Positioning of parcel pick-up locations for a large Dutch postal company

Tim Jonker

Erasmus University Rotterdam

Positioning of parcel pick-up locations for a large Dutch postal company

by

Tim Jonker

Master of Science
in Econometrics and Management Science

Program
Operations Research and Quantitative Logistics

at the Erasmus University Rotterdam,
to be defended publicly on Thursday August 6, 2020 at 14:00.

Graduation thesis:	FEM61008
Student number:	507099
Supervisor:	dr. R. Spliet
Company supervisor:	F. Shaban
Second assessor:	dr. T. Dollevoet

Abstract

Increased parcel volumes, increased service demands and low profit margins call for a sustainable cost efficient parcel delivery network. Retailers serving as parcel pick-up points that offer parcel consolidation respond to this call because they increase customer service and reduce delivery costs. However, these retailers can only be maintained or acquired into the parcel delivery network whenever the effort of parcel consolidation is small. Therefore, it is important that parcel pick-up points are not flooded with parcels for which there is no consolidation capacity available. This research investigates how much capacity is required at a retailer to satisfy customer demand in a cost efficient way. We focus on the parcel delivery network of PostNL which is the largest Dutch parcel delivery company. A modification to the Capacitated Facility Location Problem (CFLP) model is used to model the parcel delivery network and its requirements. The combination of features that we include in our model is novel. Furthermore, we introduce several methods to decompose the parcel delivery network into multiple smaller networks without loss of optimality to simplify and accelerate the solution process of the initial problem. Subsequently, we use a Branch and Bound (B&B) algorithm and a Tabu Search (TS) heuristic to solve the decomposed problems individually. The B&B algorithm is custom made and exploits some of the problem features that enable it to find exact solutions in reasonable time for almost all subproblems. The TS heuristic is capable of finding the optimal solution in 98.6 % of the subproblems. Moreover, we show that the combination of our mathematical model and solution technique is superior to PostNL's model in terms of: the probability to have sufficient capacity at parcel pick-up points, the parcel pick-up point costs, the number of additional parcel pick-up points and the required additional capacity at parcel pick-up points.

Preface

This report contains a graduation thesis as part of the Master Econometrics and Management Science with the specialization Operations Research and Quantitative Logistics. The graduation assignment is based on an agreement between PostNL and the Erasmus University Rotterdam. The assignment is provided by PostNL, who are responsible for the daily supervision. The main working environment is at home due to the corona crisis. The Erasmus University Rotterdam is responsible for the grading and the academic level of the thesis.

Tim Jonker
Rotterdam, July 2020

Contents

Acronyms

ACO Ant Colony Optimization. 12

B&B Branch and Bound. iii, 1, 2, 10, 12, 17, 22, 29–32, 34, 35, 37–39, 41, 45, 46

CFLP Capacitated Facility Location Problem. iii, 1, 2, 9–12

GA Genetic Algorithm. 10–12

HLP Hub Location Problem. 10

OLS Ordinary Least Squares. 15

PSO Particle Swarm Optimization. 10, 12

SA Simulated Annealing. 10

SCP Set Covering Problem. 12

SOCP Second Order Conic Programming. 7

TS Tabu Search. iii, 2, 10–12, 17, 32, 34, 35, 37–39, 45

UFLP Uncapacitated Facility Location Problem. 9, 10

VNS Variable Neighbourhood Search. 11, 12

List of Symbols

These symbols are used within the body of the document. Not all symbols are included nor do we give a formal description. The list is purely for reference and quick lookup of certain constants, graphs, sets, parameters and variables.

Constants

$\bar{D}_i$ Distance from customer $i \in N$ to the closest retailer $j \in M^o$

$c_j^{(k)}$ Capacity of the k^{th} largest retailer at location $j \in M$

n_j Number of additional Locatus locations for retailer $j \in N$

n_{open} The maximum number of retailers that can be opened until the upperbound is exceeded

q_{α_j} The quantile associated with α_j

Graphs

$G'(M^c, A)$ Undirected graph based on $G(M, A)$ where open retailers and tails are removed but, extra edges included

$G(M, A)$ Undirected graph with retailers on the nodes and edges from $i \in M$ to $j \in M_i^n$

$G(M, E)$ Directed graph with retailers on the nodes and edges from $i \in M^c$ to $j \in M_i^n$

