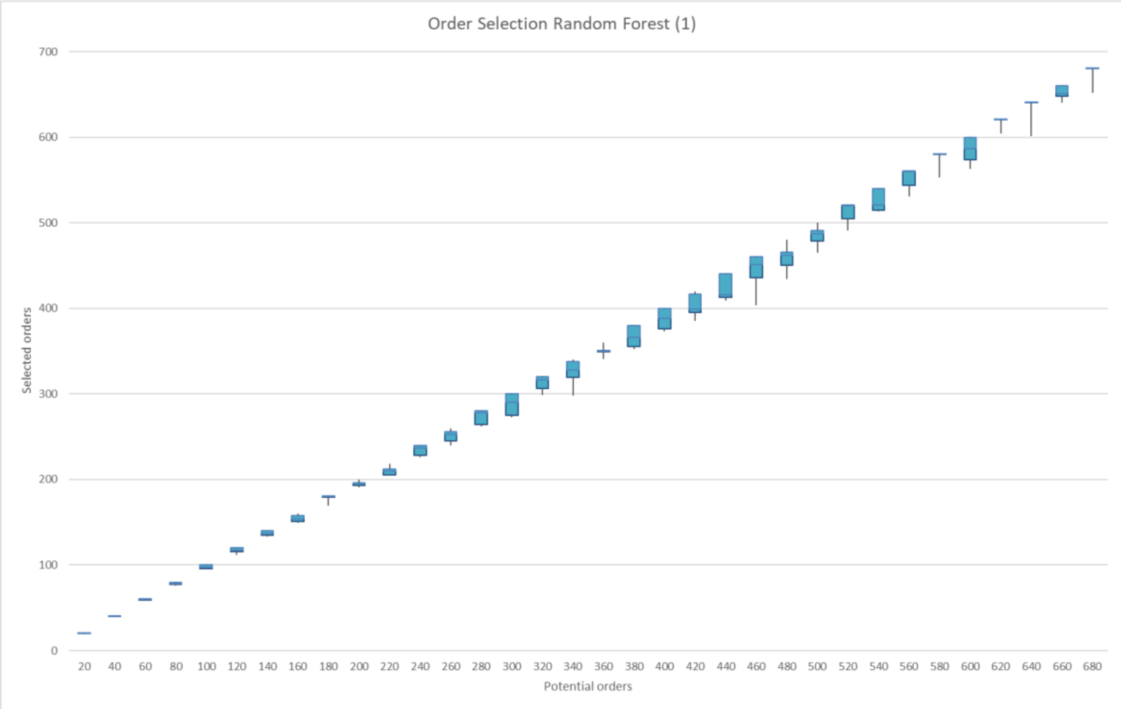

Figure 29 Order selection of Method RFR (1)

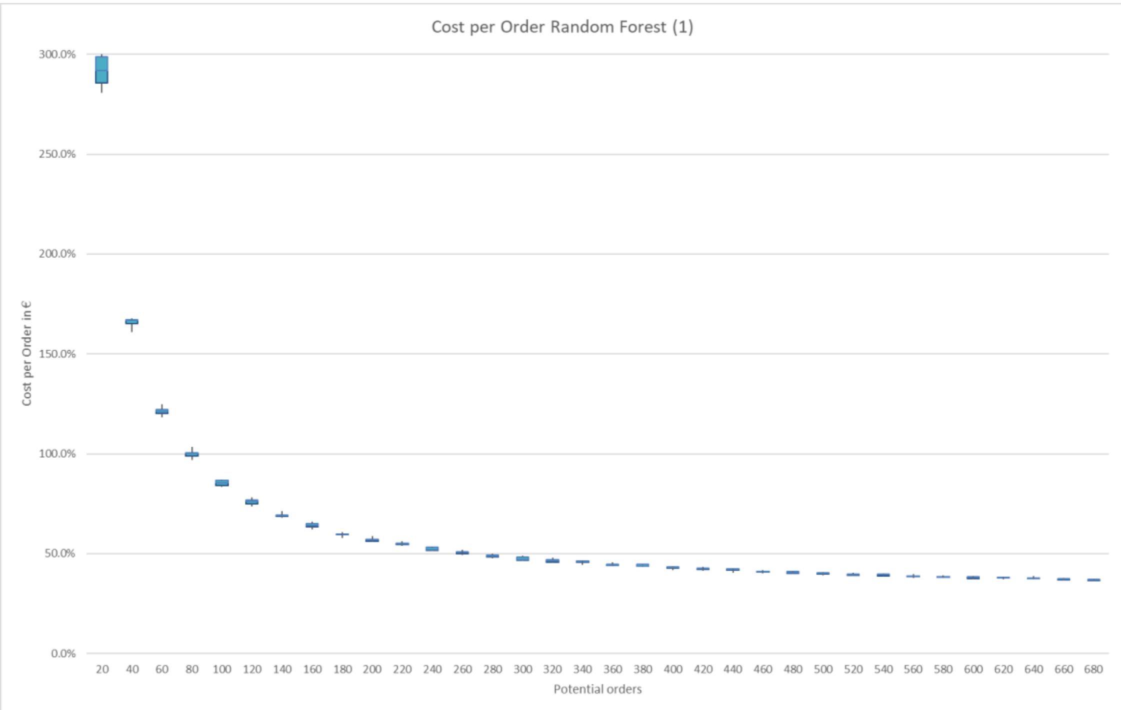

Figure 30 Costs per order of method RFR 1

5.2.3 Sensitivity analysis

We conduct a sensitivity analysis to learn if our solution methodology is valid. We observed that under the current circumstances, the order selection choice is trivial for all four methods: select a number of orders close to the maximal number of orders. To validate that our solution methodology selects the most cost-efficient order selection, we analyse the performance of the methods when circumstances change. Hence, this sensitivity analysis assesses how sensitive the most cost-efficient order selection is to changes in the input variables. We analyse variability in costs, forecast and route efficiency. We illustrate the effect on 140 forecasted potential orders in Hilversum. To compare the difference in order selection, we conduct a Wilcoxon signed-rank test with 95% confidence. To compare the difference in costs per order, we conduct a paired t-test with 95% confidence.

5.2.3.1 Costs

In the analysis, we used the current fixed and variable costs. However, these costs are subject to change due to e.g., inflation and new contracts. Furthermore, these costs are the driver for the decision of what order selection is the most cost-efficient. Hence, we expect the order selection choice of our solution methodology to be different when the costs change. Therefore, Section 5.2.3.1.1 tests the impact of changes in the fixed costs and Section 5.2.3.1.2 the impact of changes in the variable costs.

5.2.3.1.1 Fixed costs

First, we analyse the results with zero, half, full and one-and-a-half fixed costs. Figure 31 shows the boxplots of the order selection belonging to these fixed costs.

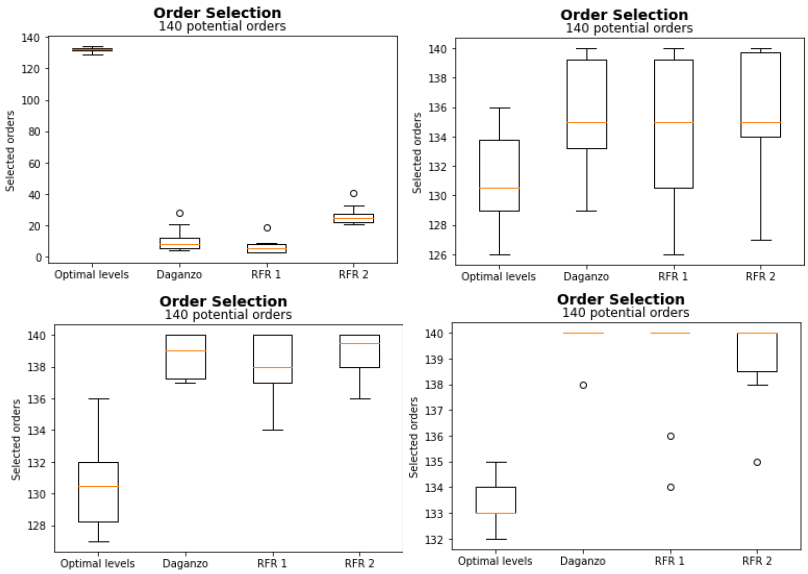

Figure 31 Boxplots order selection with zero (upper left), half (upper right), full (lower left) and one and a half (lower right) fixed costs

Zero fixed costs

Clearly, with zero fixed costs Methods Daganzo, RFR (1) & RFR (2) choose significantly fewer orders than with full fixed costs. Method ‘Optimal levels’ shows a similar order selection. This stresses that the Method ‘Optimal levels’ is not a valid order selection method: it chooses the same number of orders independent of the costs per order. Additionally, it shows that Methods Daganzo, RFR (1) & RFR (2) base their order selection decision on the costs per order. Straightforwardly, the costs per order of all methods differ significantly with zero fixed costs compared to full fixed costs.

Furthermore, Methods Daganzo, RFR (1) & RFR (2) choose significantly fewer orders than Method ‘Optimal levels’. Again, this shows the value of our solution methodology compared to the current system. The order selection of Method RFR (2) is significantly larger than Method Daganzo & RFR (1). This is because, with a few orders selected, Method RFR (2) overestimates the distance, resulting in higher variable costs. Figure 32 shows how it takes Method RFR (2) roughly 10 selected orders to observe similar curve behaviour as Methods Daganzo & RFR (1). Hence, the minimum of Method RFR (2) is around 24 orders whilst around 7 and 8 for Methods Daganzo and RFR (1). However, the difference in costs per order of the four methods is not significant. This is probably due to the low costs in general since we only consider variable costs.

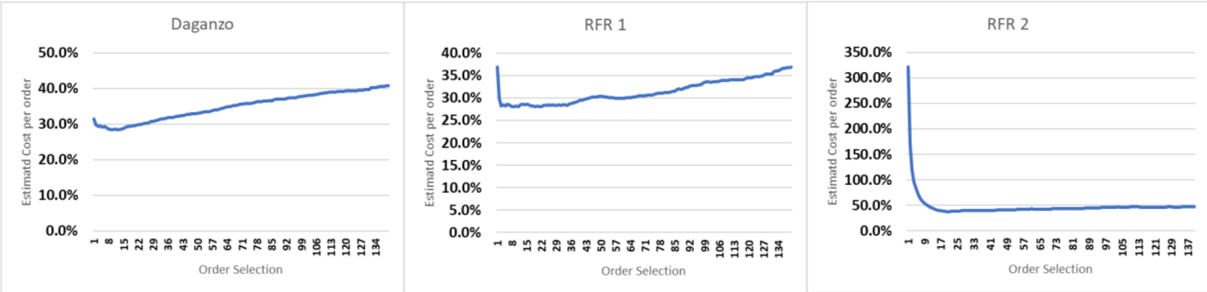

Figure 32 Costs per order per order selection based on the estimated VRP distance of Daganzo (left), RFR 1 (middle) and RFR 2 (right)

This experiment confirms the value of our methodology: it selects the subset of orders based on the current costs of delivering orders based on route efficiency, and decreases the costs per order between 3.9 % – 18.1 % compared to the current situation. Furthermore, it shows that Methods Daganzo and RFR (1) have a comparable performance. Although one might have expected the RFR (1) to outperform the Daganzo estimation, the similar performance could partly be explained by the similar features used. Both use the number of selected orders, the average distance to the hub and $\sqrt{\text{total area} * \text{selected orders}}$ to describe the distance. These feature values are determined the same, i.e., on PC4 level, for both methods. Additionally, the Daganzo estimation uses the maximal number of customers that can be served by a bike to describe the distance, and the RFR (1) uses the number of potential orders, the total area and the variance of the distance to the depot to describe the distance.

Half-fixed costs

With half, instead of full fixed costs, the inter quartile range is wider, less skewed to the number of potential orders and the minimum value is lower for all methods. This indicates that the order selection choice is less obvious than with full fixed costs, and the role of the PC4 order division is more important. However, the most cost-efficient order selection decision is comparable with full fixed costs. Accordingly, the difference in order selection between half and full fixed costs is not statistically significant. This indicates that the half-fixed costs still overrule variable costs per order, and thus the order selection decision. Straightforwardly, the costs per order of all methods differ significantly with half fixed costs compared to full fixed costs.

Similarly, as with full fixed costs, the order selection and the respective costs per order of Method ‘Optimal levels’ differ significantly compared to Method Daganzo & RFR (2). However, the order selection of Method ‘Optimal levels’ is not significantly lower than Method RFR (1), even as the costs per order. We can motivate this by the wider spread of Method RFR (1), and more overlap in the interquartile range of Method ‘Optimal levels’ & RFR (1). The mutual difference of the Methods Daganzo, RFR (1) & RFR (2) is not significant in terms of order selection and costs per order. Now, the lowest costs per order median are obtained from Method ‘Optimal levels’ (53.4%), followed by Method RFR (1) (53.9%), Method Daganzo (€54.0%) and Method RFR (2) (€54.2%). This might indicate that the order selection logic, i.e., cheapest insertion, does not always find the order selection with the least costs per order.

One-and-a-half-fixed cost
With one-and-a-half-fixed costs, Method Daganzo & RFR (1) have a stronger preference for 140 selected orders than with full fixed costs. Method RFR (2) behaves similarly, and the only difference for Method ‘Optimal levels’ comes from the randomness in order distribution. The difference in order selection for each method compared to the full fixed costs is not significant with 95% confidence. Again, this implies that the order selection decision is trivial; all orders should be selected to account for the fixed costs. Straightforwardly, the costs per order of all methods differ significantly with one-and-a-half-fixed costs compared to full fixed costs because of the rise in fixed costs.

Comparably with full fixed costs, the difference in order selection of Method ‘Optimal levels’ compared to Methods Daganzo, RFR (1) & RFR (2) is significant, but not in costs per order. Their median costs per order (84.2%, 84.4% & 84.5% respectively) are lower than the optimal levels (85.0%) that select fewer orders. Hence, this stresses that the choice for all (140) orders is necessary to account for the fixed costs. The difference between these costs is not significant. Also, the difference between Method Daganzo, RFR (1) & RFR (2) is not significant in terms of both order selection and costs per order.

We can conclude that there is no high impact on the order selection with 140 potential orders when the fixed costs in- or decrease by 50% because all orders should be selected considering the relatively high fixed costs. Furthermore, we observe that the Method ‘Optimal levels’ is insensitive to cost changes, i.e., the order selection choice is similar independent of the costs, although the difference in costs per order is not statistically significant.

5.2.3.1.2 Variable costs

Second, we examine the results when the variable costs in- or decrease by 50%. Figure 33 visualizes the impact of the order selection with 140 potential orders when the variable costs change.

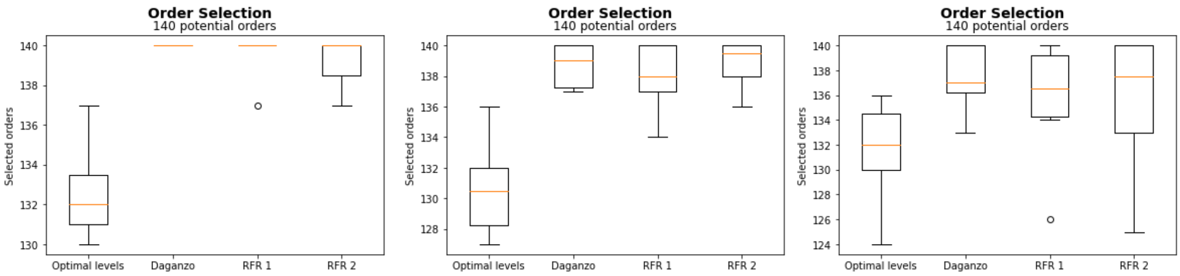

Figure 33 Boxplots order selection with 50% (left), 100% (middle) and 150% (right) variable costs

Half variable costs
We find that with half the variable costs, the order selection is quite similar to the case when there is a 50% increase in fixed costs. This shows that with 140 potential orders, it always seems worthwhile to select (almost) all orders. Similarly, the difference in order selection of the individual methods with half and full variable costs is not significant.

The difference in order selection of Method ‘Optimal levels’ compared to Methods Daganzo, RFR (1) & RFR (2) is significant, and the mutual difference of the latter is not. The difference in costs per order of Method ‘Optimal levels’ (53.4%), Method Daganzo (54.0%), Method RFR (1) (53.9%) and Method RFR (2) (54.2%) is with no combination significant.

One-and-a-half-variable costs
When the variable costs increase by 50%, we observe a comparable outcome with the experiment of half-fixed costs. It is now less obvious for Method Daganzo, RFR (1) & RFR (2) to select all potential orders compared to full variable costs, because it is relatively, to the fixed

costs, expensive to deliver an extra order. The difference in order selection compared to full variable costs is insignificant, and as expected the costs are significantly increased.

Strikingly, Method 'Optimal levels' has the smallest order selection and lowest median costs per order of 86.6% compared to 87.8% (Method Daganzo), 87.6% (Method RFR (1)) & 87.9% (Method RFR (2)). Hence, the three methods tend to select more orders than optimal and thus might underestimate the distance belonging to the order selection. Again, this might indicate that the order selection logic, i.e., the cheapest insertion heuristic, can be improved as discussed with half fixed costs. This costs per order difference of all methods is not significant. The order selection of Method 'Optimal levels' differs significantly with Method Daganzo & RFR (2), but strikingly not with Method RFR (1). This explains the second-lowest median costs of Method RFR (1). There is not enough evidence to state that the order selection of Method Daganzo, RFR (1) & RFR (2) differ.

Again, we can conclude that there is no high impact on the order selection when the variable costs in- or decrease by 50%. Thus, the fixed costs also overrule the variable costs when they in- or decrease by 50%.

5.2.3.2 Forecast

The input of our model is the total number of forecasted potential orders. However, this forecast is likely to deviate from the actual order demand. Coolblue defined a marge of +/- 10% forecast error to be acceptable. Hence, to test the effect of deviation on the forecast, we show the impact on the most cost-efficient order selection with +/- 10% orders. Figure 34 shows the results with 126 (90%), 140 (100%) and 154 (110%) potential orders. We only report Method RFR (1) trained on generated data, because all methods have similar results.

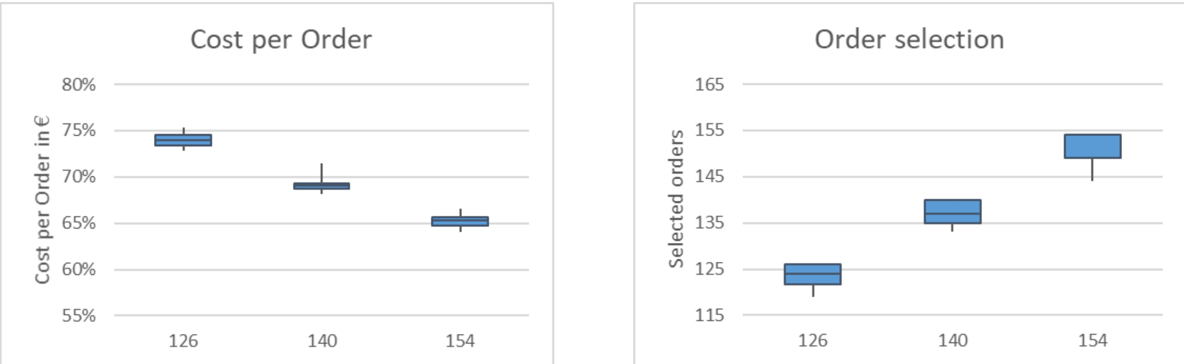

Figure 34 Boxplots of 126, 140 & 154 potential orders with Method RFR (1)

Figure 34 stresses two things. First, the sensitivity of the most cost-efficient order selection and the number of forecasted potential orders. We observe that all boxplots do not overlap. Again, the order selection is skewed to the maximal number of potential orders. We conduct a paired t-test and Wilcoxon signed rank test and find that the difference in respectively the costs per order and order selection is significant. Hence, the results are sensitive when the forecast deviates +/-10% with 140 forecasted potential orders. We learn from Figure 29 and Figure 30 that the order selection pattern is the same until at least 680 potential orders. However, the difference in costs per order gets smaller when the number of potential orders increases. This might lead to an insignificant difference in costs per order when the forecast deviates +/- 10%.

Second, it stresses the importance of an accurate forecast. The decision of how much capacity to deploy will be different depending on the forecast, as well as the cost savings. When we assume 140 potential orders what turned out to be 154 potential orders and consequently decide to select 137 orders (see Figure 34, Order selection 140), we could have saved around 69.1% - 65.3% = 3.8% per order (see Figure 34, Costs per order 140 & 154). The other way around, we would have overcapacity equivalent to delivering 137 – 124 = 13 orders (see Figure 34, Order selection 126 & 140).

5.2.3.3 Route efficiency

The model determines the route efficiency (orders per biked hour) based on the total haversine distance. This spherical distance does not account for actual roads, as well as possible detours and road disruptions. To understand how this impacts the robustness of the most cost-efficient order selection, we compare the performance of this haversine distance with a distance increase of factors 1.2 and 1.4.

We observe in Figure 35 that RFR 1 is most sensitive to changes in distance and selects fewer orders when the distance increases. However, for every method, the order selection is not significantly different with factors 1, 1.2 and 1.4, although it is more expensive to travel to an (extra) order. Straightforwardly, the costs per order increase for all methods because the travel costs increase. The difference in costs per order of the methods stays insignificant with 95% confidence with the distance factors 1, 1.2 and 1.4. Hence, the most cost-efficient order selection is insensitive to changes in route efficiency with the current costs.

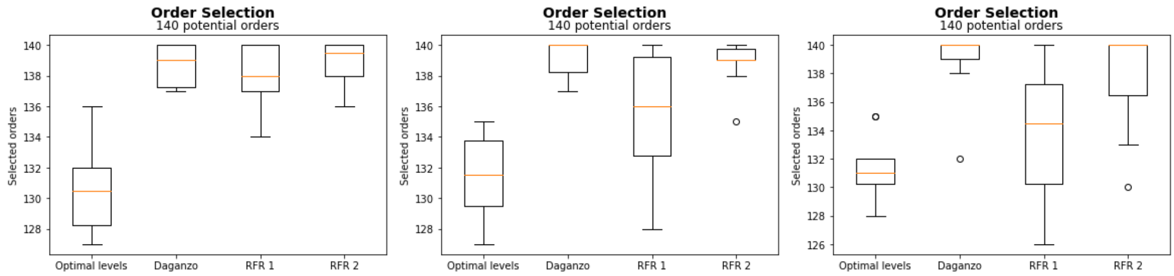

Figure 35 Boxplots order selection with factor 1 (left), factor 1.2 (middle) and factor 1.4 (right) travel distance.

5.2.3.4 Discussion

In the sensitivity analysis, we only saw during the experiment with zero fixed costs that our solution methodology did not choose an order selection close to the maximal number of potential orders. However, we did observe that the most cost-efficient order selection became less obvious when we put more weight on the selected orders, i.e., the variable costs, instead of the fixed costs. These observations stress two things; 1) that our solution methodology is capable of selecting the most cost-efficient order selection, i.e., it does not always select an order selection close to the maximal number of potential orders, and 2) that the fixed costs currently overrule the variable costs.