$G_i'(M, E)$ A component of directed graph $G_i(M, E)$ where node i is removed

$G_i(M, E)$ A sub directed graph of $G(M, E)$ that can contains all nodes reachable from node i

Retail sets

$C^c(c^o)$ All combinations of retailers $j \in M_i^c$ with an arc to a retailer $k \in c^o$

C^o All combinations of retailers M_i^o

c^o A combination of retailers M_i^o

L A set of unions of potential retailers for customers $i \in N^d$

M Set of all retailers

M^b A set of retailers for which the fixed costs will not lead to exceeding the upper bound

M^c Set of all closed retailers

M^o Set of all open retailers

M_i Set of retailers with a potential connection to customer $i \quad \forall i \in N$

M_i^c Closed retailers in directed graph $G_i(M, E)$

M_i^o Open retailers in directed graph $G_i(M, E)$

Customer sets

N Set of all customers

N_j Set of all customers that visit retailer $j \quad \forall j \in M$

N_j^+ Maximal set of customers that visit retailer $j \quad \forall j \in M$

N_j^- Minimal set of customers that visit retailer $j \quad \forall j \in M$

Parameters

α_j The certainty with which retailer $j \in M$ should have sufficient capacity

ω Shift parameter used in the sigmoid function during the Tabu Tenure

ϕ Scale parameter used in the sigmoid function during the Tabu Tenure

a Minimum maximum distance to a retailer

b Maximum maximum distance to a retailer

f_j A constant used is the variable costs of retailer $j \in M$

f_j^n Fixed cost of a new retailer $j \in M$

f_j^o Fixed cost of an open retailer $j \in M$

p Fraction that a new retailer may be further than the current closest retailer

X Minimum area of a $PC4$ area to be eligible to be split up into $PC5$ areas

Y Minimum number of inhabitants of a $PC4$ area to be eligible to be split up into $PC5$ areas

Variables

β_{ij} Fraction of customers $i \in N$ that visit retailer $j \in M$

γ_i Fraction of customers $i \in N$ that favors the closest retailer $j \in M$

a_{ij} Indicates if retailer $j \in N$ can potentially satisfy the demand of customer $i \in N$

u_i Stochastic demand volume of customer $i \in N$, $u_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$

1

Introduction

E-commerce is a rapid growing market that leads to an ever increasing volume of parcels send around the world. This, combined with demand for fast, flexible and cheap delivery, puts stress on the current parcel delivery network. Integrating parcel pick up points, where customers pick up and deliver their parcels, into the network is one of the measures to enable a future-ready parcel delivery network. Using retail locations offers multiple advantages. First, an increasing number of customers demand that their parcels can be delivered not only at their homes, but at an easily accessible place. Thereby, parcel pick up points offer added flexibility and service. Furthermore, parcel pick up points consolidate parcels of failed home deliveries that are a major contributor to the high last-mile costs. Therefore, it is of paramount importance to integrate these parcel pick up points in the future parcel delivery network.

Because the profit margins of parcel delivery are small, it is important that these parcel pick up points should only be opened at strategic locations. Moreover, these parcel pick up points should not be flooded with parcels. Therefore, when determining the locations of these parcel pick up points the capacity should be sufficient to satisfy customers with an increasing uncertain demand for a long period of time.

In the Netherlands this development is no different [1]. PostNL is the largest parcel delivery company in the Netherlands that delivers over 800.000 parcels every day. Currently, PostNL uses existing retailers as parcel pick up points. Traditionally, these retailers allocate some space and time to handle parcels. However, some of these retailers do not have sufficient capacity to satisfy the current demand. Therefore, PostNL seeks to integrate new retailers into their current parcel delivery network to enable competitive parcel delivery for the coming years. The aim is to establish a parcel delivery network that is sustainable for the upcoming 5 years. A sustainable network does not only guarantee sufficient capacity at retailers, it further ensures that all households can reach such a location easily.

To determine the network structure, we use a Capacitated Facility Location Problem (CFLP) formulation where we recognise two different streams. One customer stream, where customers act according to their preference. And one PostNL stream, that can be regulated with certain restrictions. We make assumptions based on historical data and expert knowledge to forecast customer preference. Furthermore, we do not assume that future customer demand is known precisely. Therefore, we use normal distributions to model the customer demand within both streams. For each location in the Netherlands we determine whether a new retailer should be acquired and what the required size and associated costs would be.

Throughout this report we use the terms opening and closing a retailer to describe the process of acquiring a retailer as a parcel pick up point or ending the agreement between PostNL and the retailer respectively.

The CFLP is a complex problem in itself and for the Netherlands there are many potential retail locations and even more demand locations, which means many variables. Therefore, an off the shelf exact algorithm is infeasible because it takes too much time. Therefore, we develop methods to decompose our problem, without loss of optimality, into multiple subproblems. Moreover, we develop both a custom exact algorithm and a heuristic. The custom exact algorithm is a modification to the well-known Branch and Bound (B&B) algorithm. Where we introduce methods to decompose our problem into manageable sizes without excluding the optimal solution. The heuristic that we apply

is a Tabu Search (TS) heuristic where the discrete variables related to opening or closing a retailer are determined heuristically before assigning the demand locations to these opened retailers. From each obtained solution, we explore the neighbourhood for improving solutions, while putting previous solutions on the Tabu list to avoid getting stuck in local minima.

This research is novel because, to our best knowledge, there has not been any other research that uses a CFLP that incorporates all features that we include. Furthermore, our problem size is within the highest tier of CFLP sizes. Additionally, we show that exact solutions can be found in reasonable time by exploiting the problem structure. Furthermore, we show that TS is applicable with some modifications and is able to find the optimal solution frequently. By comparing the suggested open capacity of our newly developed model with PostNL's current model we found that our new model opens far fewer locations of more realistic sizes. Moreover, we find that using a probability distribution to model customer demand is superior compared to a fixed margin in terms of costs and the probability that the open capacity is sufficient to satisfy customer demand.

We present a model that ensures that each customer has access to an open retailer. Furthermore, the probability that each retailer has sufficient capacity to satisfy customer demand is bounded from below. Additionally, we incorporate closest assignment and fixed assignment constraints to model customer demand.

Subsequently, we introduce techniques that can be used to decompose the parcel delivery network of PostNL into multiple smaller components without loss of optimality. This is a powerful tool that enables solution methods to operate much quicker. For our benchmark problem we are able to reduce a problem of 6216 binary variables to a problem of 5454 binary variables. Moreover, these are divided over 1307 subproblems of which 264 are non-trivial.

Next, we present two solution methods. The custom B&B algorithm is most interesting since it is able to find optimal solutions for the parcel delivery network of PostNL within reasonable time. The working principle is a repeated problem decomposition with a sophisticated lowerbound to enable pruning of the branches. However, the B&B algorithm gives guarantees on the solution quality it is computationally wise far slower compared to our TS heuristic that is able to find the optimal solution for 98.6% of the subproblems.

From an academic perspective we present methods that can be used to decompose a parcel delivery network and we develop an exact solution method for a large scale problem. From PostNL's perspective we upgrade their current decision support model in several ways. We provide: more local advice on where to open a new location, more detailed advice on the size of the new location, we consider customer preference in our model, we introduce the option to close a facility and we generate awareness for considering stochasticity into all of PostNL's algorithms.

1.1. PostNL's current algorithm

In this section we explain the goal and working principles of PostNL's current algorithm that is used to determine the required additional retailers and retail capacity.

The goal of PostNL's algorithm is to identify bottlenecks in the retail network and to identify which capacity deficits can be accounted for by traditional retailers (current and acquirable). These two goals are separated on a local and a nationwide level. However, the effects on the current network as a result of opening a new location are out of scope.

Locally, PostNL's algorithm uses the current network to determine which customers account for which capacity deficit at a retailer. For those retailers where the capacity deficit is larger than some threshold, the algorithm suggests to open a new retailer within each customer area. A requirement is that the customer area's fraction of total demand to the retailer with the capacity deficit is larger than a threshold fraction. Consequently, whenever x customers have their demand satisfied by retailer A, that has a capacity deficit larger than the threshold, then the algorithm suggests that in all x customer areas a new retailer should open. Whenever there is a capacity deficit at a retailer that does not meet the threshold, the algorithm suggests to consider allocating more space at the retailer or redistributing some of its volume whenever possible.