Table 10 summarizes the results of the sensitivity analysis, which presents for every intervention the median of the order selection and the costs per order per method. The method with the most cost-efficient order selection is highlighted in light blue. We observe that Method Daganzo outperforms the other methods most often. When Method RFR (1) or RFR (2) outperforms Method Daganzo, we see that the order selection median is the same or deviates only one order. Only when we have zero fixed costs, we find that the costs per order differ relatively much from the best-performing method, although not significantly. The spread, and thus the uncertainty, of the most cost-efficient order selection increases when there is more focus on the travel distance, with e.g., less fixed costs, more variable costs or an increase in distance. The order selection of the three methods is robust against route efficiency, i.e., distance, increases with factor 1.2 or 1.4, but differs significantly when the forecast in- or decreases by 10%. We learned that Method RFR (2) overestimates the costs per order, i.e., distance, when at most 10 orders are selected. This flaw can be neglected as the difference in costs per order is not statistically significant and in practice, we will most likely not encounter a situation where CBB delivers less than 10 orders. Furthermore, we observed that the Method 'Optimal levels' is insensitive to cost changes and distance increases, as it always selects roughly the same median order selection (range 130 - 133). This stresses the static, one-situation optimization of this method. Hence, it is not a valid method to select the most cost-efficient order selection. Lastly, we observed that the Method 'Optimal levels' always finds the lowest observed costs per order in one of the simulation runs. In combination with the outperformance of Method 'Optimal levels' when we put more weight on the variable costs per order, this might indicate that the order selection logic does not always find the order selection with the least costs per order.

Table 10 Summary of the sensitivity analysis

Sensitivity analysis	'Optimal' levels		Daganzo		RFR (1)		RFR (2)	
	Order selection	Cost per	Order selection	Cost per order	Order selection	Cost per order	Order selection	Cost per order
Current situation	133	68.6%	140	68.7%	137	69.1%	140	68.5%
0 Fixed costs	132	35.9%	8	34.5%	7	29.4%	24	29.9%
50% Fixed costs	130	53.4%	135	54.0%	135	53.9%	135	54.2%
150% Fixed costs	133	85.0%	140	84.2%	140	84.4%	140	84.5%
50% Variable costs	132	50.9%	140	49.9%	140	49.9%	140	50.0%
150% Variable costs	132	86.6%	137	87.8%	137	87.6%	136	87.9%
90% Forecast	119	73.6%	125	73.7%	124	73.9%	126	73.5%
110% Forecast	142	66.2%	154	65.0%	154	65.3%	148	65.7%
1.2 Distance	132	73.3%	140	72.8%	136	73.2%	139	73.0%
1.4 Distance	131	76.8%	140	76.9%	134	77.7%	140	76.4%

In general, we found that the current method 'Optimal levels' is not capable of choosing the most cost-efficient order selection. Our solution methodology with all three distance estimations can select the most cost-efficient order selection, validated by the experiment with zero fixed costs. However, the performance of the methods Daganzo, RFR (1), and RFR (2) is, unlike their predictive performance, comparable.

5.3 In-depth analysis solution methodology

To validate our solution methodology and to understand the comparable performance of the Methods Daganzo, RFR (1), and RFR (2), we perform an in-depth analysis. The similar performance is striking since their predictive performance was not similar. Therefore, we decide to analyze the predictive performance during the simulation. In the simulation, we currently estimate the distance of an order selection every time an order is added (Simulation Phase 2), and only solve the VRP for the order selection with the minimal estimated costs per order (Simulation Phase 3). To compare all the distance estimations with the actual distances, we decide to both estimate the distance and solve the VRP with the VRP solver in Python (see Section 4.4.3) every time an order is added to the order selection. Hence, we retrieve the estimated distance and the actual, constructed distance. Note that we estimate the distance of the order selections with the Daganzo estimation, RFR (1) and RFR (2), and construct the distance of these three order selections with the VRP solver. Accordingly, we can assess how much the estimated distance deviates from the constructed distance. This gives insight into the predictive performance of the methods. Additionally, the comparison of the constructed distances of each method shows what method obtains the order selection with the least distance, and thus what method actually chooses the most cost-efficient order selection. Figure 36 visualizes our solution methodology and the in-depth analysis, and thus the difference between the two. Note that we here divided the selection of the most cost-efficient subset of orders of Phase 2 into three steps (a, b, c). We build and run this in-depth analysis in Python 3.9. We report the results of 460 forecasted potential orders because this displays the differences the most clearly.

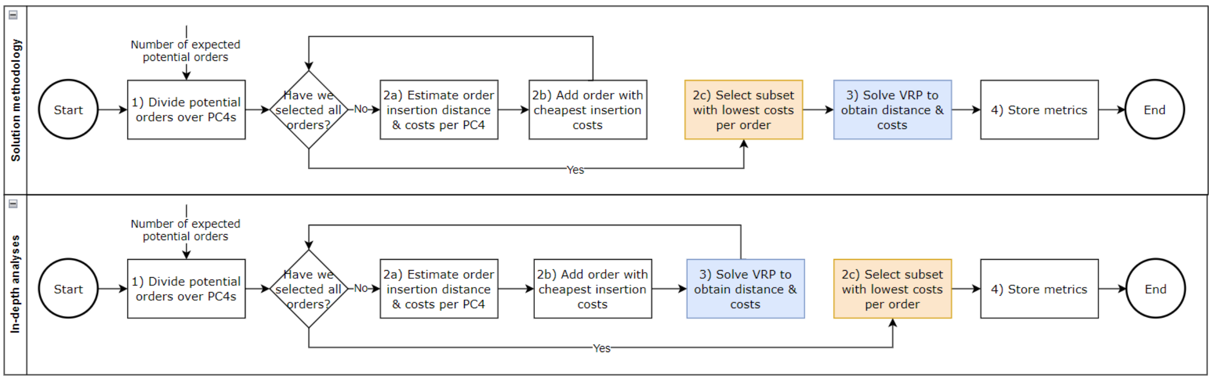

Figure 36 Difference in-depth analysis and solution methodology

Additionally, we develop an operational level RFR model that estimates the VRP distance to 1) show the value of a machine learning model such as RFR over a mathematical approach such as the Daganzo estimation, 2) show the difference in performance when we have all order information (operational level) and when we have uncertainty in all types of order information (tactical level), and 3) validate the development of our tactical RFR models, because we develop the operational RFR model with the exact same steps. We perform the abovementioned in-depth analysis also with the operational level RFR model.

Accordingly, Section 5.3.1 elaborates on the development of the operational level RFR model. Next, we conduct the in-depth analysis. Section 5.3.2 discusses the estimated and constructed distances of the orders selected by Methods Daganzo, RFR (1), RFR (2) and the operational level RFR to assess the predictive performance in the simulation, and the sequential order selection choice. Subsequently, Section 5.3.3 examines the costs per order of these order selections to understand how the costs per order affect the most cost-efficient order selection choice. Lastly, Section 5.3.4 discusses the findings of the in-depth analysis. Table 11 summarizes the terminology used in this section for clarification purposes.

Table 11 Terminology and description

Terminology	Description
Estimated distance	The estimated distance of Daganzo / RFR (1) / RFR (2) / the operational level RFR
Constructed distance	The constructed distance by the VRP solver in Python
Tactical level models	Daganzo, RFR (1), RFR (2)
Operational level model	The operational level RFR

5.3.1 Operational level RFR model

We develop the operational level RFR model similarly as the tactical level RFR models. Hence, Section 5.3.1.1 discusses the features, Section 5.3.1.2 the training data, Section 5.3.1.3 the hyperparameters, and Section 5.3.1.4 the predictive performance. Again, we illustrate the performance on the hub in Hilversum.

5.3.1.1 Features

The main difference between the operational and the tactical level RFR model is the order information available. At the tactical level, we have uncertainty in all types of order information. Therefore, we decided to describe the VRP distance with spatial features on PC4 level in Section 4.3.1. At the operational level, we usually know all the order information, e.g., location and demand. Thus, we can describe the VRP distance with this order information.

We decide to rephrase the set of tactical level features to operational level features because now we can describe those features in more detail. Accordingly, the total area in km^2 of the PC4s from the selected orders, becomes the smallest area in km^2 that contains all selected orders, i.e. the convex hull (Features 3 & 6). Figure 37 gives an example of a convex hull. Likewise, the average and variance of the distance of the centroid of the PC4s from the selected orders to the hub become the average and variance of the distance of the selected orders to the hub (Features 4 & 5).

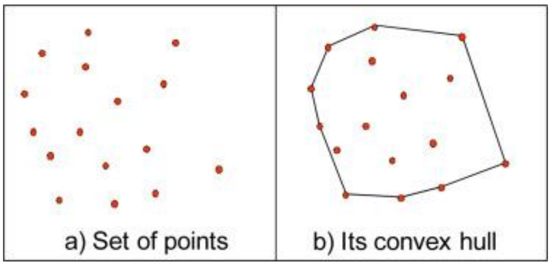

Figure 37 Convex hull (Laurini, 2017)

Additionally, we expand the set of features with operational level features. We choose to include the same features that Akkerman & Mes (2022) exploited after feature selection. Table 12 summarizes the 13 features of our operational level RFR model. The bearing β between an order and the hub, used in Feature 13, is the angle between the line connecting the two points and the north-south line of the earth (Akkerman & Mes, 2022). Figure 38 illustrates the bearing β. We use the Python library Scipy.spatial (Scipy , 2023) to determine the area and perimeter of the convex hull of all selected orders, and the library Statistics (Statistics, 2023) to determine the variance. For further information about the features and their determination, we refer to Akkerman & Mes (2022).

Table 12 Operational level features of the RFR model

Number	Feature	Description
1	Potential orders	The number of potential orders in the entire delivery area
2	Selected orders	The number of selected orders in the entire delivery area
3	Total hull area	The total convex hull area in km^2 of the selected orders
4	Average distance to hub	The average distance in km from the selected orders to the hub
5	Variance distance to hub	The variance in km from the selected orders to the hub
6	$\sqrt{\text{Total hull area} * \text{Selected orders}}$	The square root of the total convex hull area in km^2 of the selected orders multiplied by the number of selected orders in the entire delivery area
7	Total volume	The total volume in cm^3 of the selected orders
8	Total volume / Capacity	The total volume cm^3 of the selected orders divided by the bike capacity
9	Average volume	The average volume cm^3 of the selected orders
10	Hull perimeter	The perimeter of the convex hull of the selected orders
11	Average distance between orders	The average distance in km between the selected orders
12	Total distance to hub	The sum of the distance in km from the hub to the selected orders
13	Variance bearing to hub	The variance of the bearings between the selected orders and the hub

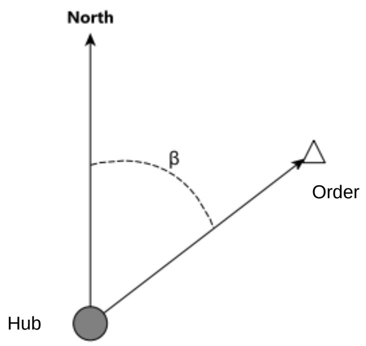

Figure 38 The bearing between the hub and an order (Akkermans, 2021)

5.3.1.2 Training data

We decide to generate data to develop the operational level RFR model. We use the exact same approach as discussed in Section 4.3.3.2, including the steps to ensure the order selection variability (see Section 4.3.3.2.5). The only difference is that the features stored in

Phase 4 (see Figure 21) are now the operational level features, instead of the tactical level features.

Again, we split the data into a train, validate and test set with a respective 80%, 10%, and 10% ratio. We train and test the models in Python with the Scikit-Learn (Scikit-learn, 2023), see Section 4.3.2, and tune the hyperparameters on the validation set using Bayesian optimization with 5-fold cross-validation with the Scikit-Learn and ScikitOptimize Python libraries (Head, Kumar, Nahrstaedt, Louppe, & Shcherbatyi, 2021), see Section 4.3.2.1.

5.3.1.3 Hyperparameters

We tune the hyperparameters of RFR as chosen and described in Section 4.3.2.1. Note that we again set the number of trees in the forest to 200 and let the trees grow to full depth, similar to the tactical level RFR model. Table 13 summarizes the tuned values of the hyperparameters.

Table 13 Hyperparameters operational RFR model

Hyperparameter	Value
Maximal number features for splitting	7
Minimum number of data points per leaf node	1

5.3.1.4 Predictive performance

We fit the generated data on the RFR model as described in Section 4.3.2. Table 14 displays the performance metrics. We observe that this operational level RFR model slightly outperforms the tactical level RFR (1) model (MAPE: 8.28%, Adjusted R^2: 0.984, rRMSE: 8.71%, rMAE: 6.17%). However, the rRMSE of the tactical level RFR (1) model is slightly better (8.71% compared to 8.77%). Furthermore, the operational level RFR model more obviously outperforms the tactical level RFR (2) model (MAPE: 9.77%, Adjusted R^2: 0.873, rRMSE: 10.45%, rMAE: 8.64%).

Table 14 Model performance of the operational level RFR model on generated data Hilversum

Performance metric	Value
MAPE	6.78 %
Adjusted R^2	0.986
rRMSE	8.77 %
rMAE	5.89 %

5.3.2 Distances of the order selections

We perform the in-depth analysis and retrieve the estimated distances from the Daganzo estimation, RFR (1), RFR (2) and the operational level RFR, and the respective constructed distances. Section 5.3.2.1 elaborates on the predictive performance of the methods to understand how the estimations deviate from the constructed distances. Section 5.3.2.2 elaborates on the capability of choosing the orders with the least distance increase to understand the order selection choices.

5.3.2.1 Predictive performance

We elaborate on the predictive performance to understand how accurate the estimated distances are compared to the constructed distances in the simulation. Table 15 shows the mean (μ) and 95% confidence interval (95% CI) of the computed performance metrics of the distance estimations. Clearly, the operational RFR model has a very good performance, and thus can make accurate predictions in both the simulation and on the test set. This shows that the development of the operational RFR model, and therefore also the development of RFR (1) and RFR (2), is valid. Furthermore, we observe that the operational RFR model makes better predictions than the tactical level models. This stresses that estimations improve when there is more order information available. The operational RFR explains on average 22.7% more than the similar model with tactical-level features (RFR (1)).

Furthermore, we find that the mutual predictive performance of all three tactical level models indeed differs. Surprisingly, the Daganzo estimation seems to perform the best, and thus better than observed in Section 5.1.2. RFR (1) outperforms RFR (2), which is in line with the observed predictive performance. Also, we observe that the confidence interval width of the tactical level models is wider than the operational level model. This is caused by the difference in order information. The tactical level models only know the number of orders located in a PC4 and therefore make predictions based on an average order in a PC4, whilst the operational level model makes predictions based on the exact order location and volume. Hence, the tactical level model chooses from what PC4 an order is added and the operational level model chooses what order to add. Because the tactical level models assign a random address to the chosen PC4 for validation purposes (see Section 4.4.3), the address could be both representative or an outlier for the PC4. The address choice determines how accurate the prediction is, and thus the random choice causes more spread in the performance metrics.

Table 15 Performance metrics of the distance estimations during the order selection

Performance metric	Daganzo		RFR (1)		RFR (2)		Operational RFR	
	µ	95% CI	µ	95% CI	µ	95% CI	µ	95% CI
MAPE	20.55	[19.16, 21.94]	33.02	[30.73, 35.31]	46.42	[41.76, 51.09]	19.87	[9.71, 30.02]
Adjusted R^2	0.794	[0.764, 0.823]	0.756	[0.696, 0.815]	0.598	[0.506, 0.690]	0.983	[0.975, 0.991]
rRMSE	27.13	[24.53, 29.72]	32.06	[28.22, 35.90]	39.96	[34.60, 45.32]	9.16	[7.47, 10.84]
rMAE	19.76	[17.76, 21.76]	28.34	[25.23, 31.45]	34.06	[29.39, 38.73]	6.65	[5.15, 8.15]

To understand the predictive performance and the respective deviations better, we show the estimated and constructed distance of the order selections made during a simulation run by the Daganzo estimation, RFR (1), RFR (2), and the operational level RFR in Figure 39. In these graphs, we indeed observe different behaviour of the four estimations.

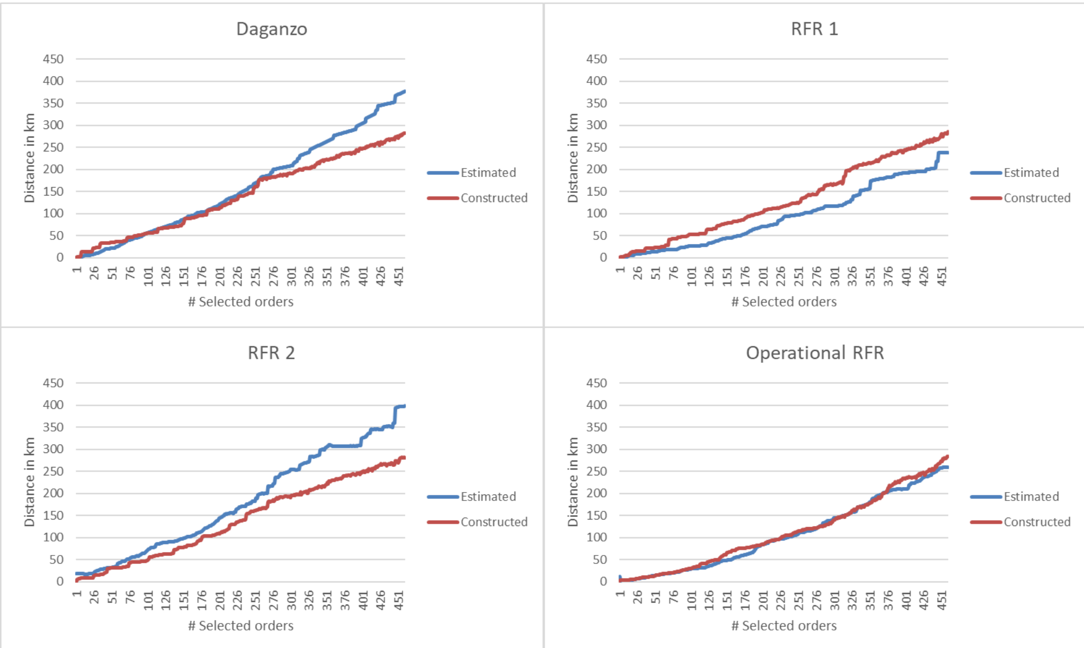

Figure 39 Estimated and constructed distance of the order selection of methods Daganzo (upper left), RFR 1 (upper right), RFR 2 (lower left) & operational level RFR (lower right)

The Daganzo estimation first underpredicts the actual distance and starts to overpredict from 95 selected orders. Clearly, this estimation is not properly aligned with the situation of CBB, because it does not mimic the constructed distance. The estimation seems to be reasonable at the beginning, but the larger the number of orders, the larger the difference between the estimated and actual distance. Additionally, first underpredicting and afterwards overpredicting is undesirable, because this gives a distorted view of the minimal distance increase.

Method RFR (1) underpredicts the distance. The prediction is quite accurate at the beginning, but afterwards, it starts to more obviously underpredict. Although the estimated distance curve rises at a different number of orders than the constructed distance curve, in general, it does seem to resemble the constructed distance.

Method RFR (2) overpredicts the distance. The overprediction can be explained by the fact that the distance in the training data of RFR (2) is the real-life distance and not the haversine distance. The real-life distance accounts for actual roads and possible road disruptions, and therefore is larger than the haversine distance. From around 251 orders selected, the difference between the estimated and actual distance tends to get larger when the number of selected orders increases. Additionally, it also does not really mimic the curve behaviour of the constructed distance. This could imply that the training data did not contain sufficient variability, i.e., observations with ≥ 251 orders selected. Also, it could imply that the actual distance gets larger if more orders are selected, due to for example the selection of orders in more sparsely populated and vast areas, which is not properly reflected in the haversine distance.

The operational level RFR seems to be very close to the actual distance, as expected by the performance metrics. This shows the potential of machine learning models in the estimation of VRP distance: it explains 18.9% more than the mathematical approach of Daganzo.

5.3.2.2 Order selection capability

We elaborate on the order selection capability to explain the most cost-efficient order selection choice made. The distance travelled drives the most cost-efficient order selection choice because the distance is reflected in the variable costs. Hence, the method that obtains the order selection with the least distance, has the highest capability to select the most cost-efficient order selection. Figure 40 compares the constructed distances of the four methods. Straightforwardly, the four methods converge into the same distance, as in the end all the (same) 460 orders are selected. It shows that the operational level RFR obtains the order selection with the smallest distance all the time, and hence with the lowest costs per order (see Appendix I). Therefore, it selects the orders the smartest, i.e., with the least distance. This again stresses the advantage of knowing and utilizing order information. Furthermore, it shows that Method RFR (1) obtains 74.13% of the time the smallest distance from the tactical level models. Hence, it chooses the orders with the actual least distance. This is in line with the best tactical level predictive performance observed in Section 5.1.2. Method RFR (2) performs

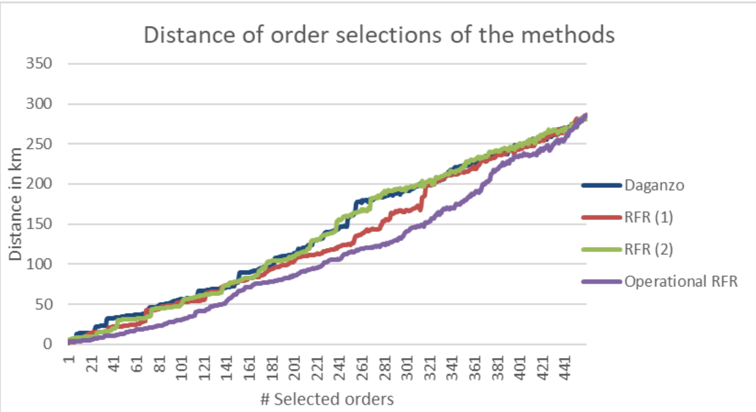

Figure 40 The constructed distance of the order selection of the 4 methods

slightly better than the Method Daganzo. Therefore, the RFR models outperform the Daganzo estimation in order selection capability.

To understand not only the variable costs but also the variable costs per order, we display the average distance travelled per order. Figure 41 shows the constructed and estimated average distance travelled per order of the four methods. The constructed distance travelled per order shows what method obtained the most cost-efficient order selection and the estimated distance travelled per order shows their distance estimation per number of orders selected. Again, we see that the operational level RFR obtains the lowest constructed average distance travelled per order. Method Daganzo has the highest constructed average travelled distance per order, and thus selects less favourable orders compared to the RFR methods. Accordingly, we observe that Method Daganzo retrieves an average travelled distance of maximal 3.2 times the distance found by the operational level RFR, whereas this is 2.3 times for Method RFR (1) and 2.7 for Method RFR (2). Note that the distance difference of the methods is wider at the beginning, because the distance increase is divided over fewer selected orders. Furthermore, we observe with the tactical level methods more sharp fluctuations than with the operational level method in the constructed distances. The tactical level methods only have order information on PC4 level. Therefore, they select an order based on the PC4. The actual location of this order is randomly drawn from an address list (see Section 4.4.3) once the order is selected, and could be the most remote corner of the PC4. Hence, we observe a sharp rise in average travelled distance if we select this order. Afterwards, the tactical level method usually selects orders from the same PC4, and thus relatively close orders. Therefore, the average travelled distance usually decreases again after a sharp rise.

Figure 41 The constructed (left) and estimated (right) average distance travelled per order of the 4 methods

Although the constructed average distance travelled per order of the operational level RFR is lower than the tactical level methods, we see in Figure 41 that the distance estimations of the tactical level RFR (1) and the operational level RFR are similar. This means that both methods make a similar decision regarding the most cost-efficient order selection. The estimated and constructed distance of RFR (1) differ because the orders within a PC4 are randomly chosen, and not based on the lowest distance increase as RFR (1) was trained on. This explains the different predictive performances observed in the simulation (Section 5.3.1.4) and on the test set (0). Therefore, the RFR (1) is capable of selecting the most cost-efficient **number** of orders to deliver, but not necessarily the most cost-efficient orders. Hence, this validates the use of a tactical level RFR to determine the most cost-efficient number of orders to deliver. Note that RFR (2) mimics the operational level RFR, but with an upward shift of around 300 meters. This indicates the potential of RFR (2) if it is evaluated on real-life distances instead of the haversine distance. The Daganzo estimations are close to the operational level RFR estimations when less than 30 orders are selected, but afterwards, it overpredicts the distance. Hence, it is more expensive to select an extra order, which gives a distorted view of these costs and thus of the most cost-efficient number of orders to select.