Nationwide, PostNL's algorithm identifies bottlenecks in the network. All retailers where the used capacity exceeds 90%, 110% and 150% of the total capacity are flagged red, purple and black respectively. This is then presented to management who use this information (blackbox) to come up with a budget for additional retailers.

Information on potential new retailers is not actively used to make decisions but these potential locations are shown to retail managers who can use this to quickly find acquirable retailers.

Both locally and nationwide, PostNL's current algorithm relies heavily on expert judgement for decision making and only cautiously makes suggestions. These suggestions are primarily supported by visualizations within a dashboard. However, to translate the suggestions in combination with the visualizations to actions, human experts are required.

1.2. Report structure

The structure of this report is as follows. First, we give a detailed problem description in Chapter 2. Next, we review some previous work on location problems and position our problem within it in Chapter 3. We follow in Chapter 4 with an overview of the data that we use and the assumptions that we make. In Chapter 5 we describe how we can decompose our problem and introduce an exact method and a heuristic to solve the decomposed problems. Later, in Chapter 6, we evaluate the performance of our decomposition method and our solution techniques. Finally, in Chapter 7 we draw conclusions from the results and give a short summary of the major results. Additionally, we indicate some future research directions.

2

Problem description

In this chapter we translate the objective and the requirements on the parcel delivery network of PostNL into a mathematical model. The question that we try to answer with our model is to identify where PostNL should open new retailers of what capacity to enable parcel consolidation for the upcoming five years in the Netherlands in a cost effective way.

Retailers can be opened at discrete locations $j \in M$. Whether a retailer at a given location is open, is signified by $z_j \in \mathbb{B}$ with $j \in M$. The parcel volume demand of Dutch citizens can be aggregated at different levels of detail. We use the term customer to specify an area $i \in N$ that is associated with the sum of demand of its residents.

We recognise two different types of residents in a customer area. The residents in $i \in N$ that prefer the closest location $j \in M$ and the residents in $i \in N$ that prefer a specific location $j \in M$. We assume that their preference remains the same whenever new retailers are opened. This means that the first group of residents selects a new retailer whenever it is closer than their current closest retailer. The residents that prefer a specific location are indifferent with regards to openings of new retailers. For the first group of residents we use the variables:

$$x_{ij} = \begin{cases} 1, & \text{If location } j \text{ is the closest open location for customer } i \\ 0, & \text{Otherwise} \end{cases} \qquad \forall i \in N, \forall j \in M \tag{2.1}$$

$\beta_{ij} \in \mathbb{R}$ indicates the fraction of residents in $i \in N$ that visit retailer $j \in M$ where j is not the closest open retailer. Therefore, we can define the fraction of residents in location i that select the closest retailer as:

$$1 - \sum_{j \in M} \beta_{ij} = \gamma_i \tag{2.2}$$

Such that:

$$\sum_{j \in M} \beta_{ij} + \gamma_i \sum_{j \in M} x_{ij} = 1 \quad \forall i \in N \tag{2.3}$$

All customer demand should be satisfied by open retailers. However, there are restrictions on the maximum distance, D_i, between customer $i \in N$ and retailer $j \in M$. Therefore, $x_{ij} = 0$ if $d_{ij} > D_i$ where d_{ij} signifies the distance between location i and j.

The demand volume, u_i, of customer $i \in N$ is forecasted. The forecast for u_i is normally distributed with $\mathcal{N}(\mu_i, \sigma_i^2)$. This means that demand for different customers is independent. PostNL wants to guarantee, with $\alpha\%$ certainty, that the capacity at each individual retailer is sufficient to satisfy their customer demand.

PostNL wishes to minimize costs. Annually, current and new retailers cost f_j^c and f_j^n irrespective of their capacity. Because of opening costs, $f_j^n > f_j^c$. Furthermore, to model acquisition efforts, we introduce an area dependent capacity penalty. The penalty decreases as the number of potential stores in area $j \in M$ that may be acquired as retailers increases. Function $g_j(\cdot)$ translates the fixed opening

costs and the capacity penalty into a single cost for additional capacity. Travelling costs are negligible with respect to these costs and can therefore be ignored.

The problem that we face shows close resemblance to the single-source, ordered capacitated facility location problem with stochastic demands. Where single-source signifies that each customer should be assigned to one and only one retailer. Ordered means that there is a distinct order in which customers select facilities, signifying customer preference. Capacitated means that the capacity of each facility is limited. Finally, stochastic demands is used to describe that demands are uncertain, but are assumed to be drawn from a known probability distribution.

Mathematical model P (Formulation 2.4-2.11) translates the objective and the requirements on the parcel delivery network to mathematical formulas. Equation 2.4 minimizes the cost of the required capacity c_j at location $j \in M$. Constraints 2.5 specify that all customer demand from residents with closest assignment preferences is satisfied by a retailer within the maximum distance limit represented by connectivity matrix a_{ij}. Constraints 2.6 specify that the capacity of a retailer should, with $\alpha\%$ certainty, be greater than or equal to the sum of its customer demand. Constraints 2.7 couple the variables c_j to the variables z_j. H is a large number which is minimally as large as the maximum required capacity at retailer j. Constraints 2.8 describe the order of assignment where each customer is assigned to the closest open retailer.[1] Whenever a retailer j is open ($z_j = 1$) and there is no retailer a closer to customer i than retailer j ($\sum_{a:d_{ia}<d_{ij}} z_a = 0$). Then, customer i should be assigned to retailer j ($x_{ij} = 1$). Finally, Constraints 2.9-2.11 provide the domains of the decision variables.

$$\text{P: min} \quad \sum_{j \in M} g_j(c_j) \tag{2.4}$$

$$\text{Subject to} \quad \sum_{j \in M} a_{ij} x_{ij} = 1 \qquad \forall i \in N \tag{2.5}$$

$$\mathbb{P}\Big(\sum_{i \in N} u_i(\beta_{ij} + \gamma_i x_{ij}) \leq c_j \Big) \geq \alpha_j \qquad \forall j \in M \tag{2.6}$$

$$c_j \leq H z_j \qquad \forall j \in M \tag{2.7}$$

$$x_{ij} + \sum_{a:d_{ia}<d_{ij}} z_a \geq z_j \qquad \forall i \in N,\ \forall j \in M \tag{2.8}$$

$$c_j \geq 0 \qquad \forall j \in M \tag{2.9}$$

$$z_j \in \{0,1\} \qquad \forall j \in M \tag{2.10}$$

$$x_{ij} \in \{0,1\} \qquad \forall i \in N,\ \forall j \in M \tag{2.11}$$

$$\tag{2.12}$$

The function $g_j(c_j)$ is represented by

$$g_j^1(c_j) = f_j^c \sum_{l=1}^{s_j} \mathbb{1}_{c_j > \sum_{k=1}^{l} c_j^{(k)}} \tag{2.13}$$

$$g_j^2(c_j) = f_j^n + \frac{f_j}{n_j + 1} c_j^2 \tag{2.14}$$

$$g_j(c_j) = g_j^1(c_j) + g_j^2(\max(c_j - \sum_{k=1}^{s_j} c_j^{(k)}, 0)) \tag{2.15}$$

where s_j is the number of already opened retailers at location j, $c_j^{(k)}$ is the capacity of the k^{th} largest retailer and n_j is the number of potential retailers at location j. f_j^c, f_j^n and f_j are costs associated with opening a retailer at location j. Equation 2.15 may seem difficult, but it is just a staircase function followed by a quadratic increasing cost of which the slope depends on the number of potential locations.

[1] This is not the tightest formulation, but it is explanatory. For the tightest formulation see [2].

See Figure 2.1 for an example. A quadratic increase is selected because it is preferred to have multiple small capacity deficits compared to a few major deficits.

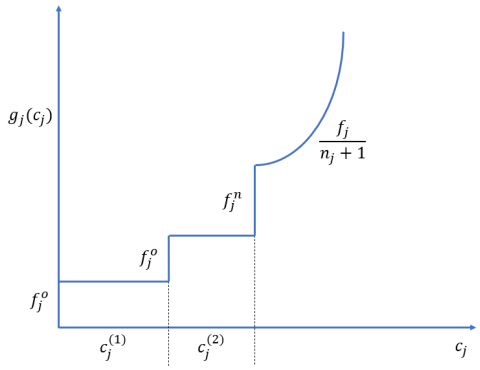


Figure 2.1: The cost function $g_j(c_j)$.