Furthermore, we observe in Figure 41 that the average distance travelled per order first decrease, and after the observed minimum tend to increase. The decrease followed by the increase in the average distance travelled per order is caused by the ‘smart’ order selection. First, the orders are selected that are closest to the hub. To deliver the first order, the delivery man has to cycle back and forth to the hub. For the second order, the delivery man has to cycle from the hub to the first order, from the first order to the second order, and from the second order to the hub. Because the second order selected is close to the first order, i.e., with the least increase in distance, the average distance travelled per order usually decreases as shown in Figure 41. Hence, cycling back and forth from the hub can be seen as the initial investment. Accordingly, all orders close by are selected, which are typically the orders from the hub’s surrounding PC4s.

Figure 42 shows an example of how all the potential orders are spread over the entire delivery area of Hilversum. Here, the orange house is the hub. Because the hub is located in the middle of the densely populated metropolitan area of Hilversum, the PC4s surrounding the hub usually observe the most potential orders per km^2, simply because it has the most inhabitants per km^2. Therefore, the average distance travelled per order can become quite small: in this example, the minimum of the average distance travelled per order is 259 meters with 49 selected orders with the operational level RFR. Hence, it is very efficient to deliver the surrounding orders of the hub.

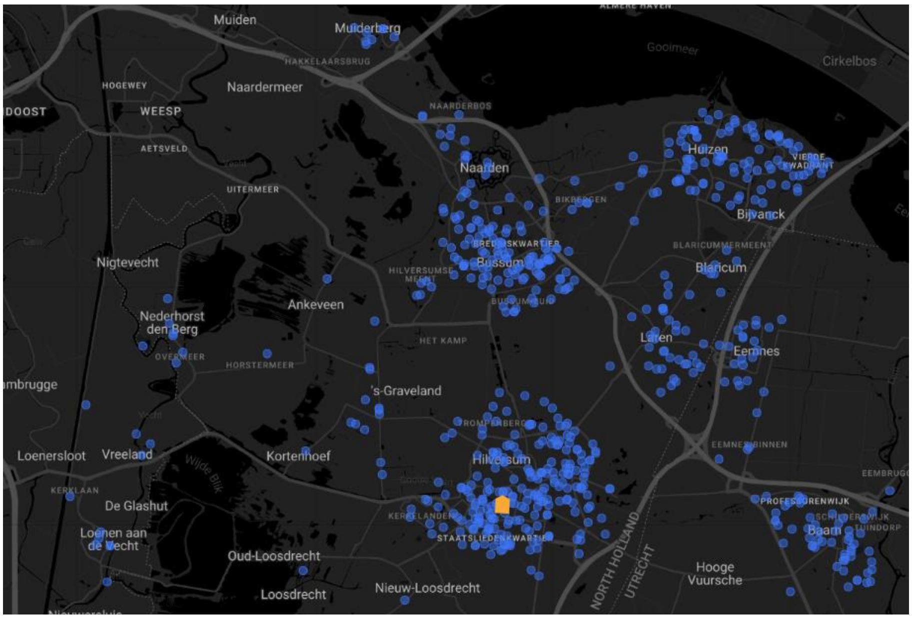

Figure 42 The spread of the potential orders over the delivery area of Hilversum

After these 49 orders, we observe that, in general, the average distance travelled per order increases. This is because 1) the area in which CBB cycles increases, and thus the average distance between the hub and the selected orders, and 2) these additional areas, in which CBB now also cycles, are more sparsely populated compared to the city of Hilversum, where the hub is located and thus where CBB initially cycles. However, Figure 41 does not only show increases, but also decreases. This is a result of the order clusters as observed in Figure 42. To deliver the orders in such a cluster, a delivery man first has to cycle to the cluster. Again, this can be seen as an initial investment, i.e., it results in a (temporary) sharp rise in average distance travelled per order (see, e.g., 113 selected orders in Figure 41). Once arrived, the extra distance to deliver an additional order in the cluster is usually small. If this extra distance is smaller than the current average travelled distance per order, the latter decreases when the order is selected. Hence, we observe multiple decreases in the average travelled distance per order. Because of the distance that has to be cycled to the cluster, and because the clusters usually have fewer inhabitants than the cluster in which the hub is located, i.e., a lower order density, it is unlikely that the minimum average travelled distance per order (259 meters with 49 selected orders) is reached again. Therefore, the average travelled distance per order increases after the observed minimum. Note that the distance in- or decrease is not always clearly visible in the graph, because it is divided over all selected orders.

In general, Figure 40 and Figure 41 show the value of using machine learning models over a literature approach such as Daganzo. These models are capable of selecting the orders that increase the overall distance the least. Also, the figures show that the order selection methods obtain difference distances for the same number of orders selected. Furthermore, we validated the use of a tactical level RFR model to determine the most cost-efficient number of orders to deliver.

5.3.3 Costs per order of the order selections

Although the order selections differ in terms of in distance to deliver the selected orders, the four methods make a comparable decision. This is caused by the comparable costs per order per number of selected orders. Therefore, Section 5.3.3.1 and Section 5.3.3.2 explain the behaviour of the fixed and variable costs per order of the order selections respectively. Finally, Section 5.3.3.3 elaborates on the total costs per order.

5.3.3.1 Fixed costs per order

Figure 43 shows the fixed costs per order per number of orders selected. As anticipated, we see that the fixed costs per order decrease the most with the first selected orders and converge to zero. Hence, it is most interesting to select the maximal number of orders based on the fixed costs per order only. Note that the fixed costs are independent of the distance estimation, and therefore, they are the same for the operational and all the tactical level methods.

Figure 43 The fixed costs per order per number of orders selected

5.3.3.2 Variable costs per order

Figure 44 shows the estimated and constructed variable costs per order of the order selection made by RFR (1). We refer to Appendix J for the graphs of the other three methods. The graph displays the variable costs per order, variable costs per kilometre per order and variable costs per hour per order, stacked per order selection. Straightforwardly, the variable costs per order are 9.8% per order for any order selection. Therefore, the fluctuations are caused by the variable costs per order per hour, and per kilometre, mostly visible in the costs per hour due to the higher price (the price per kilometre is only 0.73% of the price per hour). Hence, the fluctuations are caused by the travel time and the distance of the selected orders. Note that the time and distance are not 100% correlated, as the travel time includes the time to travel the average distance per order and the stop time per order.

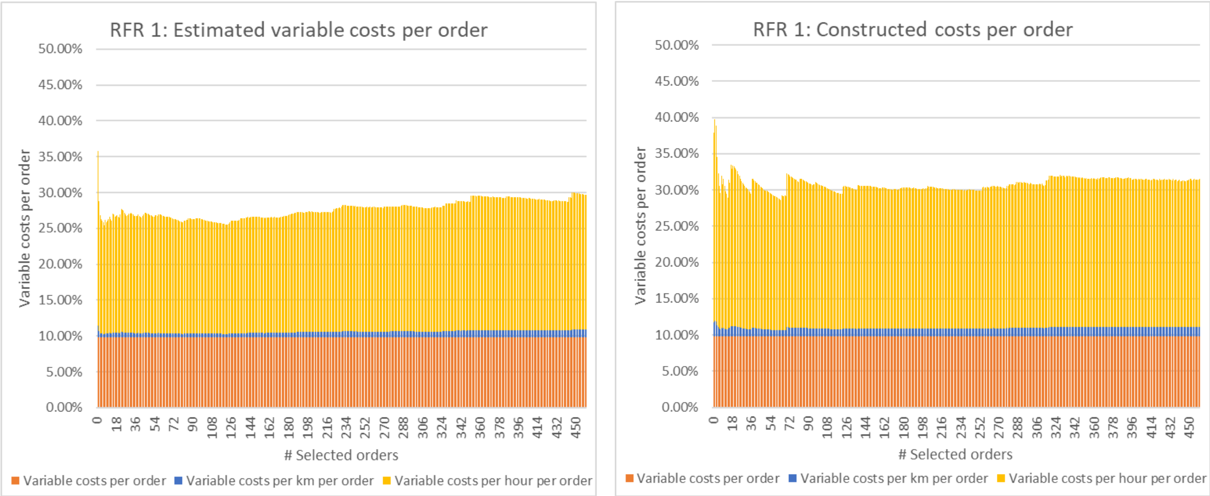

Figure 44 The estimated (left) and constructed (right) variable costs per order composition of the order selection of RFR 1

We observe that the variable costs per order per hour and per kilometre mimic the average travelled distance per order (see Figure 41). To illustrate, the minimum of the constructed distance and costs are both at 64 selected orders. However, where the average distance travelled per order more than doubles from 64 to 460 selected orders, the variable costs per order per hour only increase by 10.3% from 64 to 460 selected orders. The stop time per order accounts for this. Every order always has a stop time that is reflected in the variable costs per hour, independent of the time to travel the average distance per order. Hence, the variable costs per hour per order increase with the increase in time to cycle the average distance travelled per order relative to the sum of the average stop time per order and the time to travel the average distance per order.

Also, we find that the constructed variable costs per order have more sharp rises than the estimated variable costs per order. Hence, RFR (1) underestimates variable costs, i.e., distance increases, mostly obtained from delivering to an additional PC4. The sharp rise is a direct cause of the random address drawn from the PC4 as explained in Section 5.3.2.2.

Furthermore, we observe in Figure 44 that the difference in variable costs per order gets smaller when the number of selected orders increases. This is because the extra distance and time to deliver an additional order is divided over all selected orders. Hence, the difference in variable costs per order converges to zero. Therefore, the order selection choice is more important when not all orders are selected.

5.3.3.3 Costs per order

To understand how the fixed and variable costs per order influence the total costs per order, and thus the order selection choice, we analyse their behaviour simultaneously. Figure 45 shows the total costs per order of the order selection of RFR (1). We only display the costs per order belonging to the constructed distances, because the estimated distances show a comparable result. We observe that the costs per order decrease in general. In this particular run, the minimal costs per order are 4.09% when 460 orders are selected. Note that we observed an interquartile range of 436 – 460 selected orders, see Section 5.2.2, and thus that the most cost-efficient order selection choice is not always obviously 460 orders. Also, we observe that the ratio of fixed and variable costs per order changes. First, the majority of the costs per order consist of fixed costs per order, and as more orders are selected, the ratio revolves around. Lastly, we observe some local minima, e.g., when 69 or 318 orders are selected, indicated with a circle in Figure 45.

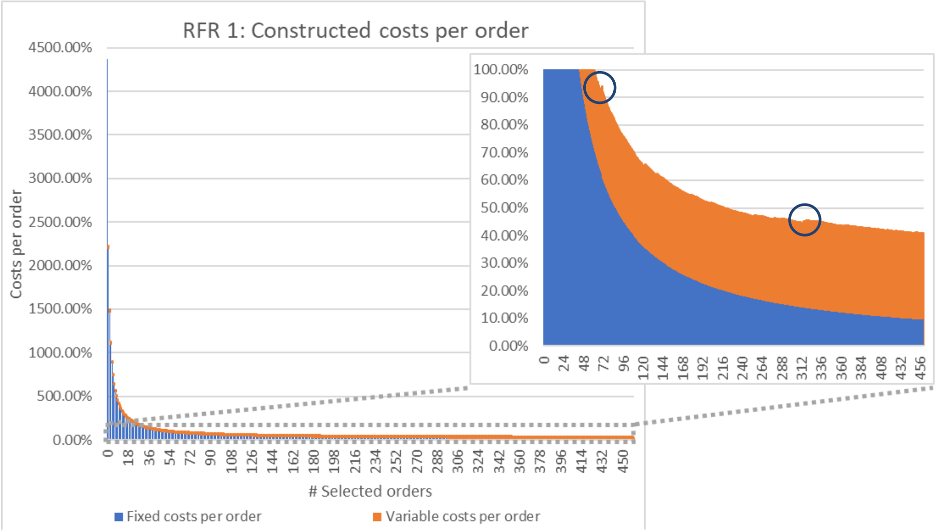

Figure 45 The constructed costs per order of the order selection of RFR (1)

Figure 46 displays the difference in fixed costs per order and variable costs per order of the constructed distances of Methods Daganzo, RFR (1), RFR (2) and the operational level RFR. The costs per order decrease if the decrease in fixed costs per order is larger than the increase in variable costs per order. Straightforwardly, the costs per order decrease if both the fixed and variable costs per order decrease. We observe that indeed in some cases, e.g., when 69 or 318 orders are selected, the increase in variable cost per order is larger than the decrease in the fixed costs per order, and thus that the costs per order increase. This increase is temporary because the sum of the costs per order decreases from these local minima (e.g., 69 or 318 orders) to the maximal number of selected orders (here 460) is larger than the respective sum of the costs per order increases. The minimum would be a global minimum if the sum of the increases is larger than the sum of the decreases. This is the case in other runs, where less than 460 orders were selected, and thus where the minimum costs per order was not observed at 460 orders (see Section 5.2.2). Note that the costs per order increase more with the tactical level models than with the operational level model due to the less smart order selection choice.

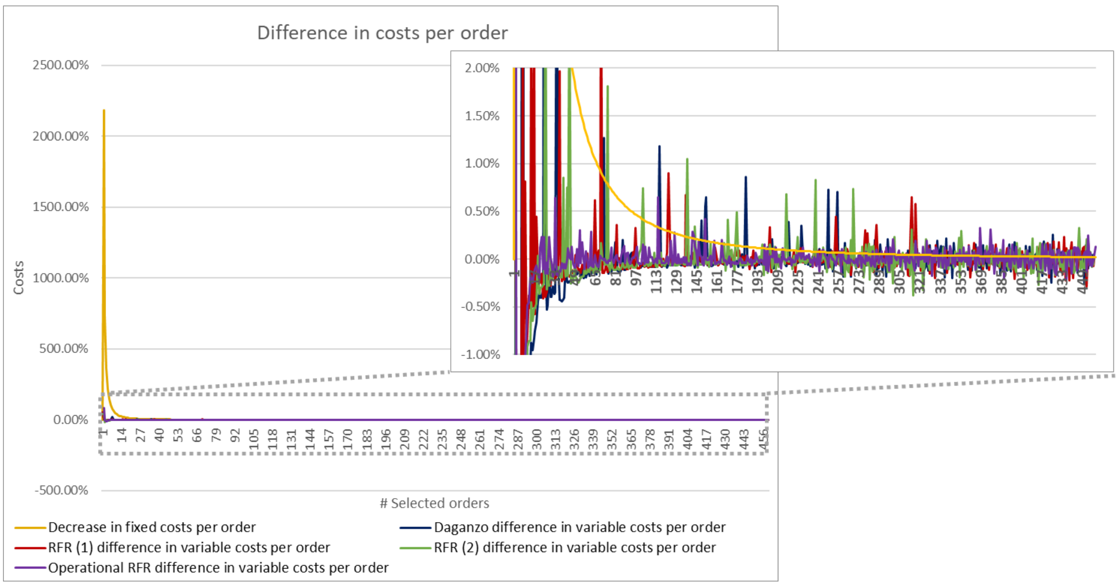

Figure 46 Difference in costs per order per order selection

Because the sum of all the costs per order decreases in Figure 46 is larger than the sum of all the costs per order increases, the costs per order decrease in general. Broadly speaking, the decrease in fixed costs per order overrules the in- or decrease in variable costs per order. Although sometimes the costs per order increase as a result of cycling to a cluster and/or adding an order from a new PC4, the reduction in fixed costs per order of selecting all the orders in the cluster or PC4 is still greater than the increase in variable costs per order resulting from the respective extra travelled distance. Accordingly, the choice to select (almost) all potential orders, when we observe the maximal number of potential orders, is trivial with the current fixed costs.

5.3.4 Discussion

To make an order selection decision based on the variable costs per order, i.e., the estimated distance of the four methods, the fixed costs should be sufficiently low that the reduction in fixed costs per order is overshadowed by the in- or decrease in variable costs per order. Currently, the fixed costs per order overrule the fluctuations in variable costs per order. Hence, the fixed costs are a dominant factor, and it is straightforward to select the maximal number of orders. Therefore, we observe that Methods Daganzo, RFR (1) and RFR (2) select a similar number of orders. Even with half fixed costs or one and a half variable costs, the fixed costs still overrule. The most cost-efficient order selection decision becomes less obvious as we reach the maximal number of potential orders (460). Here, the only reason to deviate from the maximal number of orders is the order division (over the PC4s) as observed in Section 5.2.2.1. However, the number of orders selected was still close to the total number of potential orders. Additionally, the fixed costs per order difference with, e.g., 300 to 301 orders is around only 0.1%, which explains why the observed impact in costs per order of the order selections in Section 5.2.3 is not significant.

Because RFR (1) underestimates the distance, and thus the variable costs per order, even lower fixed or higher variable costs are necessary to have an impact on the order selection with the order selection methodology. Hence, underestimation means that more orders might be selected than is most cost-efficient. Oppositely, overestimation of the distance causes more spread in the order selection, which explains the spread in the observation of Method RFR (2) with an increase in variable costs or a decrease in fixed costs (see Section 5.2.3.1).

5.4 Conclusions

This chapter showed the performance of the solution methodologies. First, we showed that the RFR and LGBM machine learning models can predict the distance of an order selection accurately and that the models trained on the generated data outperform the models trained on the historical data.

Second, we found that the number of orders per PC4 can often be described with a negative binomial distribution. We elaborated on the experimental design and discussed the experimental results. Hence, we found no significant difference in the most cost-efficient order selection found by Methods Daganzo, RFR (1) and RFR (2). We observed that the solution methodology is capable of selecting the most cost-efficient number of orders, and that the three methods are dynamic and robust against changes in costs or distance increases in the current situation, unlike the current order selection method.

Third, we conducted an in-depth analysis to understand the comparable performance of Methods Daganzo, RFR (1) and RFR (2). Additionally, we developed an operational level RFR model to validate the development of, and compare the performance with the tactical level models. We found that the tactical level RFR (1) is capable of selecting the most cost-efficient number of orders to deliver because it makes similar predictions as the operational level RFR. Hence, it can accurately estimate the distance belonging to the most cost-efficient number of orders. We found that the fixed costs per order converge to zero when the number of orders selected converges to infinity. Similarly, we found that the difference in distance per order, and thus the difference in variable costs per order, converges to zero when the number of orders selected converges to the maximal number of orders. Therefore, we learned that if we desire to make an order selection decision based on the variable costs per order, i.e., the estimated distance, the fixed costs should be sufficiently low that the reduction in fixed costs per order is overshadowed by the in- or decrease in variable costs per order. Alternatively, the variable costs should be sufficiently low, or the increase in travel distance and time should be sufficiently large to enhance the same effect. We conclude that with the current circumstances, it is straightforward for Methods Daganzo, RFR (1) and RFR (2) to select a number of orders close to the maximal number of orders.

6 Conclusions and evaluations

This chapter answers the main research question and concludes the research in Section 6.1. Section 6.2 states the recommendations based on this research for Coolblue. Furthermore, Section 6.3 elaborates on the practical and scientific contributions of this research and Section 6.4 states the limitations and recommendations for further research.

6.1 Conclusions

To determine the most cost-efficient capacity to deploy per hub at the tactical level based on the forecasted potential orders to decrease the costs per order of CBB delivery, we answer the main research question:

How should the optimal required capacity of CBB's hubs be determined such that the costs per order of CBB delivery are minimized?

The required capacity is based on the order selection and translated to the number of delivery men hours and bike routes. We analysed the current situation and concluded that order selection is equal to all forecasted potential orders from a subarea of the delivery area. However, this subarea is static, i.e., independent of the forecast and the bike route efficiency.

We developed a solution methodology to determine the number of orders to deliver from a CBB hub with minimal costs per order. The development of this methodology consists of 2 main steps:

1) Develop machine learning model(s) that estimate(s) the costs of an order selection.

We concluded that the RFR and LGBM models are the most suitable machine learning models for our problem. We developed tactical level models that describe the potential orders on PC4 level. We both generated data and grouped historical data based on hub size, i.e., small or large hubs, to train, validate and test the models. We tested the performance of RFR and LGBM on the generated (1) and historical (2) data. We concluded that all models with 6 tactical-level features have a high predictive performance and that RFR is the best model to support the order selection.

2) Build a Monte Carlo simulation, in which the machine learning model(s) are utilized to support order selection decisions.

We concluded that a Monte Carlo simulation is the most suitable to determine the most cost-efficient order selection when there is uncertainty in how the orders are distributed over the PC4s. We found that the number of orders per PC4 can be described with a negative binomial distribution. We selected the most cost-efficient order subset based on the estimated insertion costs of RFR (1) & RFR (2), without solving a VRP. We concluded that the Daganzo estimation enables the most reliable comparison with the RFR models, and therefore we also estimated the insertion costs based on Daganzo. To answer the main research question and to validate the solution methodology, we also selected orders with CBB's current order selection Method 'Optimal levels', and we solved the VRP of the 4 retrieved order selections found.

We tested our proposed solution methodology on the hub in Hilversum. We studied and compared the performance of the four order selection methods in 3 experiments: 1: the minimum observed potential orders (40), 2: the average observed potential orders (140) and 3: the maximal observed potential orders (460). We observed that Methods Daganzo, RFR (1) and RFR (2) select significantly more orders than the Method 'Optimal levels' with 40 and 140 potential orders, but not with 460 potential orders. Contrary, this difference in costs per order is not significant. We concluded that under the current circumstances, the order selection choice is trivial for all four methods: select a number of orders close to the maximal number of orders.

To validate that our solution methodology selects the most cost-efficient order selection, we conducted a sensitivity analysis. We learned that when we put more weight on the distance estimation, and thus the selected orders, the most cost-efficient number of orders to select is less obvious, but not significantly different. The Methods Daganzo, RFR (1) and RFR (2) still have a comparable performance. However, we concluded that CBB's current order selection method is not sensitive to these changes in costs or distance and that it is not capable of choosing the most cost-efficient number of orders. With zero costs, the difference in order selection median is extreme, 132 orders compared to 8 (Daganzo), 7 (RFR (1)), 24 (RFR (2)), and results in higher median costs per order, 35.9% compared to 34.5% (Daganzo), 29.4% (RFR (2)), 29.9% (RFR (1)). This experiment confirms the value of our solution methodology: it selects the subset of orders based on the current costs of delivering orders, and decreases the costs per order between 3.9 % – 18.1 %.

We performed an in-depth analysis to understand the comparable performance of the Methods Daganzo, RFR (1), and RFR (2). Additionally, we develop an operational level RFR to show the performance difference between having all order information (operational level) and uncertainty in all types of order information (tactical level), and thus to validate the development and use of a tactical level RFR. The operational level RFR has a high predictive performance both in the simulation and on the test set. We learned that RFR (1) makes similar estimations as the operational level RFR, and thus that both models make a similar decision regarding the most cost-efficient order selection. Hence, we concluded that the tactical level model is capable of selecting the most cost-efficient number of orders to deliver. Furthermore, we concluded that if we want to make an order selection decision based on the travelled distance, the ratio of fixed costs, variable costs, travel distance and travel time should be such that the reduction in fixed costs per order is overshadowed by the in- or decrease in variable costs per order. Under the current circumstances in Hilversum, although the methods underestimate the distances, i.e., they evaluate with a haversine distance, the decrease in fixed costs per order overshadows the change in variable costs per order. Hence, the order selection decision of the three methods is straightforward, and thus comparable: select a number of orders skewed to the maximal number of potential orders.

The last step to answer the main research question is providing an implementation guide for the other hubs. We explained in Appendix G where to find or how to retrieve the input information of each hub and how to load this information into the simulation. Furthermore, we handed over the tool to Coolblue's data scientist team, such that they can implement the tool at other hubs.

We conclude that the proposed solution methodology with order selection based on RFR (1) is valid to determine the most cost-efficient number of orders selected. RFR (2) overpredicts the distance, but showed potential if evaluated on real-life data. The Daganzo estimation gives a distorted view of the distance and thus of the most cost-efficient number of orders to select. The Methods Daganzo, RFR (1) & RFR (2) have a comparable performance because of the current fixed and variable costs of the hub in Hilversum. Although the difference in costs retrieved from these solution methodologies is not statistically significant with 95% confidence compared to the current order selection method, the methodology is dynamic, robust, easily implementable and generalizable to other hubs, unlike the current method.

6.2 Recommendations

To realize the conclusions stated in Section 6.1, we recommend Coolblue to implement the solution methodology as described in Section 4.2 to determine the most cost-efficient capacity to deploy. Also, we observed some inefficiencies in the current way of working of Coolblue for which we have some improvement suggestions. Hence, we advise the following:

1. We recommend selecting orders in the solution methodology based on the distance estimation of RFR (1). This model proved to select the most cost-efficient number of orders.
2. To improve the costs per order of all hubs, we recommend implementing the solution methodology for all hubs. We suggest using the guide of Appendix G to find (how to retrieve) the input information of each hub and how to load this information into the simulation. We recommend comparing the order selection costs considering the distance estimation of RFR (1).
3. A strength of the simulation is that the input parameters are estimated on actual data, creating realistic output. Therefore, we suggest storing the input parameters and files of the simulation in Coolblue's data warehouse BiqQuery and to connect them to the Python model. This can easily be done with the Google Cloud BiqQuery library in Python. Hence, every time the model runs, the newest and latest data is used to obtain the most cost-efficient capacity to deploy in the current situation without any manual effort.
4. We recommend utilizing the capacity tool as an advisory tool and not as a binding tool. The tool is developed with the goal to understand the impact of capacity deployment per hub, but it is not comprehensive and should not be treated as such.
5. We suggest running the tool once for every number of forecasted potential orders per hub under the current circumstances. We recommend storing the optimal capacity advice retrieved from these runs, such that the tool does not have to rerun when CBB encounters the same number of potential orders with the same input parameters. We advise rerunning the tool only when the input parameters or the current circumstances change.
6. We recommend assigning ownership of the capacity model to ensure its continuity. We found a team of data scientists of Coolblue with whom we have discussed the tool, and who are willing to and capable of owning the tool. Hence, we handed over the tool to this team and provided a guide with the (storage place of the) information necessary to implement the tool at other hubs.
7. During data analysis, we found that Coolblue stores the historical potential order information only on PC4 level, whilst they store the information of the delivered orders on order level. Hence, we had to estimate the order information in our research based on the orders that were delivered with CBB. However, we recommend storing historical order information on order level. This could facilitate more in-depth and representative analysis. Accordingly, more data is usually more preferable than having too less data.
8. Currently, Coolblue stores a lot of information in Google Sheets. However, there exist a lot of sheets. These sheets are not connected, and they are owned by different employees. Therefore, it is hard to find the information one is looking for. Hence, we suggest storing all information in one central place, for example in Google Drive, and to document where to find specific types of information.
9. A lot of information on the current working methods of Coolblue is not documented, but stored in the head of the employees. However, if an employee is sick or leaves Coolblue, the knowledge will be gone. Therefore, we recommend documenting the current processes, and the responsibilities and activities of every employee such that the information is always available.

6.3 Contribution to practice and theory

This section discusses the contribution of this research both to practice and literature in Section 6.3.1 and Section 6.3.2 respectively.

6.3.1 Practical contribution

This research stressed the urge to improve the capacity planning process of Coolblue by showing the impact of capacity planning on the costs per order. Hence, it showed the potential of dynamic capacity determination depending on the forecasted potential orders.

Furthermore, we developed a solution methodology that advises on the most cost-efficient capacity to deploy at the tactical level. Also, we showed the practical implementation of machine learning models for order selection in routing problems. Although our solution methods did not improve the costs per order statistically significantly, the method is valid and, unlike the current situation, dynamic and robust against changes in the input variables such as the costs.

Additionally, the methodology indicates how efficient a hub is and where the improvement potential lies to obtain lower costs per order, i.e., for Hilversum in increasing the number of potential orders. We showed the impact of the fixed and variable costs, route efficiency and forecast accuracy on the capacity, and stressed the importance of the latter.

We provided a tool easily generalizable to other hubs, if we know the order distribution, bike speed, stop time, area, distance to the depot per PC4 and the fixed and variable costs per hub. We presented a guide in Appendix G with this information, or how to retrieve this information for other hubs. Also, the tool is easily adjustable to the CBB distance estimation and route planning mechanism by using their Application Programming Interface (API) in Python, to which we did not have access. An API is a software interface that allows two computer programs, in this case, Python and Coolblue's route planning program, to communicate with each other. Utilizing this API returns representative distance estimations and bike routes. Hence, the tool can easily be integrated with changing operational circumstances, e.g., from the cherry-picking project.

Furthermore, with a few adjustments, the tool can be used for order selection and route planning at the operational level as well. We recommend using the operational level RFR model developed in Section 5.3.1 for this purpose because it is capable of selecting the most cost-efficient orders, and not only the most cost-efficient number of orders as the tactical level models can. Accordingly, we select orders based on the cheapest insertion costs estimated by the operational level RFR. Note that there is no order information uncertainty anymore, so the Monte Carlo simulation is not necessary. Hence, one replication is sufficient to determine the most cost-efficient order selection and the routes. For this implementation, the following 4 adjustments should be made in the respective phases of the simulation (see Figure 19):

1. **Phase 1:** The actual orders are known in this stage, and thus the potential orders do not have to be distributed over the PC4s. Hence, the potential order input list with random order information should be converted to the actual order list with the actual order information, which includes the latitude, longitude and order volume.
2. **Phase 2:** At this stage, the number of deployable delivery men hours and bikes are known and fixed. Hence, instead of adding orders until no potential orders are left, the orders should be added until there is no capacity or time left. Therefore, a feasibility check that ensures that the maximal tour time or bike capacity is not exceeded should be performed when evaluating what order to add. Because we know the distance increase, the travel time increase and the order volume, this check is easy to implement with just a few lines of code.
3. **Phase 2:** The subset with the lowest costs per order, determined by the distance estimation and the total number of selected orders, is the final order selection. However, now the costs per hour per order should be determined based on the hours of the deployed delivery men instead of the travel time. At this stage, delivery men are already hired and will be paid for the maximal tour time. Hence, instead of determining the tour time to determine the variable costs per hour, the variable costs per hour should be considered fixed costs equal to $\#\, delivery\ men\ per\ hour * maximal\ tour\ hours * salary\ per\ hour$.
4. **Phase 3:** To determine the distance between the customers and the routes to travel, the CVRP solver in Python should be replaced by the CBB distance estimation and route

planning mechanism API in Python. This easy and straightforward implementation returns the routes of the order selection and thus the route planning for the bike delivery.

Lastly, the solution methodology is generalizable to any tactical and operational (as described above) delivery routing problem where the most cost-efficient order selection to deliver should be determined. Although we proposed a methodology for bike delivery, the methodology is suitable for delivery with any vehicle if the travel speed is aligned with the vehicle choice.

6.3.2 Theoretical contribution

Although we conducted a case study for Coolblue, our research also contributes to scientific literature. In summary, we contribute to the literature with a methodology that selects the most cost-efficient order subset to determine delivery capacity in routing problems at the tactical level. This includes 1) an application of distance estimation, prior to solving, to support order selection decisions to determine the capacity to deploy, 2) an application of machine learning model(s) to estimate the variable costs of a VRP, 3) the provision of insight into the dependencies of several tactical level features with respect to the VRP distance instead of estimated or assumed operational level features, and its respective potential, 4) a method that takes variability and stochasticity into account and assesses the robustness of the optimal order selection given stochastic input sources, 5) a literature review regarding order selection in capacity planning and 6) an illustration of our solution method on the bike delivery case of Coolblue.

Most literature focuses on scheduling as many orders as possible given the fixed capacity, instead of determining the most profitable subset of orders to determine capacity on. Also, literature more often discusses order selection in routing problems at the operational level, which requires operational information unknown at the tactical level. We showed that with tactical level information, i.e., the spatial information of a PC4, we could retrieve a similar estimation as with operational level information. Therefore, this methodology might be interesting for the delivery capacity determination in routing problems at the tactical level of any organization. Furthermore, with some extensions, the model can also be used as an order selection method and route planning algorithm at the operational level, even with limited capacity (for implementation adjustments, see Section 6.3.1). The methodology is not only an answer to how much capacity to deploy but also indicates the robustness of the optimal capacity.

6.4 Limitations and further research

This research includes some limitations mainly coming from the scope, limited time and available data. We discuss the limitations and suggest Coolblue the following for further research to overcome these limitations:

1. In our solution methodology, we determine, compare and validate the most cost-efficient order selection based on a simplified distance estimation and a heuristic-based route planning mechanism. Hence, the output indicates the (behaviour of the) most cost-efficient order selection, and thus required capacity to deploy, but not an exact, CBB-tailored answer. Therefore, we recommend implementing the routing planning software of CBB into the solution methodology and studying the respective performance. This can be done by calling the route planning API of CBB in Python that returns the travel matrix and if necessary the routes of the given orders, which we did not have access to. This is an easy, straightforward implementation into the capacity tool that facilitates a realistic performance comparison. Once implemented, we recommend evaluating the performance of the Methods Daganzo, RFR (1) and RFR (2) again to assess if RFR (1) is still the most valid method. If the conclusion is that RFR (2) has potential, but that the grouped historical data does not contain sufficient variability, we recommend upsampling, i.e., generating or

duplicating data points of the underrepresented observations, and re-evaluating the performance. Note that for RFR (1), also new training data should be generated with CBB's distance estimation and route planning algorithm.

2. We determine the most cost-efficient order selection based on the cheapest insertion heuristic. We chose this heuristic to illustrate the improvement potential within the limited available time. This heuristic adds the cheapest order and subsequently searches for the next cheapest order to add. However, it does not evaluate all possible combinations, i.e., when it inserts the first cheapest order it assumes that this order is part of the optimal order subset. To find an order selection that is close(r) to the actual minimal costs per order, we suggest improving the order selection heuristic such that it analyses multiple order combinations. For the sake of computational time, we advise using a metaheuristic such as Tabu Search like Hernandez, Gendreau, & Potvin (2017). This metaheuristic can keep a (tabu) list of the costs of different order selection combinations. It accepts not only the order selection with the lowest costs per order but also worse solutions to explore if the combination has the best costs per order when an extra order is added. We suggest combining this implementation with Recommendation 1.
3. The solution methodology tries to minimize the costs per order per hub. However, minimizing all hubs individually does not necessarily imply that the aggregated costs per order of all hubs are minimized. For example, we divided some fixed costs, such as the office salary costs, (partly) proportional to the desirable number of orders to deliver. However, if CBB decides to deliver significantly fewer orders based on the analysis, the office salary costs per order increase. Therefore, it might be possible that the best scenario includes delivering more orders than the o most cost-efficient orders for one hub because this absolute increase in (variable) costs per order from o to $o+1$ orders is less than the absolute decrease in (fixed) costs per order from o to $o+1$ orders of another hub. Hence, we advise developing a collective capacity model that minimizes the costs per order of all CBB hubs simultaneously. This model could for example exploit the same order selection logic on all hubs and compare the aggregated costs per order of all possible combinations of hubs and order selections with a metaheuristic. It would be interesting to incorporate economies of scale into the model obtained from, e.g., linehaul costs, recruitment costs and office salary costs. To illustrate the effect of economies of scale, we suggest treating the linehaul costs as variable costs instead of fixed costs, and including the possibility to supply multiple hubs with the same truck or trailer. We suggest combining this implementation with Recommendations 1 and 2.
4. We assumed a linear relationship between the number of selected orders and the recruitment costs because of limited available time. Generally speaking, it is harder to find more to be recruited employees than only a few. However, although it points in the right direction, this relation is not necessarily linear. It is dependent on many features such as the tight or ample labour market, the number of extra biker delivery men that need to be hired, the hub location etc. Therefore, we suggest researching and implementing this relationship to obtain representative recruitment costs.
5. We would recommend researching how to increase the number of potential orders in Hilversum. We found that up to at least 680 potential orders the costs per order of the hub in Hilversum can decrease if the number of potential orders increases. The current average of 140 potential orders indicates that there is a lot of improvement potential for the costs per order. Additionally, with 460 potential orders, the average costs per order obtained from our tool are 18.71% cheaper than the average costs per order of DP from June 2021 to September 2022. Besides increasing market share, doing marketing activities and conducting market research, we advise simulating the following scenarios:
    a. We restricted the capacity tool to the fixed number of PC4s of the current delivery area of CBB, i.e., 45 PC4s in Hilversum. A possibility to increase the potential

orders is to enlarge the delivery area with more surrounding PC4s, e.g., > 45 PCs in Hilversum. To study the effect of additional PC4s in the delivery area, we recommend expanding the capacity tool with the surrounding PC4s of the current delivery area. We advise determining the order distribution of those PC4s based on historical orders of the PC4s that suffice the CBB delivery restrictions, similarly as explained in Section 4.4.1. The simulation can be conducted the same as described in Section 4.4. We would suggest studying additional interventions combined with the delivery area enlargement during this simulation, such as more available tour time and more bike capacity. Both might be necessary to obtain feasible routes with the enlargement of the delivery area. This might obtain insight into the potential of longer routes and bikes with more volume.

b. The current delivery area of Hilversum is adjacent to the delivery area of Amsterdam, Utrecht and Amersfoort. However, the hub in Hilversum is never allowed to deliver orders in the delivery area of one of these three. Multiple hubs encounter this phenomenon. Hence, we recommend researching the potential of delivering a PC4 from multiple hubs. This can in- or decrease the number of potential orders from the hub if it needs to be more cost-efficient, and it might create more efficient routes. Therefore, we suggest analysing a flexible, instead of fixed delivery area per hub, to find out if it is worthwhile to determine a daily delivery area per hub. We recommend combining this analysis with Recommendation 3.
c. We restricted the capacity tool to the volume restrictions of CBB. However, relaxing this constraint might lead to more order inflow for CBB and thus more potential orders. A downside can be that the bike capacity might be a bottleneck causing less utilized tour time compared to the maximal tour time, more time spend cycling from and to the hub, and subsequently less efficient routes. Hence, we recommend researching the allowance of larger orders, combined with more available tour time and more bike capacity. We suggest evaluating several volume dimensions and determining what volume restrictions lead to the lowest costs per order for multiple available tour time and bike capacity constraints.

6. The method determines the optimal number of orders to be delivered, to decide how much capacity to deploy. CBB wanted a model that advices on the capacity optimum, to understand the situation per hub under perfect conditions, solely making a trade-off between the fixed and variable costs. However, other costs such as overcapacity costs might be interesting to incorporate as well. This risk of overcapacity, and thus extra costs that should be divided over all orders, increases when CBB decides to deploy capacity close or equal to the total demand. Evaluation of under capacity is not necessary, since CBB simply outsources orders to DP when they have insufficient capacity. Therefore, we recommend evaluating the impact of the overcapacity costs, and to compare the results with our capacity model. Especially because CBB accepts a forecast deviation of +/- 10%, which increases the risk of overcapacity even more. We suggest introducing a cost penalty in the model to account for overcapacity, which could be defined by a probability of overcapacity given the order selection multiplied by the costs of overcapacity. Alternatively, a machine learning model such as RFR can provide a prediction interval of the number of orders to select. Hence, with a specified confidence level, we will be confident that the number of orders will fall within a range in terms of orders, and thus what the probability of overcapacity is in this case. This prediction interval can be utilized to make a trade-off between under- and overcapacity, because the capacity planning does not necessarily have to consist of the point forecast.
7. The optimal order selection depends on available time and capacity. For example, if CBB needs an extra biker for only one order, CBB needs to pay for the entire tour time instead of only the travel time. We excluded this information during the order selection because on

the operational level still can be decided not to select this one order and we did not want to make any assumptions about both order location and volume. It would require a priori knowledge about the capacity to deploy, or a lot of computational time to determine the number of tours, which is the desired output. However, it might be interesting to know the capacity optimum based on these constraints as well and compare the (robustness of the) results with the (robustness of the) current results. Hence, we recommend including these features in the order selection process, assigning locations randomly and order volume based on a probability function. The costs of adding an order are then determined on the number of tours instead of the travel time, e.g., with time and capacity feasibility checks. We suggest simulating this scenario multiple times to find statistically significant results.

8. In case one of the RFR models is utilized, we recommend the following:
   a. The data on which the models are trained, reflect the current situation at CBB. However, to ensure that the model is still effective under changing circumstances, we recommend updating the training data with the newest data (and forgetting or diminishing the oldest data when the dataset is large enough), and retraining the model with this new dataset. For more information on such an adaptive learning framework, we refer to Akkerman & Mes (2022). This framework can directly be applied to the model trained on historical data. To update the model trained on generated data, new data should be generated considering the changed circumstances. Note that this only makes an impact on the order selection decision if the fixed costs do not overshadow the variable costs.
   b. We recommend being aware of the cyclical effect of this solution methodology. The distance approximation influences the tactical order selection and thus the capacity deployed. The capacity deployed influences the actual order selection. This data is used for the distance approximation, and thus influences the number of orders selected based on the (performance of the) distance approximation.
9. Finally, we suggest researching the other two influenceable core problems as stated in Section 1.2.1. First, we observed that the forecast deviates from reality. In the current system, the capacity decision depends on the forecast. Consequently, Section 5.2.3.2 stressed the importance of an accurate forecast. At the moment, the forecast of CBB is a derivative of the Coolblue wide forecast. Hence, we recommend researching how to obtain specific CBB forecast to increase the forecast accuracy. Second, we noticed that the current route planning algorithm is sub-optimal. The orders selected to be delivered have a direct impact on the costs per order. Although there is a research project (cherry-picking) going on, we recommend improving this planning within a short timeframe. We suggest extending this tactical level capacity model, as explained in Section 6.3.1, with the operational level restrictions, such as the available time and capacity, to utilize it as an operational route planning tool. This is a quick improvement to the current situation.

7 References

Akkerman, F., & Mes, M. (2022). Distance approximation to support customer selection in vehicle routing problems. *Annals of Operations Research*.

Akkermans, F. R. (2021). *Delivery Cost Approximations for Dynamic Time Slot Pricing.* Enschede: University of Twente.

Amarim, P., Belo-Filho, M., Toledo, F., Almeder, C., & Almada-Lobo, B. (2013). Lot sizing versus batching in the production and distribution planning. *International Journal of Production Economics, 146*(1), 208-218.

Archetti, C., Grazia-Speranza, M., & Vigo, D. (2014). Vehicle Routing Problems with Profits. In D. Vigo, & P. Toth, *Vehicle Routing: Problems, Methods, and Applications* (pp. 273-298). Bologna: Society for Industrial and Applied Mathematics.

Arnold, F., & Sörensen, K. (2019). What makes a VRP solution good? The generation of problem-specific knowledge for heuristics. *Computers and Operations Research, 106*, 280-288.

Basel, J., & Willemain, T. (2001). Random tours in the traveling salesman problem: Analysis and application. *Computational Optimization and Applications, 20*(2), 211-217.

Beardwood, J., Halton, J., & Hammersley, J. (1959). The shortest path through many points. *Mathematical Proceedings of the Cambridge Philosophical Society, 55*(4), 299-327.

Berhan, E., Beshah, B., Kitaw, D., & Abraham, A. (2014). Stochastic Vehicle Routing Problem: A Literature Survey. *Journal of Information & Knowledge Management, 13*(3).

Bertazzi, L., Savelsbergh, M., & Grazia-Speranza, M. (2008). Inventory Routing. In B. Golden, S. Raghavan, & E. Wasil, *The Vehicle Routing Problem - Latest Advances and New Challenges* (pp. 49-72). New York: Springer.

Bodin, L., Golden, B., Assad, A., & Ball, M. (1983). Routing and scheduling of vehicles and crews: The state of art. *Computers & Operations Research, 10*(2), 63-211.

Botchkarev, A. (2019). A New Typology Design of Performance Metrics to Measure Errors in Machine Learning Regression Algorithms. *Interdisciplinary Journal of Information, Knowledge, and Management, 14*, 45-76.

Brahimi, N., Aouam, T., & Aghezzaf, E. (2015). Integrating order acceptance decisions with flexible due dates in a production planning model with load-dependent lead times. *International Journal of Production Research, 53*(12), 3810-3822.

Breiman, L. (2001). Random Forests. *Machine Learning, 45*, 5-32.

Bushuev, M. (2014). Convex optimization for aggregate production planning. *International Journal of Production Research, 52*(4), 1050-1058.

Caric, T., & Gold, H. (2008). *Vehicle Routing Problem.* Rijeka: IntechOpen.

Çavdar, B., & Sokol, J. (2015). A distribution-free TSP tour length estimation model for random graphs. *European Journal of Operational Research, 243*(2), 588-598.

Chen, C., Twycross, J., & Garibaldi, J. (2017). A new accuracy measure based on bounded relative error for time series forecasting. *PLOS ONE, 12*(3), e0174202.

Chien, T. (1992). Operational estimators for the length of a traveling salesman tour. *Computers & Operations Research, 19*(6), 469-478.

Chu, C. (2005). A heuristic algorithm for the truckload and less-than-truckload problem. *European Journal of Operational Research, 165*(3), 657-667.

Cleophas, C., & Ehmke, J. (2014). When Are Deliveries Profitable? Considering Order Value and Transport Capacity in Demand Fulfillment for Last-Mile Deliveries in Metropolitan Areas. *Business & Information Systems Engineering, 6*, 153-163.

Coolblue. (2022). *About Coolblue*. Retrieved from History: https://aboutcoolblue.com/en/history/

Couillard, J. (1993). A decision support system for vehicle fleet planning. *Decision Support Systems, 9*, 149-159.

Daganzo, C. (1987). The Distance Traveled to Visit N Points with a Maximum of C Stops per Vehicle: An Analytic Model and an Application. *Transportation Science, 21*(3), 331-350.

Dantzig, G., & Ramser, J. (1959). The Truck Dispatching Problem. *Management Science, 6*(1), 80-91.

De Boer, R. (1998). *Resource-Constrained Multi-Project Management: A hierarchical decision support.* Enschede: Febodruk.

Díaz-Madroñero, M., Peidro, D., & Mula, J. (2015). A review of tactical optimization models for integrated production and transport routing planning decisions. *Computers & Industrial Engineering, 88*(2015), 518-535.

Doan, T., Bostel, N., & Hà, M. (2021). The vehicle routing problem with relaxed priority rules. *EURO Journal on Transportation and Logistics, 10*, 100039.

Ehmke, J., & Campbell, A. (2014). Customer acceptance mechanisms for attended home deliveries in metropolitan areas. *European Journal of Operational Research, 223*(1), 193-207.

Ferando, J. (2021, September 12). *R-Squared Formula, Regression, and Interpretations*. Retrieved from Investopedia: https://www.investopedia.com/terms/r/r-squared.asp

Figliozzi, M. (2008). Planning Approximations to the Average Length of Vehicle Routing Problems with Varying Customer Demands and Routing Constraints. *Transportation Research Record, 2089*(1), 1-8.

Fisher, N., & Kordupleski, R. (2018). Good and bad market research: A critical review of Net Promoter Score. *Applied Stochastic Models in Business and Industry, 35*(1), 138-151.

Gendreau, M., Laporte, G., & Seguin, R. (1996). A Tabu Search Heuristic for the Vehicle Routing Problem with Stochastic Demands and Customers. *Operations Research, 44*(3), 469-477.

Geunes, J., Romeijn, H., & Taaffe, K. (2002). Models for Integrated Production Planning and Order Selection. *Proceedings of the 2002 Industrial Engineering Research Conference (IERC).* Orlando.

Gheysens, F., Golden, B., & Assad, A. (1986). A new heuristic for determining fleet size and composition. *Mathmatical Programming Study, 26*, 233-236.

Hans, E., Herroelen, W., Leus, R., & Wullink, G. (2005). *A hierarchical approach to multi-project planning uncer uncertainty.* Enschede: University of Twente.

Hans, E., Schutten, J., & Maan-Leeftink, A. (2021). *Planning & Scheduling - Lecture 1 & 2 [Powerpoint slides].* Retrieved from Canvas utwente.

Hayes, A. (2022, June 27). *Fixed Cost: What It Is and How It's Used in Business*. Retrieved from Investopedia: https://www.investopedia.com/terms/f/fixedcost.asp

Head, T., Kumar, M., Nahrstaedt, H., Louppe, G., & Shcherbatyi, I. (2021, 10 12). *Scikit-optimize/scikit-optimize.* Retrieved from Zenodo: https://doi.org/10.5281/zenodo.5565057

Heerkens, H., & van Winden, A. (2017). *Solving Managerial Problems Systematically.* Groningen: Noordhoff Uitgevers bv.

Heijungs, R. (2020). On the number of Monte Carlo run in comparative probabilistic LCA. *The International Journal of Life Cycle Assessment, 25*, 394-402.

Hernandez, F., Gendreau, M., & Potvin, J. (2017). Heuristics for tactical time slot management: a periodic vehicle routing problem view. *International Transactions in Operational Research, 24*(6), 1233-1252.

Hindle, A., & Worthington, D. (2004). Models to estimate average route lengths in different geographical environments. *The Journal of the Operational Research Society, 55*(6), 662-666.

Huang, L.-T., Hsieh, I.-C., & Farn, C.-K. (2011). On ordering adjustment policy under rolling forecast in supply chain planning. *Computers & Industrial Engineering, 60*(3), 397-410.

James, G., Witten, D., Hastie, T., & TIbshirani, R. (2021). *An Introduction to Statistical Learning.* New York: Springer.

Juan, A., Faulin, J., Grasman, S., Rabe, M., & Figureira, G. (2015). A Review of Simheuristics: Extending Metaheuristics to Deal With Stochastic Combinatorial Optimization Problems. *Operations Research Perspectives*.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., & Ye, Q. L. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. *Advances in Neural Information Processing Systems 30*, 3146-3154.

Kenton, W. (2022, August 11). *Monte Carlo Simulation: History, How it Works, and 4 Key Steps*. Retrieved from Investopedia: https://www.investopedia.com/terms/m/montecarlosimulation.asp

Kenton, W. (2022, August 17). *Variable Cost: What It Is and How to Calculate It*. Retrieved from Investopedia: https://www.investopedia.com/terms/v/variablecost.asp

Kim, B., Li, H., & Johnson, A. (2012). An augmented large neighborhood search method for solving the team orienting problem. *Expert Systems with Applications, 40*(8), 3065-3072.

Klein, R., Mackert, J., Neugebauer, M., & Steinhardt, C. (2018). A model-based approximation of opportunity cost for dynamic pricing in attended home delivery. *OR Spectrum, 40*(12), 969-996.

Konstantakopoulos, G., Gayialis, S., & Kechagias, E. (2020). Vehicle routing problem and related algorithms for logistics. *Operational Research*, 2033-2062.

Kou, S., Golden, B., & Poikonen, S. (2022). Optimal TSP tour length estimation using standard deviation as a predictor. *Computers and Operations Research, 148*, 105993.

Kou, S., Golden, B., & Poikonen, S. (2023). Estimating optimal objective values for the TSP, VRP, and other combinatorial problems using randomization. *International Transactions in Operational Research, 0*, 1-16.

Kwon, O., Golden, B., & Wasil, E. (1995). Estimating the Length of the Optimal TSP Tour: An Emperical Study using Regression and Neural Networks. *Computers Operational Research, 22*(10), 1039-1046.

Lagemann, H., & Meier, H. (2014). Robust capacity planning for the delivery of Industrial Product-Service Systems. *Robust Manufacturing Conference, 19*(2014), 99-104.

Lalla-Ruiz, E., & Voß, S. (2020). A POPMUSIC approach for the multi-depot cumulative capacitated vehicle. *Optimization Letters, 14*(3), 671-691.

Laurini. (2017). Geographic Relations. *Geographic Knowlegde Infrastructure*, 83-109.

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

Lawson, B., & Leemis, L. (2008). MONTE CARLO AND DISCRETE-EVENT SIMULATIONS IN C AND R. *Proceedings of the 2008 Winter Simulation Conference* (pp. 11-16). Miami: IEEE.

Lin, H.-T., & Chang, W.-L. (2008). Order selection and pricing methods using flexible quantity and fuzzy approach for buyer evaluation. *European Journal of Operational Research, 187*(2), 415-428.

List, G., Wood, B., Nozick, L., Turnquist, M., Jones, D., Kjeldgaard, E., & Lawton, C. (2003). Robust optimization for fleet planning under uncertainty. *Transportation Research, 39*, 209-227.

Mes, M., Schutten, J., & Perez-Rivera, A. (2014). Inventory routing for dynamic waste collection. *Waste Management, 34*(9), 1564-1576.

Microsoft Corporation. (2023). *LightGBM Parameters.* Retrieved from https://lightgbm.readthedocs.io/en/latest/

Mierzejewski, E. (1998). A Strategic Approach to the Transportation Planning Process. *Journal of Public Transportation, 2*(1), 27-42.

Nicola, D., Vetschera, R., & Dragomir, A. (2018). Total distance approximations for routing solutions. *Computers and Operations Research, 102*(2019), 67-74.

Olhager, J., Rudberg, M., & Wikner, J. (1999). Long-term capacity management: Linking the perspectives from manufacturing strategy and sales and operations planning. *Internal Journal of Production Economics, 69*(2001), 215-225.

Oord, T. (2022, May). *Cherry picking CoolblueBikes [Powerpoint slides].*

Pandala, S. R. (2022). *Lazy Predict Documentation*. Retrieved from https://lazypredict.readthedocs.io/en/latest/

Pilcher, M. (1990). Fleet Size Planning when Outside Carrier Services are Available. *Transportation Sciences, 24*(3), 169-182.

Rey, D., & Neuhäuser, M. (2011). Wilcoxon Signed-Rank Test. 1658-1659.

Robusté, F., Estrada, M., & López-Pita, A. (2004). Formulas for Estimating Average Distance. *Transportation Research Record*(1873), 64-69.

Roldán, R., Basagoiti, R., & Coelho, L. (2016). Robustness of inventory replenishment and customer selection policies for the dynamic and stochastic inventory-routing problem. *Computers & Operations Research, 74*, 14-20.

Rouberol, B. (2022, September 4). *haversine 2.7.0*. Retrieved from Pypi: https://pypi.org/project/haversine/

Scikit-learn. (2023). *RandomForestRegressor.* Retrieved from Scikit-learn: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

Scipy . (2023). *ConvexHull.* Retrieved from Scipy.spatial: https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.ConvexHull.html

Shyshou, A., ibkovskaia, I., & Barcelo, J. (2010). A simulation study of the fleet sizing problem arising in offshore anchor handling operations. *European Journal of Operational Research, 203*(1), 230-240.

Statistics. (2023). *Mathematical statistics functions.* Retrieved from Statistics: https://docs.python.org/3/library/statistics.html

SteadieSeifi, M., Dellaert, N., Nuijten, W., Van Woensel, T., & Raoufi, R. (2014). Multimodal Freight Transportation Planning: A Literature Review.

TIBC. (2023). *What is a Random Forest?* Retrieved from TIBC : https://www.tibco.com/reference-center/what-is-a-random-forest

Vidal, T. (2022). Hybrid genetic search for the CVRP: Open-source implementation and SWAP* neighborhood. *Computers and Operations Research, 140*, 105643.

Webb, M. (1968). Cost Functions in the Location of Depots for Multiple-delivery Journeys. *Operational Research Quarterly, 19*(3), 311-320.

Wright, M. N., & Ziegler, A. (2017). Ranger: A Fast Implementation of Random Forests for High Dimensional Data in C++ and R. *Journal of Statistical Software , 77*(March), 1-17.

Wu, J., Chen, X., Zhang, H., Xiong, L., Lei, H., & Deng, S. (2019). Hyperparameter Optimization for Machine Learning Models Based on Bayesian Optimization. *Journal of Electronic Science and Technology, 17*(1), 26-40.

Yang, X., Deb, S., & Fonf, S. (2013). Metaheuristic Algorithms: Optimal Balance of Intensification and Diversification. *Applied Mathematics & Informatin Sciences, 8*(3), 977-983.

Yu-Lee, R. T. (2002). *Essentials of Capacity Management.* New York: John Wiley & Sons.

8 Appendices

Appendix A Types of costs per order

This appendix defines and explains the types of costs per order. Also, it argues for the chosen cost values and how the costs are divided over the hubs.

Fixed costs

Housing costs

The housing costs are the costs of the hubs. CBB has three types of hubs. A hub can be in a Coolblue store, it can be attached to a CoolblueDelivers depot or it can be a stand-alone hub. Each format has different associated costs. We use the average housing costs per hub of 2022 in our analysis.

Office salary costs

CBB has office employees to support the CBB delivery network. They monitor and improve the performance of CBB, but they have no direct added value. These salary costs are fixed since an in- or decrease in orders does not cause an (un)employment of office support.

The office salary costs should be divided over all hubs. Roughly half the employees equally distribute their time over all the hubs, i.e., process engineers implementing innovation projects. These projects should be implemented on every hub, independent of the size of the hub. The other half of the employees spend their time proportional to the number of potential orders per hub, i.e., human resource employees. The more orders the hub is going to deliver, the more delivery men the hubs needs and thus the more work for human resource employees. Hence, we use the most representative salary costs, from December 2022, and divide half of the costs equally over the hubs and half of the costs proportional to the potential orders per hub.

Hub lead costs

Every hub has a lead or manager leading the operation. Their salary is fixed and we use the most representative salary costs per hub, from December 2022, in our analysis.

Linehaul costs

The line haul costs are the costs to transport the orders from the warehouse to the hubs. CBB transports these orders with a box truck or trailer. The costs consist of the travel cost per kilometre and the driver cost per hour. Furthermore, the costs depend on the number and the size of the orders transported, because this influences the total number of trucks and trailers necessary for transport. Every day delivery is necessary to ensure next day delivery.

However, the number of orders fitting in a box truck or trailer is sufficient to cover the highest number of orders per day to be transported. Based on the expected value of the order volume (4298 cm^3) of empirical probability distribution function based on historically observed orders in Hilversum (see Section 4.3.3.2), and the inner volume (964592 cm^3) of a roller container, we observe roughly 224 orders per roller container. A box truck carries 45 roller containers and a trailer 10, so around 10080 or 2243 orders. With a maximum of 460 observed potential orders per day, we can safely assume that the line haul costs are fixed for simplification purposes, although officially they are variable costs. We use the average line haul costs of 2022 per hub in our analysis.

Other operating expenses

Other operating expenses are several fixed expenses to keep the hubs from CBB up and running. An example is a hub's computer expenses. The use of the other operating expenses per hub is proportional to the size of the hub, so we use the average other operating expenses of 2022 proportional to the potential orders per hub.

Variable costs

Biker salary costs
The biker salary is the salary of the delivery men of CBB. The more days the delivery men work, the higher their salaries. The delivery men are planned for a morning, an afternoon, an evening, a morning and afternoon or an afternoon and evening. In each day part, maximal one bike route is driven per delivery man. The salary per day part is fixed, so the more orders per hour are delivered by the delivery men, the less biker salary costs are passed on per order.

We use the hourly salary costs of the most representative month, December 2022. Additional to the hours spent on the bike, the delivery men are also paid for the hours spent on the hub and the shrinkage hours (see Section 2.1.2). Furthermore, we cannot assume that all orders are delivered in the first try, because customers (or neighbours) might not be at home. Hence, we correct the hourly salary costs value such that it also accounts for the hours spent on the bike (80% of the working time), the hours spent on the hub (20% of the working time), the shrinkage hours (see Section 2.1.2) (additional 8% for holidays and 7% for absence on top of the working time), and the percentage not at home customers (2% of all the orders). CBB determined these percentages on historical data and uses those for their analyses. Hence, to be in line with their working methods, we use the same percentages.

Vehicle maintenance costs
The vehicle costs are the maintenance costs of the bikes. This includes preventive maintenance and ad hoc repairs. This maintenance occurs regularly, so the more orders per hour are delivered by the vehicles in this timespan, the fewer maintenance costs are passed on per order. The storage costs of the bikes are included in the housing costs because all depots have sufficient space to store the bikes. We use the average price per kilometre of 2022 that Coolblue paid for the bikes in our analysis.

Fraud/theft costs
When fraud and theft occur, CBB pays for these costs. Prevention of fraud and theft does not relate to capacity planning, so we decided together with Coolblue that we cannot influence these costs. We use the average fraud/theft costs per order of 2022 in our analysis.

Recruitment costs
The recruitment costs are the costs to recruit new delivery men. These costs fluctuate based on the recruitment targets. If CBB needs a lot of new delivery men to deliver (more) orders, they have more costs than if they have sufficient delivery men. Therefore, we assume a linear relation and use the average recruitment costs per order of 2022 in our analysis.

Allocated customer service costs
The allocated customer service costs are the customer service costs from the CBB customers. Coolblue has one general customer service and allocates the costs to the responsible department. These costs are dependent on the general service provided, for example, if the delivery contains everything ordered or if the order arrived on time. We decided together with Coolblue that we cannot influence these costs with capacity planning. Hence, we use the average allocated customer service costs per order of 2022 in our analysis.

Allocated costs process returns
The allocated costs process return are the costs related to returning to the customer. This includes for example the costs to return to a customer when CBB could not deliver at the initial delivery date. Delivery is not possible when the customer is not at home and the neighbours neither. We decided together with Coolblue that we cannot influence these costs with capacity planning. We use the average costs process returns per order of 2022 in our analysis.

Other staffing costs
Other staffing costs are clothing, coffee and tea costs from the staff. These costs increase when the number of staff increases and can be seen as additional biker salary costs. Therefore, the more orders are delivered per hour, the less staffing costs are passed on per order. We use the other staffing costs per hour of the most representative month, December 2022.

Appendix B Feature orders selected per PC4

This appendix explains the inclusion of the feature number of orders selected per PC4 in the machine learning model. We include the feature per PC4 in the RFR model. We test and compare the predictive performance of the RFR model on the hub in Hilversum in line with Section 5.1, because the RFR model turned out to have the best predictive performance. This means 45 extra features, one for every PC4 in Hilversum.

Table 16 displays the out-of-sample performance metrics of the RFR model with the 45 extra features on the historical data, and the RFR models trained historical data (2) to compare with. Likewise, Table 17 shows the in-sample performance metrics. We observe a worse predictive performance from the RFR with extra features on all performance metrics. Furthermore, the RFR model with extra features shows a high R^2 on the training set and a way lower R^2 on the test set. The difference is even bigger with the adjusted R^2, which is adjusted for the number of features of the model. This indicates that the model is overfitted. Hence, the inclusion of the 45 extra features decreases the general predictive performance, the accuracy, and creates noise, which is an unrelated variation from the relation of the features with the response. Additionally, including 45 extra features make the model more complex.

Table 16 Performance metrics on the test data set of RFR on Hilversum

Hilversum Test Data	RFR extra features (2)	RFR (2)
# Features	51	6
rRMSE	11.25 %	10.44 %
rMAE	8.35 %	8.03 %
R^2	0.838	0.887
Adjusted R^2	0.754	0.881
MAPE	8.77 %	8.67 %

Table 17 Performance metrics on the train data set of RFR on Hilversum

Hilversum Train Data	RFR extra features (2)	RFR (2)
Features	51	6
rRMSE	4.31 %	3.58 %
rMAE	3.29 %	2.70 %
R^2	0.977	0.983
Adjusted R^2	0.975	0.983
MAPE	3.97 %	2.98 %

Furthermore, we assess the feature importance of the RFR model with extra features. Figure 47 shows the importance of each feature, where the individual PC4s are the 4 numbers on the x-axis. They represent the number of orders selected per PC4. A higher importance score indicates that the feature has a larger effect on the predictive performance of the model. Indeed, we observe that the 45 features are not extremely important. Hence, we decide to exclude these 45 features from the model.

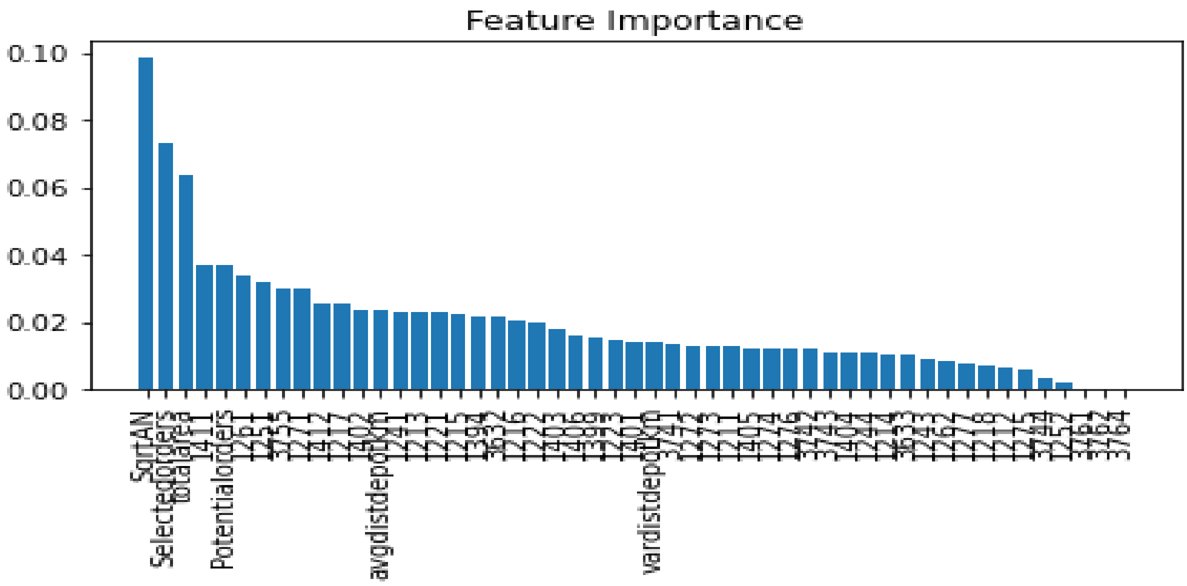

Figure 47 Feature importance RFR with extra features on Hilversum

Appendix C Predictive performance grouped data on all hubs

This appendix shows the predictive performance of Method RFR (2) trained on historically grouped data. We display the number of features, the tuned hyperparameters and the predictive performance of the small hubs RFR (2) model and the large hubs RFR (2) model in general and on each of the small and large hubs. The performance metrics are evaluated on unseen test data. Table 18 and Table 19 show the performance metrics of the model trained on the historical data of all the small hubs and Table 20 of the model trained on the historical data of all the large hubs. The general small hubs and large hubs models perform usually the best compared to the application of the models on an individual small or large hub. However, the hub in Eindhoven, Zwijndrecht and Rotterdam outperform the general method. Striking is that the hyperparameters are tuned to the same value. Generally speaking, the aggregated models have a more than sufficient predictive performance for each individual hub. This implies that if the RFR model trained grouped data works on one hub, it most likely works on all hubs.

Table 18 Predictive performance of the RFR model on small hubs (1/2)

Small hubs	**Small hubs**	**Hilversum**	**Breda**	**Deventer**	**Eindhoven**
# features	6	6	6	6	6
Max. number of features for splitting	2	2	2	2	2
Min. number of data points per leaf node	1	1	1	1	1
rRMSE	11.50 %	10.46 %	9.74 %	15.11 %	10.98 %
rMAE	8.50 %	8.64 %	7.87 %	11.41 %	7.91 %
Adjusted R^2	0.949	0.873	0.909	0.905	0.958
MAPE	9.98 %	9.78 %	8.40 %	14.47 %	9.32 %

Table 19 Predictive performance of the RFR model on small hubs (2/2)

Small hubs	**Groningen**	**Haarlem**	**Leiderdorp**	**Tilburg**	**Veenendaal**	**Zwijndrecht**
# features	6	6	6	6	6	6
Max. number of features for splitting	2	2	2	2	2	2
Min. number of data points per leaf node	1	1	1	1	1	1
rRMSE	15.04 %	11.03 %	12.20 %	11.19 %	15.00 %	7.05 %
rMAE	9.94 %	8.95 %	7.74 %	8.44 %	12.29 %	4.80 %
Adjusted R^2	0.912	0.950	0.962	0.937	0.773	0.963
MAPE	10.42 %	12.48 %	7.72 %	9.78 %	13.96 %	5.06 %

Table 20 Predictive performance of the RFR model on large hubs

Large hubs	**Large hubs**	**Den Haag**	**Utrecht**	**Amsterdam**	**Rotterdam**
# features	6	6	6	6	6
Max. number of features for splitting	3	3	3	3	3
Min. number of data points per leaf node	1	1	1	1	1
rRMSE	9.20 %	8.34 %	9.42 %	9.47 %	8.87 %
rMAE	6.64 %	6.33 %	7.04 %	6.39 %	6.38 %
Adjusted R^2	0.975	0.972	0.960	0.982	0.977
MAPE	8.14 %	7.47%	7.75 %	8.66 %	7.99 %

Appendix D Estimation of the PC4 order distributions

Table 21 summarizes per PC4 the Chi-square p-values and the estimated parameters of the negative binomial distribution. Table 22 shows the Chi-square p-values that indicate the goodness of fit of the Poisson, binomial, geometric and negative binomial distributions, only for the PC4s for which we did not observe significant evidence to state that there is distribution is negative binomial.

Table 21 PC4 p-value and estimated parameters negative binomial distribution

PC4	P-value	Mu	Size	P
1211	0.9419	5.245651	10.17014	0.659722
1213	0.9786	5.951209	10.39119	0.635842
1214	0.9786	3.683974	5.695433	0.607227
1215	0.7856	5.671792	7.142007	0.557368
1216	0.7738	4.433641	8.118164	0.646773
1217	0.9594	7.083798	7.73096	0.521842
1218	0.5376	1.889235	40.37072	0.955295
1221	0.8881	3.94815	6.305357	0.614946
1222	0.8524	4.620472	7.274529	0.611562
1223	0.5338	5.631133	8.810972	0.610089
1241	0.8746	3.141305	10.98077	0.777561
1251	0.9973	6.572892	8.160366	0.553874
1261	0.8595	6.231571	7.733758	0.553783
1271	0.9866	5.098274	7.218504	0.586071
1272	0.7319	4.00169	9.647113	0.70681
1273	0.8762	3.702005	9.713948	0.724059
1274	0.6277	2.880728	11.7307	0.802844
1276	0.3409	2.788455	11.11355	0.799421
1394	0.7902	2.96323	8.951957	0.751306
1399	0.07361	1.86169	11480.98	0.999838
1401	0.289	3.438727	9.118024	0.726145
1402	0.7258	5.115522	7.615918	0.598198
1403	0.7849	3.657508	8.634452	0.702447
1404	0.2514	2.223278	53.17654	0.959868
1405	0.2233	2.267656	17.0156	0.882403
1406	0.3426	2.260786	11.2495	0.832662
1411	0.9949	6.573633	9.499253	0.591011
1412	0.7998	4.957732	7.822675	0.612083
3632	0.7722	2.613093	13.64324	0.839257
3741	0.9238	4.78625	11.35863	0.703544
3742	0.8438	4.057835	8.188722	0.668655
3743	0.7216	3.290725	13.49486	0.803955
3755	0.9604	5.041061	10.22055	0.66969
3761	0.9319	2.400232	26.37429	0.916585
3762	0.8829	3.928855	3.541119	0.474047
3764	0.9873	1.999756	11.82691	0.85537

Table 22 P-values of the Chi-square test of 4 discrete distributions

PC4	Poisson p-value	Binomial p-value	Geometric p-value	Negative Binomial p-value
1212	0.005229	3.33E-04	1.83E-43	0.004473
1243	7.09E-12	2.43E-06	5.68E-55	0.004472585
1244	6.332332e-08	1.43E-04	4.06E-51	0.004472585
1252	4.820363e-10	4.820363e-10	4.820363e-10	4.820363e-10
1262	3.922460e-06	3.922460e-06	3.922460e-06	3.922460e-06
1275	5.106583e-09	5.106583e-09	5.106583e-09	5.106583e-09
1277	4.069923e-04	4.069923e-04	4.069923e-04	4.069923e-04
3633	1.417649e-06	1.417649e-06	1.417649e-06	1.417649e-06
3744	8.512302e-11	8.512302e-11	8.512302e-11	8.512302e-11

Appendix E Additional performance metrics of the conducted experiments

In this appendix, we report the median and interquartile range of the distance, travel time, number of routes, number of postal codes and the orders deliver per hour belonging to the experiments, replicated 10 times, conducted in Section 5.2.2. We show the metrics per order selection method. The metrics are retrieved by selecting orders with the respective order selection method and solving the CVRP with the solver in Python, such that we can compare and validate the methods mutually. Table 23 and Table 24 show the performance metrics of the order selection Method 'Optimal levels', Table 25 and Table 26 of the Method Daganzo, Table 27 and Table 28 of the Method RFR (1), and Table 29 and Table 30 of Method RFR (2).

We define the metrics as follows. The distance is the total daily distance of all bike routes in kilometres. The travel time is the total daily travel time of all bike routes in minutes, which includes the stop time at an order. The number of routes is the number of bike routes necessary to deliver the selected orders. Note that this number is not necessarily the same as the number of bikes per day, since bikes can be reused in the morning, afternoon and evening. The orders per hour is the number of orders delivered per hour and indicates the route efficiency of the order selection. Here, more orders per hour mean a higher route efficiency. The number of PC4s is the number of PC4s in which we deliver orders. Finally, the necessary time in minutes is the total time we need the delivery men, and thus the delivery men minutes. It accounts for the hours spent on the bike (80% of the working time), the hours spend on the hub (20% of the working time), the shrinkage hours (see Section 2.1.2) (an additional 8% for holidays and 7% for absence on top of the working time), and the percentage not at home customers (2% of all the orders). We refer to Appendix A for more information about the necessary time.

Method 'Optimal levels'

Table 23 Performance metrics Method 'Optimal levels' (1/2)

# potential orders	Distance (km)		Travel time (min.)		# routes	
	Q2	Q1-Q3	Q2	Q1-Q3	Q2	Q1-Q3
20	49.56	49.56 - 52.86	197.9	197.9 - 214.03	1	1 - 2
40	70.2	70.2 - 73.9	310.51	310.51 - 321.17	2	2 - 2
60	74.87	74.87 - 79.35	366.01	366.01 - 378.92	2	2 - 2
80	89.39	89.39 - 91.63	450.66	450.66 - 460.36	3	3 - 3
100	97.95	97.95 - 99.05	518.7	518.7 - 527.87	3	3 - 3
120	105.16	105.16 - 105.93	593.89	593.89 - 595.34	4	4 - 4
140	111.42	111.42 - 118.72	653.71	653.71 - 675.67	4	4 - 4
160	117.2	117.2 - 121.91	718.01	718.01 - 723.76	5	5 - 5
180	128.49	128.49 - 134.25	790.41	790.41 - 820.17	5	5 - 5
200	139.16	139.16 - 147.93	861.8	861.8 - 886.25	6	6 - 6
220	147.81	147.81 - 149.42	937.16	937.16 - 963.11	6	6 - 6
240	157.86	157.86 - 158.34	1019.41	1019.41 - 1030.21	7	7 - 7
260	163.4	163.4 - 170.57	1083	1083 - 1084.13	7	7 - 7
280	171.69	171.69 - 182.51	1148.58	1148.58 - 1157.5	8	8 - 8
300	179.72	179.72 - 183.21	1205.57	1205.57 - 1226.59	8	8 - 8
320	194.95	194.95 - 201.54	1301.11	1301.11 - 1316.39	9	9 - 9
340	202.42	202.42 - 214.95	1370.8	1370.8 - 1412.2	9	9 - 10
360	207.12	207.12 - 215.36	1401.59	1401.59 - 1440.32	10	10 - 10
380	226.77	226.77 - 234.57	1525.78	1525.78 - 1554.59	11	11 - 11
400	226.27	226.27 - 236.02	1589.91	1589.91 - 1603.39	11	11 - 11
420	230.14	230.14 - 237.05	1631.6	1631.6 - 1653.75	11	11 - 12

440	241.46	241.46 - 251.49	1715.31	1715.31 - 1742.62	12	12 - 12
460	251.25	251.25 - 258.16	1799.34	1799.34 - 1816.57	12	12 - 13
480	250.82	250.82 - 251.72	1820.82	1820.82 - 1832.32	12	12 - 12
500	270.63	270.63 - 278.15	1914.13	1914.13 - 1936.47	14	14 - 14
520	281.8	281.8 - 284.35	2016.12	2016.12 - 2022.86	14	14 - 14
540	288.56	288.56 - 300.62	2075.72	2075.72 - 2091.24	14	14 - 15
560	290.79	290.79 - 299.3	2111.4	2111.4 - 2166.62	15	15 - 15
580	301.64	301.64 - 311.6	2188.57	2188.57 - 2234.97	15	15 - 16
600	309.49	309.49 - 319.96	2252	2252 - 2287.33	16	16 - 16
620	317.8	317.8 - 336.96	2298.95	2298.95 - 2429	17	17 - 17
640	335.47	335.47 - 344.72	2435.12	2435.12 - 2464.8	17	17 - 18
660	335.27	335.27 - 349.86	2492.12	2492.12 - 2527.47	18	18 - 18
680	339.26	339.26 - 348.76	2538.64	2538.64 - 2562.77	18	18 - 19

Table 24 Performance metrics Method 'Optimal levels' (2/2)

# potential orders	**Orders per hour**		**# PC4s**		**Necessary time (min)**	
	Q2	**Q1-Q3**	**Q2**	**Q1-Q3**	**Q2**	**Q1-Q3**
20	5.48	5.48 - 5.89	17	17 - 19	288.86	288.86 - 312.41
40	7.45	7.45 - 7.59	27	27 - 28	453.23	453.23 - 468.78
60	9.17	9.17 - 9.34	30	30 - 31	534.24	534.24 - 553.08
80	9.74	9.74 - 10	35	35 - 36	657.8	657.8 - 671.95
100	10.88	10.88 - 10.9	37	37 - 37	757.1	757.1 - 770.49
120	11.56	11.56 - 11.65	37	37 - 38	866.85	866.85 - 868.97
140	12.12	12.12 - 12.18	37	37 - 38	954.16	954.16 - 986.22
160	12.61	12.61 - 12.77	36	36 - 37	1048.02	1048.02 - 1056.42
180	12.89	12.89 - 13.13	38	38 - 38	1153.7	1153.7 - 1197.13
200	13	13 - 13.35	37	37 - 37	1257.9	1257.9 - 1293.59
220	13.28	13.28 - 13.39	37	37 - 38	1367.9	1367.9 - 1405.78
240	13.41	13.41 - 13.56	38	38 - 38	1487.95	1487.95 - 1503.71
260	13.63	13.63 - 14.04	37	37 - 38	1580.77	1580.77 - 1582.42
280	13.94	13.94 - 14.01	38	38 - 38	1676.48	1676.48 - 1689.52
300	14.13	14.13 - 14.3	37	37 - 38	1759.68	1759.68 - 1790.36
320	13.89	13.89 - 14.45	37	37 - 39	1899.12	1899.12 - 1921.43
340	13.92	13.92 - 14.13	37	37 - 38	2000.84	2000.84 - 2061.27
360	14.3	14.3 - 14.53	36	36 - 37	2045.79	2045.79 - 2102.32
380	14.08	14.08 - 14.38	38	38 - 38	2227.05	2227.05 - 2269.1
400	14.4	14.4 - 14.6	37	37 - 37	2320.67	2320.67 - 2340.34
420	14.63	14.63 - 14.8	37	37 - 38	2381.51	2381.51 - 2413.85
440	14.46	14.46 - 14.59	37	37 - 38	2503.7	2503.7 - 2543.57
460	14.61	14.61 - 14.69	36	36 - 37	2626.35	2626.35 - 2651.5
480	14.88	14.88 - 14.99	36	36 - 37	2657.71	2657.71 - 2674.49
500	14.55	14.55 - 14.84	37	37 - 39	2793.9	2793.9 - 2826.51
520	14.8	14.8 - 14.82	37	37 - 38	2942.77	2942.77 - 2952.61
540	14.71	14.71 - 14.85	37	37 - 38	3029.76	3029.76 - 3052.42
560	14.95	14.95 - 14.99	37	37 - 38	3081.84	3081.84 - 3162.45
580	15.02	15.02 - 15.2	37	37 - 38	3194.48	3194.48 - 3262.21

600	14.88	14.88 - 15.18	38	38 - 38	3287.06	3287.06 - 3338.64
620	14.94	14.94 - 15.14	37	37 - 38	3355.6	3355.6 - 3545.41
640	14.92	14.92 - 15.05	38	38 - 38	3554.36	3554.36 - 3597.67
660	15.01	15.01 - 15.25	37	37 - 38	3637.55	3637.55 - 3689.15
680	15.15	15.15 - 15.29	38	38 - 38	3705.45	3705.45 - 3740.67

Method Daganzo

Table 25 Performance metrics Method Daganzo (1/2)

# potential orders	Distance (km)		Travel time (min.)		# routes	
	Q2	Q1-Q3	Q2	Q1-Q3	Q2	Q1-Q3
20	60.13	60.13 - 68.55	236.11	236.11 - 262.16	2	2 - 2
40	80.71	80.71 - 84.39	347.11	347.11 - 357.88	2	2 - 2
60	92.98	92.98 - 94.11	428.96	428.96 - 434.52	3	3 - 3
80	105.26	105.26 - 108.32	511.89	511.89 - 522.84	3	3 - 3
100	113.16	113.16 - 114.52	582.71	582.71 - 585.09	3	3 - 3
120	123.05	123.05 - 126.97	660.92	660.92 - 672.06	4	4 - 4
140	131.29	131.29 - 133.92	731.11	731.11 - 738.4	4	4 - 5
160	135.88	135.88 - 136.36	784.7	784.7 - 794.68	5	5 - 5
180	150.1	150.1 - 150.24	872.68	872.68 - 882.14	5	5 - 5
200	156.64	156.64 - 160.18	932.46	932.46 - 961.35	6	6 - 6
220	167.23	167.23 - 169.01	1023.78	1023.78 - 1035.02	6	6 - 6
240	174.91	174.91 - 178.07	1083.91	1083.91 - 1085.18	7	7 - 7
260	178.31	178.31 - 184.89	1128.28	1128.28 - 1158.48	8	8 - 8
280	191.29	191.29 - 195.17	1223.98	1223.98 - 1244.52	8	8 - 8
300	194.42	194.42 - 202.66	1269.9	1269.9 - 1302.37	8	8 - 9
320	210.05	210.05 - 215.15	1370.55	1370.55 - 1383	9	9 - 9
340	221.1	221.1 - 228.69	1457.81	1457.81 - 1467.42	10	10 - 10
360	235.94	235.94 - 241.06	1547.9	1547.9 - 1560.03	10	10 - 11
380	247.53	247.53 - 251.94	1617.23	1617.23 - 1647.71	11	11 - 11
400	247.76	247.76 - 255.69	1671.71	1671.71 - 1695.6	11	11 - 12
420	258.94	258.94 - 268.86	1751.45	1751.45 - 1791.59	12	12 - 12
440	264.3	264.3 - 270.15	1806.41	1806.41 - 1839.43	12	12 - 13
460	272.58	272.58 - 278.59	1884.85	1884.85 - 1916.57	12	12 - 13
480	269.71	269.71 - 273.21	1900.85	1900.85 - 1911.61	12	12 - 13
500	289.03	289.03 - 307.41	2000.81	2000.81 - 2040.8	14	14 - 15
520	286.65	286.65 - 310.32	2042.33	2042.33 - 2144.43	14	14 - 14
540	313.09	313.09 - 322.55	2140.68	2140.68 - 2215.94	15	15 - 16
560	314.24	314.24 - 324.48	2213.38	2213.38 - 2277.47	15	15 - 16
580	323.97	323.97 - 334.53	2279.44	2279.44 - 2339.1	15	15 - 17
600	330.4	330.4 - 339.39	2369.69	2369.69 - 2378.39	16	16 - 16
620	358.34	358.34 - 363.04	2510.21	2510.21 - 2529.25	18	18 - 18
640	351.05	351.05 - 355.52	2526.75	2526.75 - 2559.51	18	18 - 18
660	349.13	349.13 - 362.72	2582.05	2582.05 - 2602.92	18	18 - 18
680	357.86	357.86 - 367.44	2613.99	2613.99 - 2657.41	18	18 - 19

Table 26 Performance metrics Method Daganzo (2/2)

# potential orders	Orders per hour		# PC4s		Necessary time (min)	
	Q2	Q1-Q3	Q2	Q2	Q2	Q2
20	5.08	5.08 - 5.61	19	19 - 20	344.63	344.63 - 382.66
40	6.91	6.91 - 7.17	29	29 - 30	506.65	506.65 - 522.37
60	8.28	8.28 - 8.47	33	33 - 33	626.12	626.12 - 634.24
80	9.27	9.27 - 9.38	39	39 - 40	747.16	747.16 - 763.15
100	10.25	10.25 - 10.47	40	40 - 41	850.53	850.53 - 854.01
120	10.89	10.89 - 10.96	39	39 - 42	964.69	964.69 - 980.95
140	11.49	11.49 - 11.49	40	40 - 42	1067.15	1067.15 - 1077.78
160	12.08	12.08 - 12.13	40	40 - 40	1145.36	1145.36 - 1159.92
180	12.28	12.28 - 12.41	42	42 - 42	1273.78	1273.78 - 1287.59
200	12.49	12.49 - 12.85	40	40 - 40	1361.04	1361.04 - 1403.2
220	12.87	12.87 - 12.89	41	41 - 41	1494.33	1494.33 - 1510.73
240	12.96	12.96 - 13.22	41	41 - 42	1582.1	1582.1 - 1583.95
260	13.23	13.23 - 13.47	39	39 - 41	1646.86	1646.86 - 1690.94
280	13.5	13.5 - 13.57	41	41 - 43	1786.55	1786.55 - 1816.52
300	13.84	13.84 - 13.92	41	41 - 41	1853.56	1853.56 - 1900.96
320	13.57	13.57 - 14.04	41	41 - 42	2000.48	2000.48 - 2018.65
340	13.54	13.54 - 14.03	40	40 - 42	2127.85	2127.85 - 2141.88
360	13.95	13.95 - 14.19	40	40 - 41	2259.35	2259.35 - 2277.05
380	13.75	13.75 - 14.06	41	41 - 41	2360.53	2360.53 - 2405.03
400	14.15	14.15 - 14.33	40	40 - 42	2440.06	2440.06 - 2474.94
420	14.15	14.15 - 14.54	39	39 - 40	2556.45	2556.45 - 2615.04
440	14.25	14.25 - 14.36	40	40 - 41	2636.68	2636.68 - 2684.86
460	14.4	14.4 - 14.62	38	38 - 40	2751.17	2751.17 - 2797.47
480	14.63	14.63 - 14.71	38	38 - 39	2774.52	2774.52 - 2790.22
500	14.48	14.48 - 14.64	40	40 - 42	2920.42	2920.42 - 2978.79
520	14.55	14.55 - 14.67	39	39 - 40	2981.03	2981.03 - 3130.05
540	14.41	14.41 - 14.79	40	40 - 42	3124.59	3124.59 - 3234.43
560	14.75	14.75 - 14.81	39	39 - 42	3230.7	3230.7 - 3324.24
580	14.76	14.76 - 14.92	40	40 - 41	3327.11	3327.11 - 3414.2
600	14.76	14.76 - 15.08	40	40 - 41	3458.85	3458.85 - 3471.55
620	14.63	14.63 - 14.76	41	41 - 42	3663.95	3663.95 - 3691.75
640	14.7	14.7 - 14.9	40	40 - 41	3688.1	3688.1 - 3735.91
660	14.94	14.94 - 15.06	38	38 - 41	3768.82	3768.82 - 3799.27
680	14.85	14.85 - 15.15	39	39 - 39	3815.43	3815.43 - 3878.8

Method RFR (1)

Table 27 Performance metrics Method RFR (1) (1/2)

# potential orders	Distance (km)		Travel time (min.)		# routes	
	Q2	Q1-Q3	Q2	Q1-Q3	Q2	Q1-Q3
20	60.13	60.13 - 68.55	236.11	236.11 - 262.16	2	2 - 2
40	80.71	80.71 - 84.39	347.11	347.11 - 357.88	2	2 - 2
60	92.98	92.98 - 95.27	428.96	428.96 - 434.52	3	3 - 3

80	105.18	105.18 - 108.32	508.78	508.78 - 522.01	3	3 - 3
100	111.55	111.55 - 114.52	573.21	573.21 - 585.09	3	3 - 3
120	123.05	123.05 - 124.08	653.45	653.45 - 660.51	4	4 - 4
140	130.94	130.94 - 132.29	723.41	723.41 - 731.32	4	4 - 5
160	132.44	132.44 - 134.61	770.87	770.87 - 775.96	5	5 - 5
180	150.24	150.24 - 151.43	877.25	877.25 - 881.54	5	5 - 5
200	156.93	156.93 - 159.37	933.38	933.38 - 952.29	6	6 - 6
220	158.61	158.61 - 160.82	981.76	981.76 - 997.55	6	6 - 6
240	176.99	176.99 - 179.38	1080.9	1080.9 - 1105.85	7	7 - 7
260	175.84	175.84 - 184.85	1120.56	1120.56 - 1150.58	7	7 - 8
280	189.47	189.47 - 197.22	1225.19	1225.19 - 1252.51	8	8 - 8
300	194.85	194.85 - 198.65	1266.89	1266.89 - 1294.45	8	8 - 8
320	214.21	214.21 - 219.61	1382.36	1382.36 - 1408.89	9	9 - 10
340	221.38	221.38 - 229.95	1450.4	1450.4 - 1466.13	9	9 - 10
360	224.81	224.81 - 239.77	1485.83	1485.83 - 1553.97	10	10 - 11
380	244.68	244.68 - 247.14	1567.87	1567.87 - 1631.37	11	11 - 11
400	241.86	241.86 - 260.24	1639.5	1639.5 - 1699.04	11	11 - 12
420	245.18	245.18 - 264.06	1673.67	1673.67 - 1777.11	11	11 - 13
440	259.31	259.31 - 271.21	1776.27	1776.27 - 1842.03	12	12 - 12
460	263.09	263.09 - 278.38	1859.62	1859.62 - 1911.78	12	12 - 13
480	270.76	270.76 - 273.38	1863.26	1863.26 - 1945.64	12	12 - 13
500	296.9	296.9 - 305.27	2039.5	2039.5 - 2067.26	14	14 - 15
520	303.29	303.29 - 310.67	2123.61	2123.61 - 2137.73	14	14 - 14
540	311.02	311.02 - 330.51	2160.05	2160.05 - 2246.25	15	15 - 16
560	321.88	321.88 - 327.46	2264.93	2264.93 - 2277.85	16	16 - 16
580	336.76	336.76 - 350.25	2364.75	2364.75 - 2402.69	16	16 - 17
600	333.26	333.26 - 338.32	2372.79	2372.79 - 2398.6	16	16 - 16
620	359.39	359.39 - 366.47	2528.84	2528.84 - 2537.11	18	18 - 18
640	368.44	368.44 - 377.66	2592.27	2592.27 - 2623.23	18	18 - 18
660	365.71	365.71 - 375.79	2637.61	2637.61 - 2661.48	18	18 - 19
680	371.85	371.85 - 383.7	2697.85	2697.85 - 2752.61	19	19 - 20

Table 28 Performance metrics Method RFR (2) (2/2)

# potential orders	**Orders per hour**		**# PC4s**		**Necessary time (min)**	
	Q2	**Q1-Q3**	**Q2**	**Q1-Q3**	**Q2**	**Q1-Q3**
20	5.08	5.08 - 5.61	19	19 - 20	344.63	344.63 - 382.66
40	6.91	6.91 - 7.04	29	29 - 30	506.65	506.65 - 522.37
60	8.28	8.28 - 8.51	32	32 - 33	626.12	626.12 - 634.24
80	9.2	9.2 - 9.34	38	38 - 39	742.63	742.63 - 761.94
100	10.25	10.25 - 10.44	39	39 - 41	836.67	836.67 - 854.01
120	10.84	10.84 - 10.9	38	38 - 40	953.79	953.79 - 964.09
140	11.46	11.46 - 11.5	40	40 - 41	1055.91	1055.91 - 1067.45
160	11.94	11.94 - 12.15	38	38 - 39	1125.18	1125.18 - 1132.61
180	12.29	12.29 - 12.69	42	42 - 43	1280.46	1280.46 - 1286.72
200	12.54	12.54 - 12.82	39	39 - 41	1362.39	1362.39 - 1389.98
220	12.8	12.8 - 12.9	39	39 - 40	1433	1433 - 1456.04

240	13.02	13.02 - 13.15	40	40 - 42	1577.71	1577.71 - 1614.12
260	13.24	13.24 - 13.45	38	38 - 42	1635.59	1635.59 - 1679.41
280	13.46	13.46 - 13.61	40	40 - 43	1788.31	1788.31 - 1828.19
300	13.7	13.7 - 13.91	40	40 - 41	1849.18	1849.18 - 1889.4
320	13.63	13.63 - 13.9	41	41 - 42	2017.73	2017.73 - 2056.45
340	13.46	13.46 - 14.07	40	40 - 41	2117.03	2117.03 - 2140
360	13.97	13.97 - 14.19	39	39 - 41	2168.75	2168.75 - 2268.21
380	13.74	13.74 - 13.98	40	40 - 41	2288.49	2288.49 - 2381.19
400	14.07	14.07 - 14.23	40	40 - 41	2393.05	2393.05 - 2479.95
420	14.27	14.27 - 14.54	39	39 - 40	2442.92	2442.92 - 2593.9
440	14.2	14.2 - 14.33	41	41 - 41	2592.68	2592.68 - 2688.66
460	14.44	14.44 - 14.6	39	39 - 40	2714.34	2714.34 - 2790.47
480	14.59	14.59 - 14.72	38	38 - 40	2719.65	2719.65 - 2839.89
500	14.51	14.51 - 14.71	40	40 - 42	2976.9	2976.9 - 3017.42
520	14.54	14.54 - 14.69	40	40 - 41	3099.66	3099.66 - 3120.27
540	14.37	14.37 - 14.82	40	40 - 42	3152.85	3152.85 - 3278.67
560	14.75	14.75 - 14.81	41	41 - 42	3305.94	3305.94 - 3324.79
580	14.72	14.72 - 14.9	41	41 - 42	3451.64	3451.64 - 3507.02
600	14.75	14.75 - 15.11	40	40 - 41	3463.37	3463.37 - 3501.05
620	14.66	14.66 - 14.76	42	42 - 42	3691.14	3691.14 - 3703.22
640	14.71	14.71 - 14.86	41	41 - 42	3783.73	3783.73 - 3828.91
660	14.88	14.88 - 15.16	39	39 - 41	3849.91	3849.91 - 3884.75
680	14.82	14.82 - 15.12	41	41 - 42	3937.83	3937.83 - 4017.77

Method RFR (2)

Table 29 Performance metrics Method RFR (2) (1/2)

# potential orders	Distance (km)		Travel time (min.)		# routes	
	Q2	Q1-Q3	Q2	Q1-Q3	Q2	Q1-Q3
20	60.13	60.13 - 68.55	236.11	236.11 - 262.16	2	2 - 2
40	80.71	80.71 - 84.39	347.11	347.11 - 357.88	2	2 - 2
60	94.02	94.02 - 95.45	434.52	434.52 - 438.47	3	3 - 3
80	104.95	104.95 - 107.59	511.89	511.89 - 515.86	3	3 - 3
100	112.29	112.29 - 114.52	563.25	563.25 - 585.09	3	3 - 3
120	123.03	123.03 - 126.97	658.21	658.21 - 670.93	4	4 - 4
140	131.29	131.29 - 135.59	731.32	731.32 - 739.71	5	5 - 5
160	134.12	134.12 - 135.83	775.97	775.97 - 788.63	5	5 - 5
180	148.43	148.43 - 150.77	869.45	869.45 - 878.58	5	5 - 5
200	158.83	158.83 - 160.66	938.09	938.09 - 949.36	6	6 - 6
220	165.68	165.68 - 166.49	1010.15	1010.15 - 1027.09	6	6 - 6
240	177.31	177.31 - 180.07	1099.89	1099.89 - 1109.52	7	7 - 7
260	178.7	178.7 - 181.17	1121.75	1121.75 - 1149.03	8	8 - 8
280	184.77	184.77 - 190.92	1192.85	1192.85 - 1229.59	8	8 - 8
300	192.66	192.66 - 199.94	1265.65	1265.65 - 1284.49	8	8 - 8
320	210.18	210.18 - 215.61	1364.62	1364.62 - 1371.04	9	9 - 9
340	221.7	221.7 - 227.47	1449.63	1449.63 - 1468.12	9	9 - 10
360	221.49	221.49 - 225.03	1473.56	1473.56 - 1486.53	10	10 - 10

380	242.66	242.66 - 249.94	1591.03	1591.03 - 1637.28	11	11 - 11
400	240.35	240.35 - 255.96	1644.86	1644.86 - 1677.96	11	11 - 11
420	250.77	250.77 - 256.36	1696.79	1696.79 - 1710.7	11	11 - 12
440	260.73	260.73 - 272.58	1800.17	1800.17 - 1846.42	12	12 - 12
460	260.08	260.08 - 264.43	1816.59	1816.59 - 1864.74	12	12 - 13
480	265.85	265.85 - 273.1	1889.71	1889.71 - 1919.99	13	13 - 13
500	283.41	283.41 - 301.47	1989.5	1989.5 - 2040.16	14	14 - 14
520	289	289 - 302.47	2041.51	2041.51 - 2092.57	14	14 - 14
540	310.69	310.69 - 322.17	2150.31	2150.31 - 2196.44	15	15 - 15
560	320.15	320.15 - 323.56	2261.63	2261.63 - 2274.54	15	15 - 16
580	320.68	320.68 - 334.34	2288.51	2288.51 - 2330.99	15	15 - 17
600	331.41	331.41 - 339.72	2379.19	2379.19 - 2392.71	16	16 - 17
620	355.46	355.46 - 357.65	2493.02	2493.02 - 2517.89	17	17 - 18
640	350	350 - 363.03	2527.01	2527.01 - 2542.42	18	18 - 18
660	355.81	355.81 - 362.26	2587.6	2587.6 - 2593.99	18	18 - 18
680	368.45	368.45 - 375.31	2651.82	2651.82 - 2683.72	19	19 - 19

Table 30 Performance metrics Method RFR (2) (2/2)

# potential orders	Orders per hour		# PC4s		Necessary time (min)	
	Q2	Q1-Q3	Q2	Q1-Q3	Q2	Q1-Q3
20	5.08	5.08 - 5.61	19	19 - 20	236.11	236.11 - 262.16
40	6.91	6.91 - 7.17	29	29 - 30	347.11	347.11 - 357.88
60	8.28	8.28 - 8.51	33	33 - 33	434.52	434.52 - 438.47
80	9.36	9.36 - 9.44	38	38 - 39	511.89	511.89 - 515.86
100	10.13	10.13 - 10.42	39	39 - 39	563.25	563.25 - 585.09
120	10.84	10.84 - 10.9	39	39 - 42	658.21	658.21 - 670.93
140	11.38	11.38 - 11.5	40	40 - 41	731.32	731.32 - 739.71
160	11.95	11.95 - 12.06	39	39 - 40	775.97	775.97 - 788.63
180	12.29	12.29 - 12.56	40	40 - 42	869.45	869.45 - 878.58
200	12.58	12.58 - 12.79	40	40 - 41	938.09	938.09 - 949.36
220	12.85	12.85 - 12.98	41	41 - 42	1010.15	1010.15 - 1027.09
240	13.02	13.02 - 13.09	41	41 - 42	1099.89	1099.89 - 1109.52
260	13.21	13.21 - 13.42	38	38 - 40	1121.75	1121.75 - 1149.03
280	13.43	13.43 - 13.57	40	40 - 41	1192.85	1192.85 - 1229.59
300	13.89	13.89 - 13.89	40	40 - 40	1265.65	1265.65 - 1284.49
320	13.64	13.64 - 13.92	41	41 - 41	1364.62	1364.62 - 1371.04
340	13.49	13.49 - 13.96	39	39 - 40	1449.63	1449.63 - 1468.12
360	13.91	13.91 - 14.21	39	39 - 40	1473.56	1473.56 - 1486.53
380	13.69	13.69 - 14.08	41	41 - 41	1591.03	1591.03 - 1637.28
400	14.11	14.11 - 14.37	40	40 - 41	1644.86	1644.86 - 1677.96
420	14.28	14.28 - 14.42	39	39 - 40	1696.79	1696.79 - 1710.7
440	14.21	14.21 - 14.3	40	40 - 41	1800.17	1800.17 - 1846.42
460	14.5	14.5 - 14.62	37	37 - 40	1816.59	1816.59 - 1864.74
480	14.71	14.71 - 14.78	39	39 - 40	1889.71	1889.71 - 1919.99
500	14.36	14.36 - 14.76	39	39 - 40	1989.5	1989.5 - 2040.16
520	14.61	14.61 - 14.72	40	40 - 41	2041.51	2041.51 - 2092.57

540	14.31	14.31 - 14.72	40	40 - 42	2150.31	2150.31 - 2196.44
560	14.77	14.77 - 14.81	41	41 - 42	2261.63	2261.63 - 2274.54
580	14.75	14.75 - 14.99	41	41 - 41	2288.51	2288.51 - 2330.99
600	14.85	14.85 - 15.08	41	41 - 41	2379.19	2379.19 - 2392.71
620	14.73	14.73 - 14.87	41	41 - 42	2493.02	2493.02 - 2517.89
640	14.78	14.78 - 14.86	41	41 - 41	2527.01	2527.01 - 2542.42
660	14.91	14.91 - 15.06	39	39 - 41	2587.6	2587.6 - 2593.99
680	14.82	14.82 - 15.14	40	40 - 42	2651.82	2651.82 - 2683.72

Appendix F Assumption normal distribution

We state in Section 5.2.2 that we cannot assume a normal distribution for the order selections retrieved from the simulation. Hence, we show what the effect is of not assuming a normal distribution, and how this might have affected our results. Accordingly, we conduct a paired t-test with a significance level of 5% between the order selections, i.e., we assume a normal distribution ($\sim N(\mu, \sigma)$), and we conduct a Wilcoxon Signed Rank Test with a significance level of 5%, i.e., a non-parametric statistical hypothesis that does not assume normality in the data ($\neq N(\mu, \sigma)$).

Table 31 shows for 40, 140 and 460 potential orders if the difference in order selection retrieved by two individual methods is significant or not, based on the Paired T-Test and Wilcoxon Signed Rank Test. We observe that the difference in order selection methods is never significant if we assume a normal distribution. However, if we do not assume a normal distribution, the difference between the Method 'Optimal levels' and the other three order selection methods is significant. The mutual difference between the other three methods stays insignificant, independent of the normal distribution assumption.

Table 31 Effect in significance between methods whilst assuming a normal distribution

# Potential orders	Method	Daganzo		RFR (1)		RFR (2)	
		$\sim N(\mu, \sigma)$	$\neq N(\mu, \sigma)$	$\sim N(\mu, \sigma)$	$\neq N(\mu, \sigma)$	$\sim N(\mu, \sigma)$	$\neq N(\mu, \sigma)$
40	'Optimal levels'	Not sign.	Sign.	Not sign.	Sign.	Not sign.	Sign.
	Daganzo	x	x	Not sign.	Not sign.	Not sign.	Not sign.
	RFR (1)	x	x	x	x	Not sign.	Not sign.
	RFR (2)	x	x	x	x	x	x
140	'Optimal levels'	Not sign.	Sign.	Not sign.	Sign.	Not sign.	Sign.
	Daganzo	x	x	Not sign.	Not sign.	Not sign.	Not sign.
	RFR (1)	x	x	x	x	Not sign.	Not sign.
	RFR (2)	x	x	x	x	x	x
460	'Optimal levels'	Not sign.	Not sign.	Not sign.	Not sign.	Not sign.	Not sign.
	Daganzo	x	x	Not sign.	Not sign.	Not sign.	Not sign.
	RFR (1)	x	x	x	x	Not sign.	Not sign.
	RFR (2)	x	x	x	x	x	x

This difference can be explained by the assumptions of the tests. The Paired T-Test compares the means of the two methods and tests whether the mean of the differences between the two methods is significantly different from zero. Contrary, the Wilcoxon Signed Rank Test evaluates the ordering of the data, and thus whether the median of the differences between the two methods is significantly different from zero. Note that in a normal distribution, the mean is equal to the median, and thus we would expect similar conclusions from both tests.

We observed that for 40 and 140 potential orders, Methods Daganzo, RFR (1) and RFR (2) select a higher number of orders (40 or (skewed to) 140) than Method 'Optimal levels'. The different results of the two tests confirm that indeed no normal distribution is observed. However, with 460 potential orders, the order selection choice is less obvious and less skewed to the 460 potential orders. The order selection choice of the Method 'Optimal levels' coincides more with the order selection of the other three methods. Hence, we observe a spread of all four methods that corresponds more with the spread of a normal distribution. Therefore, we can conclude that a Wilcoxon Signed Rank Test should be performed to compare the order selection methods when the order selection choice is not obvious, i.e., with a small number of potential orders. When the choice becomes less obvious, the spread of the order selection becomes (closer to) the spread of a normal distribution, and hence the results of the two tests are probably similar.

Appendix G Information guide

This guide explains where to find or how to retrieve the input data to implement the solution methodology at all CBB's hubs. Hence, the current input data should be replaced or expanded with the data of all hubs. We refer to both the storage location and the responsible employee of this data. Note that we assign the responsibility of our created databases in both BiqQuery and Google Sheets to the data science team. All BiqQuery tables contain the most up-to-date information. **Due to confidentiality,** we **do not specify the exact storage location in this appendix. Hence, we only specify if the information is stored in a Google Sheet or a BiqQuery table.**

The following information is necessary per hub for the order selection:

1. PC4 information list

This list must contain all PC4s located in the delivery area of the hub.

- Storage place: Google Sheet
- Responsible: Owner capacity CoolblueBikes

For every PC4 on the PC4 information list, the information listed below must be provided. Note that we made this list already in BiqQuery (with information a, b & c). However, we display where to find (the source of) this information as well for the sake of completeness.

a. Level

The level assigned by CBB, if a comparison with the current system is desired.

- Storage place: Google Sheet
- Responsible: Owner capacity CoolblueBikes

b. Area

The level assigned by CBB, if a comparison with the current system is desired.

- Storage place: BiqQuery table
- Responsible: Data scientist

c. Distance

The distance of the centroid of the PC4 to the depot.

- Storage place: BiqQuery table
- Responsible: Data scientist team

d. Stop time

The stop time of an order per PC4.

- Storage place: Google Sheet
- Responsible: Owner on-time performance Delivery & Installations

e. Order probability distribution

The probability of observing *o* orders in a PC4.

- Storage place: This distribution still has to be defined. We learned from the hub in Hilversum that most likely the discrete probability distribution is negative binomial. The data to fit the distribution can be found in the BiqQuery table. Here, ensure that the chosen hub is set to the hub that you are going to analyze. Then, filter all observations per PC4 and fit the data on the negative binomial distribution as described in Section 4.4.1.
- Responsible: Data scientist team

2. Address

The addresses of the hub

- Storage place: Google Sheet
- Responsible: Manager CoolblueBikes Operations

3. **'Optimal levels'**
   The 'optimal levels' of the hub, if a comparison with the current system is desired.
   - Storage place: Google Sheet
   - Responsible*:* Owner capacity CoolblueBikes

4. **Bike speed**
   The average bike speed per hub.
   - Storage place: Google Sheet
   - Responsible: Owner on-time performance Delivery & Installations

5. **Costs per order**
   The fixed costs, the variable costs per order, the variable costs per hour and the variable costs per kilometre. We summarized these costs for all hubs, similarly to how we retrieved those costs for the hub in Hilversum. Note that this information should be updated to be the most representative.
   - Storage place: Google Sheet
   - Responsible: Owner capacity CoolblueBikes and data scientist team

6. **Order volume probability distribution**
   The probability of observing an order with volume *v.* If wished for, the distribution of all the hubs can also be used. We defined this distribution and stored this in the Google Sheet.
   - Storage: BiqQuery table. Here, ensure that depot_potential_huidig is set to the hub that you are going to analyze. Also, ensure that the bins in cm^3 are: 0-1999, 2000-3999, 4000-5999, 6000-7999, 8000-9999, 10000-11999, 12000-13999, 14000-15999, 16000-17999 & 18000-19999.
   - Responsible: Analyst CoolblueBikes and data scientist team

Furthermore, for the development of a machine learning model, the following information is necessary:

7. **Tour information (training data)**
   The grouped historical data includes all the features and the response (distance).
   - Storage: BiqQuery table. Here, ensure that the hub is set to all small or large hubs when utilizing this data set. The large hubs are Amsterdam Overamstel, Rotterdam, Den Haag and Utrecht. The remaining hubs are small hubs.
   - Responsible: Data scientist team

8. **PC4 order division**
   The historical division of all potential orders over the PC4s from almost 4 years of data.
   - Storage: BiqQuery table. Here, ensure that the depot is set to the hub that you are going to analyze.
   - Responsible: Data scientist team

9. **Address list**
   The random addresses per PC4 are based on historical orders.
   - Storage: BiqQuery table. Again, note that the depot should be changed into the hub to be analysed.
   - Responsible: Data scientist team

Appendix H Costs per order per order selection

Figure 48 shows the costs per order per order selection of the estimated and actual, constructed costs per order of the Methods Daganzo, RFR (1), RFR (2) and the operational level RFR.

Figure 48 Estimated and constructed costs per order of methods Daganzo, RFR (1), RFR (2) and operational level RFR

Appendix I Costs per order

Table 32 presents the costs per order per number of selected orders by Method Daganzo, RFR 1, RFR 2 and the operational level RFR, corresponding with Appendix H. The lowest value of each order selection is marked green. We observe that the operational level RFR outperforms the tactical level methods most of the time.

Table 32 Constructed costs per order of the Methods Daganzo, RFR (1), RFR (2) and operational level RFR

	Daganzo	RFR (1)	RFR (2)	Operational RFR
1	4410.9%	4411.4%	4405.3%	4428.1%
2	2230.5%	2226.5%	2259.6%	2241.6%
3	1494.2%	1496.7%	1513.5%	1501.9%
4	1127.5%	1128.0%	1146.9%	1132.0%
5	907.0%	907.0%	922.4%	910.0%
6	759.5%	759.5%	772.8%	762.1%
7	654.4%	654.4%	666.5%	656.4%
8	596.5%	578.6%	585.9%	577.0%
9	532.4%	516.7%	523.7%	515.2%
10	481.7%	468.9%	473.7%	465.9%
11	440.2%	428.2%	432.7%	425.6%
12	405.4%	394.3%	398.6%	392.0%
13	375.9%	365.9%	369.7%	363.8%
14	350.8%	341.3%	344.8%	339.6%
15	329.1%	323.0%	323.6%	318.5%
16	310.0%	304.3%	304.7%	300.2%
17	293.0%	290.8%	288.1%	284.0%
18	278.0%	275.9%	273.3%	269.6%
19	264.6%	263.6%	260.1%	256.7%
20	252.6%	252.0%	248.6%	245.0%
21	241.9%	241.3%	237.8%	234.7%
22	232.0%	231.6%	228.0%	225.4%
23	222.9%	222.8%	219.1%	216.7%
24	214.7%	214.5%	210.9%	209.0%
25	211.9%	206.9%	203.4%	201.6%
26	205.1%	199.8%	199.6%	194.8%
27	198.3%	193.2%	193.1%	188.4%
28	192.1%	187.2%	187.1%	182.9%
29	186.6%	181.6%	181.4%	177.5%
30	181.2%	176.3%	176.4%	172.4%
31	176.1%	171.4%	171.5%	167.5%
32	171.4%	166.9%	166.9%	163.1%
33	166.9%	162.6%	162.7%	159.6%
34	162.6%	158.5%	158.8%	155.6%
35	162.7%	154.7%	155.0%	151.7%
36	159.2%	151.0%	151.4%	148.1%
37	155.4%	149.7%	148.0%	144.7%
38	151.9%	146.7%	144.7%	141.6%
39	148.5%	143.5%	141.5%	138.5%
40	145.3%	140.6%	139.6%	135.7%

41	142.2%	137.7%	136.9%	132.9%
42	139.6%	135.1%	134.2%	130.4%
43	137.0%	132.5%	132.5%	128.1%
44	134.4%	130.1%	130.0%	125.8%
45	132.0%	127.7%	130.8%	123.6%
46	129.6%	125.4%	128.4%	121.6%
47	127.3%	123.3%	126.2%	119.5%
48	125.2%	121.2%	124.0%	117.5%
49	123.2%	119.2%	122.1%	115.5%
50	121.3%	117.4%	120.1%	113.8%
51	119.4%	115.7%	118.2%	112.0%
52	117.6%	113.9%	116.3%	110.7%
53	115.9%	112.2%	114.5%	109.1%
54	114.2%	110.5%	112.8%	107.7%
55	112.6%	109.0%	111.1%	106.2%
56	111.0%	107.5%	109.5%	104.7%
57	109.6%	106.0%	108.1%	103.4%
58	108.1%	104.5%	106.6%	102.1%
59	106.7%	103.3%	105.2%	100.7%
60	105.3%	102.0%	103.8%	99.4%
61	103.9%	100.7%	102.5%	98.5%
62	102.6%	99.4%	101.3%	97.3%
63	101.4%	98.2%	100.1%	96.1%
64	100.2%	97.0%	98.9%	95.0%
65	99.0%	96.6%	97.7%	93.9%
66	97.9%	95.4%	96.7%	92.8%
67	96.9%	94.5%	95.5%	91.8%
68	95.8%	93.6%	94.4%	90.9%
69	94.8%	92.6%	93.4%	89.9%
70	93.9%	94.8%	92.7%	89.0%
71	92.9%	93.8%	91.8%	88.1%
72	93.3%	92.8%	90.9%	87.3%
73	92.4%	91.8%	90.0%	86.4%
74	91.5%	90.9%	89.1%	85.6%
75	90.6%	90.0%	90.2%	84.8%
76	89.7%	89.1%	89.3%	84.0%
77	88.9%	88.3%	88.5%	83.4%
78	88.0%	87.5%	87.7%	82.6%
79	87.2%	86.7%	86.9%	81.9%
80	86.5%	85.9%	86.1%	81.2%
81	86.0%	85.1%	85.3%	80.6%
82	85.4%	84.8%	84.6%	79.9%
83	84.6%	84.2%	83.9%	79.2%
84	83.8%	83.6%	83.1%	78.6%
85	83.1%	82.9%	82.4%	78.0%
86	82.4%	82.2%	81.7%	77.5%
87	82.0%	81.5%	81.1%	77.1%
88	81.3%	80.9%	80.4%	76.4%

89	80.6%	80.3%	79.8%	75.9%
90	80.0%	79.6%	79.1%	75.5%
91	79.5%	79.0%	78.6%	74.9%
92	78.9%	78.4%	78.0%	74.3%
93	78.4%	77.8%	77.4%	73.8%
94	78.0%	77.2%	76.9%	73.3%
95	77.4%	76.8%	76.3%	72.9%
96	76.9%	76.3%	75.7%	72.4%
97	76.4%	76.1%	75.2%	72.1%
98	75.9%	75.6%	74.6%	71.6%
99	75.4%	75.0%	74.2%	71.1%
100	74.9%	74.5%	73.8%	70.6%
101	74.4%	74.0%	73.5%	70.1%
102	73.9%	73.5%	73.2%	69.8%
103	73.4%	73.1%	73.6%	69.4%
104	73.0%	72.6%	73.1%	69.1%
105	72.6%	72.1%	72.6%	68.7%
106	72.1%	71.7%	72.2%	68.2%
107	71.6%	71.2%	71.7%	68.0%
108	71.2%	70.7%	71.3%	67.6%
109	70.7%	70.3%	70.9%	67.3%
110	70.3%	69.9%	70.5%	66.9%
111	69.9%	69.4%	70.1%	66.5%
112	69.5%	69.0%	69.8%	66.1%
113	69.1%	68.6%	69.4%	65.7%
114	68.7%	68.2%	69.0%	66.0%
115	68.3%	67.8%	68.6%	65.7%
116	69.1%	67.4%	68.3%	65.4%
117	68.8%	67.0%	68.0%	65.0%
118	68.4%	66.6%	67.6%	64.7%
119	68.0%	66.3%	67.3%	64.3%
120	67.6%	66.0%	66.9%	64.0%
121	67.3%	65.6%	66.6%	63.7%
122	66.9%	65.4%	66.2%	63.3%
123	66.5%	66.0%	66.0%	63.1%
124	66.2%	65.9%	65.6%	63.0%
125	65.9%	65.6%	65.3%	62.7%
126	65.5%	65.2%	64.9%	62.4%
127	65.3%	64.9%	64.7%	62.4%
128	64.9%	64.6%	64.3%	62.1%
129	64.6%	64.2%	64.0%	61.7%
130	64.3%	63.9%	63.7%	61.5%
131	64.0%	63.6%	63.4%	61.2%
132	63.7%	63.3%	63.1%	61.0%
133	63.4%	63.0%	62.8%	60.7%
134	63.1%	62.8%	62.5%	60.5%
135	62.8%	62.5%	62.2%	60.2%
136	62.6%	62.1%	61.9%	60.0%

137	62.3%	62.6%	61.6%	59.7%
138	62.0%	62.3%	62.4%	59.4%
139	61.7%	62.0%	62.1%	59.1%
140	61.5%	61.8%	61.9%	58.9%
141	61.3%	61.6%	61.6%	58.9%
142	61.0%	61.4%	61.4%	58.9%
143	60.8%	61.2%	61.1%	58.9%
144	60.5%	61.0%	61.2%	58.7%
145	60.2%	60.7%	61.0%	58.5%
146	60.0%	60.5%	60.7%	58.4%
147	59.8%	60.3%	60.5%	58.3%
148	59.5%	60.1%	60.2%	58.1%
149	59.7%	59.9%	60.0%	57.8%
150	59.5%	59.6%	59.7%	57.6%
151	59.3%	59.4%	59.6%	57.8%
152	59.7%	59.3%	59.4%	57.6%
153	60.2%	59.1%	59.2%	57.4%
154	59.9%	58.8%	59.0%	57.2%
155	59.7%	58.6%	58.7%	57.1%
156	59.4%	58.3%	58.7%	56.8%
157	59.2%	58.1%	58.5%	56.7%
158	58.9%	57.9%	58.3%	56.7%
159	58.8%	57.7%	58.1%	56.5%
160	58.5%	57.7%	57.8%	56.3%
161	58.3%	57.5%	57.6%	56.1%
162	58.1%	57.3%	57.4%	55.9%
163	57.9%	57.1%	57.2%	55.7%
164	57.7%	56.9%	57.0%	55.7%
165	57.6%	56.7%	56.9%	55.6%
166	57.4%	56.5%	56.7%	55.4%
167	57.2%	56.3%	56.7%	55.3%
168	57.0%	56.1%	56.4%	55.1%
169	56.9%	55.9%	56.3%	54.9%
170	56.8%	55.7%	56.5%	54.7%
171	56.6%	55.7%	56.4%	54.5%
172	56.5%	55.5%	56.3%	54.3%
173	56.3%	55.4%	56.2%	54.1%
174	56.1%	55.2%	56.1%	53.9%
175	55.9%	55.0%	55.8%	53.8%
176	55.8%	55.0%	55.8%	53.6%
177	55.6%	54.9%	56.2%	53.4%
178	55.4%	54.8%	56.0%	53.3%
179	55.2%	54.8%	55.9%	53.1%
180	55.1%	54.6%	55.7%	53.0%
181	55.0%	54.5%	55.5%	52.8%
182	54.7%	54.4%	55.3%	52.6%
183	54.6%	54.2%	55.1%	52.4%
184	55.3%	54.1%	55.0%	52.4%

185	55.2%	54.0%	54.8%	52.2%
186	55.0%	53.8%	54.6%	52.1%
187	54.8%	53.6%	54.4%	51.9%
188	54.7%	53.5%	54.3%	51.8%
189	54.6%	53.4%	54.1%	51.6%
190	54.5%	53.3%	53.9%	51.5%
191	54.2%	53.2%	53.7%	51.4%
192	54.1%	53.0%	53.6%	51.3%
193	53.9%	52.8%	53.4%	51.2%
194	53.8%	52.7%	53.3%	51.0%
195	53.6%	52.5%	53.2%	50.9%
196	53.5%	52.4%	53.1%	50.8%
197	53.3%	52.3%	53.1%	50.6%
198	53.1%	52.2%	52.8%	50.6%
199	53.0%	52.2%	52.7%	50.4%
200	52.8%	52.0%	52.7%	50.4%
201	52.8%	51.9%	52.4%	50.2%
202	52.7%	51.9%	52.4%	50.1%
203	52.6%	52.1%	52.4%	50.0%
204	52.6%	51.8%	52.4%	49.9%
205	52.7%	51.8%	52.2%	49.9%
206	52.5%	51.7%	52.0%	49.8%
207	52.4%	51.5%	52.1%	49.8%
208	52.3%	51.4%	51.8%	49.7%
209	52.2%	51.3%	51.8%	49.6%
210	52.1%	51.1%	51.6%	49.5%
211	52.0%	51.0%	51.6%	49.3%
212	51.9%	50.9%	51.5%	49.3%
213	51.8%	50.8%	51.5%	49.2%
214	51.8%	50.7%	51.4%	49.1%
215	51.6%	50.6%	51.3%	49.0%
216	51.5%	50.5%	51.8%	48.9%
217	51.4%	50.3%	51.7%	48.8%
218	51.7%	50.2%	51.6%	48.7%
219	51.5%	50.1%	51.5%	48.6%
220	51.4%	50.0%	51.4%	48.5%
221	51.3%	49.9%	51.2%	48.4%
222	51.2%	49.9%	51.2%	48.2%
223	51.1%	49.6%	51.0%	48.2%
224	51.0%	49.5%	50.9%	48.1%
225	50.9%	49.4%	51.1%	48.1%
226	50.7%	49.3%	51.0%	48.1%
227	50.7%	49.3%	50.8%	48.0%
228	51.0%	49.3%	50.8%	48.0%
229	50.9%	49.1%	50.7%	47.9%
230	50.8%	49.1%	50.6%	47.8%
231	50.7%	49.0%	50.5%	47.8%
232	50.6%	48.9%	50.4%	47.7%

233	50.4%	48.8%	50.3%	47.7%
234	50.4%	48.7%	50.2%	47.6%
235	50.3%	48.6%	50.1%	47.5%
236	50.2%	48.5%	50.0%	47.3%
237	50.1%	48.4%	49.9%	47.3%
238	50.0%	48.4%	49.9%	47.2%
239	50.0%	48.2%	50.7%	47.0%
240	49.9%	48.3%	50.6%	47.0%
241	49.8%	48.2%	50.5%	46.8%
242	49.8%	48.2%	50.4%	46.8%
243	49.7%	48.1%	50.3%	46.8%
244	49.6%	48.0%	50.2%	46.8%
245	49.5%	47.9%	50.3%	46.8%
246	49.4%	47.8%	50.2%	46.8%
247	49.3%	47.7%	50.1%	46.7%
248	49.1%	47.6%	49.9%	46.6%
249	49.8%	47.5%	49.9%	46.5%
250	49.7%	47.4%	49.8%	46.5%
251	49.7%	47.3%	49.7%	46.4%
252	49.7%	47.3%	49.6%	46.4%
253	49.5%	47.2%	49.6%	46.3%
254	49.4%	47.2%	49.5%	46.2%
255	49.4%	47.5%	49.4%	46.1%
256	50.1%	47.5%	49.4%	46.1%
257	50.1%	47.4%	49.4%	46.1%
258	50.0%	47.3%	49.2%	45.9%
259	49.9%	47.3%	49.1%	45.9%
260	49.8%	47.2%	49.1%	45.9%
261	49.6%	47.1%	49.1%	45.8%
262	49.7%	47.2%	48.8%	45.7%
263	49.6%	47.1%	48.9%	45.6%
264	49.5%	47.1%	48.7%	45.6%
265	49.4%	47.1%	48.5%	45.5%
266	49.3%	47.0%	48.4%	45.4%
267	49.1%	46.8%	48.6%	45.3%
268	49.1%	46.8%	48.4%	45.2%
269	49.1%	46.8%	49.1%	45.1%
270	49.0%	46.7%	49.1%	45.1%
271	48.8%	46.6%	48.9%	44.9%
272	48.7%	46.5%	48.8%	44.9%
273	48.7%	46.3%	48.8%	44.9%
274	48.7%	46.3%	48.7%	44.7%
275	48.6%	46.2%	48.5%	44.7%
276	48.6%	46.1%	48.6%	44.6%
277	48.4%	46.1%	48.4%	44.5%
278	48.3%	46.3%	48.3%	44.5%
279	48.2%	46.2%	48.4%	44.5%
280	48.1%	46.4%	48.3%	44.4%

281	48.1%	46.3%	48.4%	44.4%
282	48.0%	46.3%	48.2%	44.2%
283	47.9%	46.2%	48.2%	44.3%
284	47.9%	46.2%	48.1%	44.2%
285	47.7%	46.0%	48.1%	44.0%
286	47.9%	46.1%	48.0%	44.1%
287	47.7%	46.4%	47.9%	44.0%
288	47.6%	46.3%	47.8%	43.9%
289	47.4%	46.2%	47.7%	43.9%
290	47.5%	46.1%	47.7%	43.9%
291	47.4%	46.1%	47.6%	43.8%
292	47.2%	46.0%	47.5%	43.8%
293	47.2%	45.9%	47.5%	43.7%
294	47.1%	46.0%	47.3%	43.8%
295	46.9%	45.9%	47.3%	43.7%
296	47.0%	45.8%	47.1%	43.6%
297	47.0%	45.7%	47.1%	43.6%
298	46.9%	45.5%	46.9%	43.5%
299	46.9%	45.4%	47.0%	43.7%
300	46.7%	45.5%	46.7%	43.7%
301	46.7%	45.4%	46.9%	43.7%
302	46.6%	45.3%	46.7%	43.6%
303	46.5%	45.2%	46.7%	43.6%
304	46.5%	45.2%	46.6%	43.7%
305	46.5%	45.1%	46.6%	43.5%
306	46.4%	45.1%	46.5%	43.5%
307	46.3%	45.0%	46.3%	43.5%
308	46.3%	45.0%	46.4%	43.5%
309	46.3%	45.0%	46.3%	43.4%
310	46.2%	44.9%	46.2%	43.4%
311	46.1%	44.9%	46.2%	43.3%
312	46.1%	44.7%	46.1%	43.2%
313	46.1%	44.5%	46.1%	43.2%
314	46.0%	44.6%	45.9%	43.2%
315	46.0%	45.2%	46.1%	43.1%
316	45.8%	45.2%	45.7%	42.9%
317	45.8%	45.2%	45.8%	43.1%
318	45.6%	45.7%	45.8%	43.1%
319	45.7%	45.7%	45.7%	43.0%
320	45.6%	45.6%	45.7%	43.0%
321	45.5%	45.6%	45.7%	42.9%
322	45.5%	45.4%	45.5%	42.9%
323	45.4%	45.3%	45.5%	43.0%
324	45.4%	45.4%	45.2%	42.9%
325	45.2%	45.3%	45.1%	42.8%
326	45.1%	45.3%	45.3%	43.0%
327	45.1%	45.3%	45.4%	42.9%
328	45.0%	45.4%	45.3%	43.0%

329	45.1%	45.2%	45.1%	43.0%
330	44.9%	45.2%	45.1%	42.8%
331	45.0%	45.2%	45.1%	42.8%
332	44.9%	45.1%	45.0%	42.7%
333	44.8%	44.9%	45.0%	42.8%
334	44.7%	45.0%	44.9%	42.9%
335	44.9%	45.0%	44.9%	42.7%
336	44.8%	44.9%	44.8%	42.7%
337	44.8%	44.9%	44.8%	42.7%
338	44.9%	44.8%	44.7%	42.6%
339	44.8%	44.7%	44.7%	42.5%
340	44.9%	44.7%	44.7%	42.6%
341	44.6%	44.6%	44.6%	42.4%
342	44.8%	44.7%	44.5%	42.4%
343	44.7%	44.5%	44.5%	42.3%
344	44.8%	44.5%	44.3%	42.4%
345	44.7%	44.4%	44.4%	42.4%
346	44.7%	44.3%	44.3%	42.3%
347	44.4%	44.3%	44.3%	42.3%
348	44.5%	44.3%	44.2%	42.2%
349	44.5%	44.2%	44.2%	42.2%
350	44.5%	44.2%	44.2%	42.2%
351	44.4%	44.1%	44.2%	42.2%
352	44.3%	44.0%	44.3%	42.2%
353	44.2%	44.0%	44.1%	42.2%
354	44.2%	43.9%	44.4%	42.1%
355	44.1%	43.8%	44.3%	42.2%
356	44.1%	43.8%	44.2%	42.2%
357	44.1%	43.8%	44.2%	42.2%
358	44.0%	43.8%	44.2%	42.1%
359	43.9%	43.7%	44.1%	42.2%
360	43.9%	43.8%	44.1%	42.1%
361	43.9%	43.6%	44.0%	42.1%
362	43.8%	43.6%	44.0%	42.1%
363	43.8%	43.5%	44.0%	42.2%
364	43.7%	43.6%	44.0%	42.0%
365	43.9%	43.6%	43.9%	41.9%
366	43.7%	43.6%	43.8%	42.0%
367	43.7%	43.7%	43.8%	41.9%
368	43.6%	43.7%	43.7%	42.2%
369	43.7%	43.6%	43.7%	42.2%
370	43.6%	43.6%	43.7%	42.1%
371	43.7%	43.4%	43.6%	42.1%
372	43.6%	43.4%	43.5%	42.2%
373	43.6%	43.4%	43.6%	42.1%
374	43.5%	43.4%	43.7%	42.2%
375	43.5%	43.4%	43.6%	42.1%
376	43.4%	43.3%	43.5%	42.4%

377	43.4%	43.3%	43.5%	42.4%
378	43.4%	43.2%	43.5%	42.5%
379	43.3%	43.2%	43.4%	42.3%
380	43.2%	43.1%	43.4%	42.3%
381	43.2%	43.3%	43.3%	42.4%
382	43.2%	43.2%	43.3%	42.3%
383	43.1%	43.2%	43.2%	42.3%
384	43.1%	43.0%	43.1%	42.3%
385	42.9%	43.0%	43.2%	42.3%
386	42.9%	43.0%	43.2%	42.4%
387	42.9%	42.9%	43.1%	42.2%
388	42.8%	42.8%	43.1%	42.2%
389	42.8%	42.8%	43.0%	42.2%
390	42.7%	42.9%	42.9%	42.3%
391	42.7%	42.9%	42.8%	42.2%
392	42.9%	42.8%	42.8%	42.0%
393	42.7%	42.8%	42.9%	42.2%
394	42.7%	42.7%	42.7%	42.1%
395	42.6%	42.7%	42.8%	42.1%
396	42.7%	42.5%	42.7%	42.2%
397	42.7%	42.4%	42.5%	42.2%
398	42.6%	42.5%	42.7%	42.0%
399	42.5%	42.5%	42.5%	42.0%
400	42.5%	42.4%	42.6%	41.9%
401	42.4%	42.4%	42.6%	41.9%
402	42.4%	42.4%	42.6%	41.9%
403	42.4%	42.3%	42.4%	41.8%
404	42.3%	42.3%	42.3%	41.8%
405	42.4%	42.4%	42.4%	41.9%
406	42.3%	42.2%	42.3%	41.7%
407	42.2%	42.2%	42.3%	41.6%
408	42.3%	42.1%	42.3%	41.6%
409	42.2%	42.2%	42.1%	41.5%
410	42.1%	42.2%	42.3%	41.4%
411	42.1%	42.1%	42.1%	41.4%
412	42.1%	42.0%	42.1%	41.4%
413	42.1%	42.0%	42.0%	41.4%
414	42.0%	41.9%	42.1%	41.4%
415	42.0%	41.9%	41.8%	41.3%
416	41.9%	42.1%	42.0%	41.2%
417	41.9%	42.0%	42.0%	41.3%
418	41.7%	41.9%	41.9%	41.3%
419	41.9%	41.9%	41.9%	41.2%
420	41.8%	41.8%	42.0%	41.3%
421	41.7%	41.8%	42.0%	41.3%
422	41.8%	41.9%	41.9%	41.1%
423	41.8%	41.7%	41.9%	41.2%
424	41.7%	41.7%	41.9%	41.1%

425	41.4%	41.7%	41.9%	41.2%
426	41.7%	41.7%	41.8%	41.2%
427	41.6%	41.8%	41.9%	41.0%
428	41.5%	41.6%	41.7%	40.8%
429	41.4%	41.6%	41.7%	40.9%
430	41.5%	41.6%	41.7%	40.9%
431	41.5%	41.7%	41.7%	41.0%
432	41.4%	41.6%	41.6%	40.9%
433	41.4%	41.5%	41.6%	41.0%
434	41.5%	41.3%	41.5%	41.0%
435	41.5%	41.6%	41.4%	41.0%
436	41.5%	41.3%	41.4%	40.8%
437	41.4%	41.4%	41.3%	40.8%
438	41.4%	41.4%	41.4%	40.9%
439	41.3%	41.2%	41.3%	40.8%
440	41.3%	41.2%	41.2%	40.7%
441	41.3%	41.4%	41.3%	40.7%
442	41.2%	41.2%	41.3%	40.8%
443	41.2%	41.1%	41.1%	40.9%
444	41.2%	41.2%	41.0%	40.8%
445	41.2%	41.1%	40.9%	40.9%
446	41.0%	41.0%	40.9%	40.9%
447	41.2%	41.0%	41.2%	40.9%
448	41.1%	41.1%	41.1%	40.9%
449	41.1%	41.1%	41.1%	40.9%
450	41.1%	41.2%	41.0%	40.9%
451	40.9%	41.3%	40.9%	41.0%
452	41.0%	41.3%	40.9%	40.8%
453	41.1%	40.9%	41.1%	41.1%
454	40.9%	41.1%	41.1%	41.0%
455	41.0%	41.1%	41.1%	41.0%
456	40.9%	41.1%	41.1%	40.9%
457	41.0%	41.0%	41.0%	40.9%
458	40.9%	40.9%	40.9%	40.9%
459	40.9%	40.9%	41.0%	41.0%
460	40.8%	40.8%	40.8%	40.8%

Appendix J Costs per order composition Method Daganzo and RFR (2)

Figure 49 shows the estimated and constructed costs per order composition of the order selection by Method Daganzo. Figure 50 shows the estimated and constructed costs per order composition of the order selection by Method RFR (2). Figure 51 shows the estimated and constructed costs per order composition of the order selection by the operational level RFR. We observe more fluctuations in the constructed costs per order by Method Daganzo, probably because the orders selected are further away. Also, we see again that RFR overestimates the distance in the beginning, resulting in relatively high costs per order at the first orders selected (>360.0%)

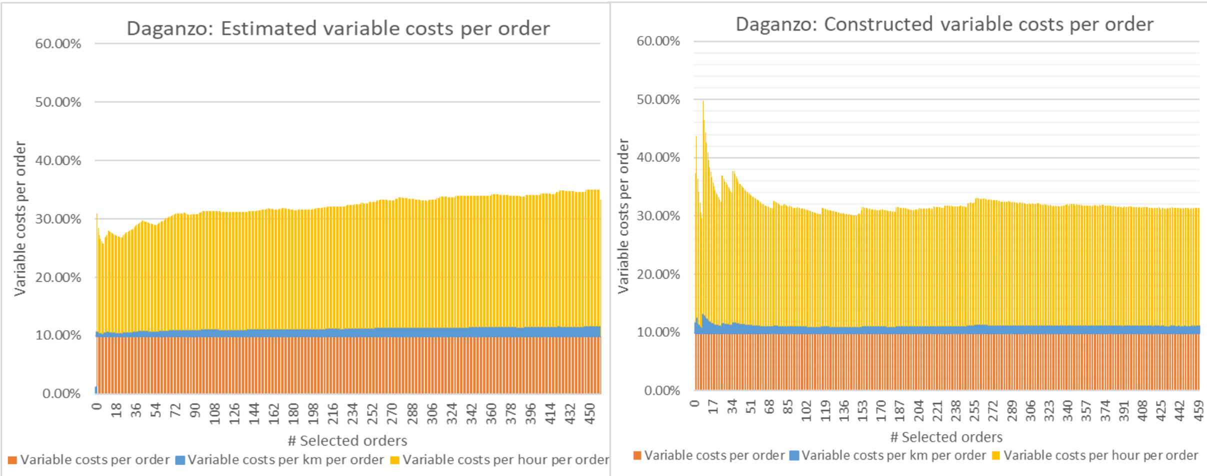

Figure 49 The estimated (left) and constructed (right) cost per order composition of the order selection of Daganzo

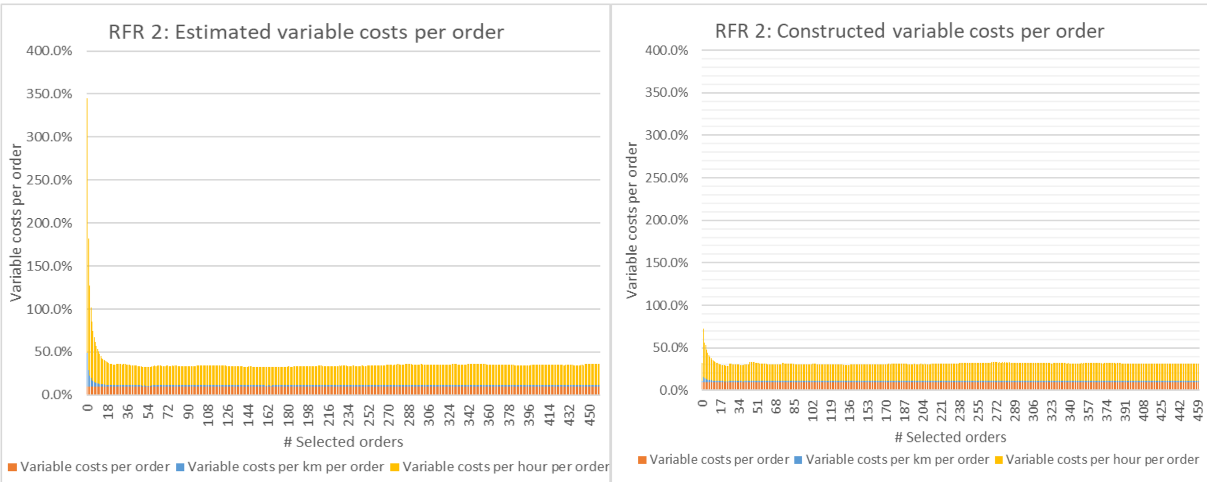

Figure 50 The estimated (left) and constructed (right) costs per order composition of the order selection of RFR (2)

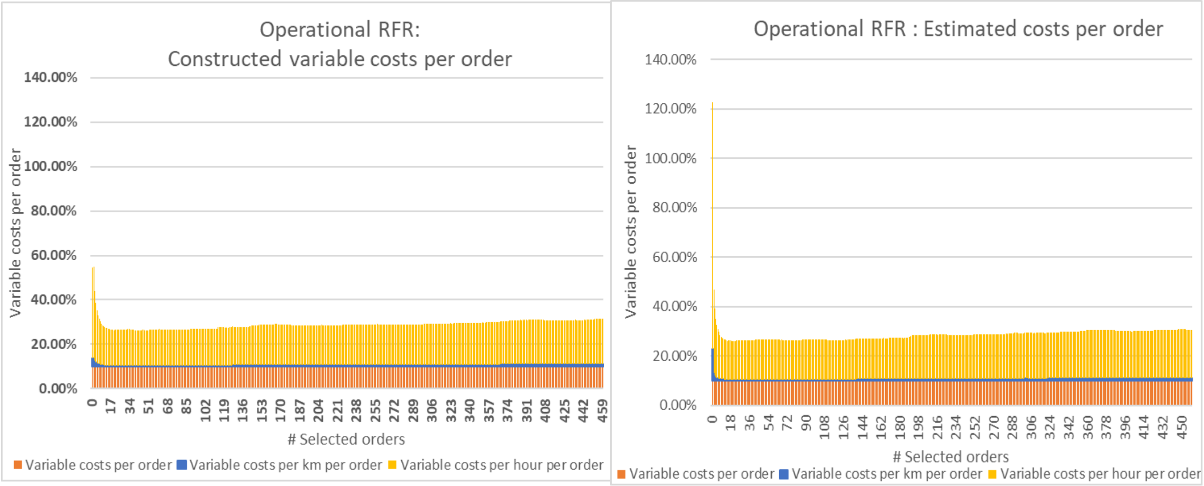

Figure 51 The estimated (left) and constructed (right) costs per order composition of the order selection of the operational level RFR

Appendix K Confidentiality appendix

We do not display the content of this appendix due to confidentiality.