Because of Constraints 2.6 classical linear programming techniques do not apply. However, since u_i is normally distributed with $\mathcal{N}(\mu_i, \sigma_i^2)$, we can reformulate our Constraints 2.6 as Second Order Conic Programming (SOCP) constraints. This means that Constraints 2.6 are interchanged with Constraints 2.16. Basically, this means that a new random variable for the accumulated demand at retailer $j \in M$ is constructed as the sum of the individual demands. Since the individual demands are normally distributed this is straightforward.

$$\sum_{i \in N} \mu_{ui}(\beta_{ij} + \gamma_i x_{ij}) + q_{\alpha_j} \left\| \mathbf{\Sigma_u}^{1/2} (\boldsymbol{\beta_j} + \gamma_i \mathbf{x_j}) \right\|_2 \leq c_j \quad \forall j \in M \tag{2.16}$$

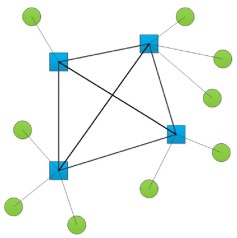
In Constraints 2.16 $\boldsymbol{\beta_j}$ and $\mathbf{x_j}$ represent a column of all β_{ij} and x_{ij} variables respectively for $i \in N$. $\mathbf{\Sigma_u}$ is the covariance matrix of the stochastic demands $\mathbf{u}$. Finally, q_{α_j} is the q centile of the standard normal distribution. Basically, we constructed a combined distribution of demand and specify that with $\alpha\%$ certainty this should be below the maximum capacity of a facility.

3

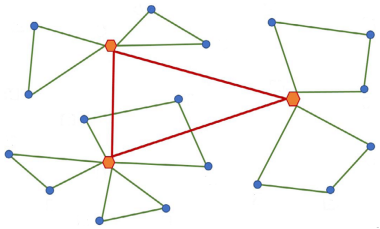
Literature review

In this chapter we present the current state-of-the-art knowledge with regards to our problem. Subsequently, we position our work within it.

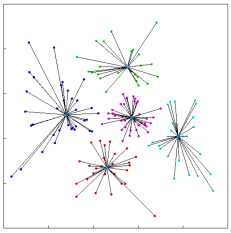
Three different problem families can be recognised in the field of location problems that are often applied to optimize parcel delivery. These are: The Hub Location Problem, Multi Depot Location Vehicle Routing Problem and the Facility Location Problem. The Hub Location Problem considers multiple stages of consolidation to exploit economies of scale. First, parcels are consolidated in small depots, these are then transported to a hub that consolidates parcels from multiple depots. Within the first hub parcels are sorted before being transported to another hub. Subsequently, the parcels are sorted and delivered to smaller depots before reaching a customer [3–6]. The transport in between hubs is often discounted to express the economies of scale [3, 6]. The Multi Depot Vehicle Routing Problem focuses on determining the locations of multiple depots, assigning customers to these depots and finding routes to visit these customers. The objective is to minimize the costs associated with opening a depot and travel cost to visit customers within a route starting and ending at a given depot [5, 7]. Finally, Facility Location Problems focus on determining the locations of facilities and to assign customers to a particular facility. The objective is to minimize the cost of opening a facility and the travel cost between customer and facility [8–10]. Contrary to the Multi Depot Location Vehicle Routing Problem, the travel costs in Facility Location Problems are solely based on the distance between the customer and its assigned facility irrespective of the tour to visit multiple customers consecutively. Figure 3.1 schematically shows these three problems.



(a) An illustration of the Hub Location Problem [11].



(b) An illustration of the Multi Depot Location Vehicle Routing Problem [12].



(c) An illustration of the Facility Location Problem [13].

Figure 3.1: Different location problems in the parcel distribution industry.

This report focuses on the Facility Location Problem family. Of the three discussed problems it has the least interaction between different customers. However, because of the size of the problem it might still be a challenging problem. In [14] it is shown that the Facility Location Problem is NP-hard. Within the family of Facility Location Problems we can distinguish between the Uncapacitated Facility Location Problem (UFLP) and the Capacitated Facility Location Problem (CFLP). The difference is that in the former facilities are assumed to have infinite capacity whereas in the latter facilities have a finite capacity [8].

The UFLP is applied to the parcel industry by [15] to determine the location of unmanned parcel lockers. An implicit result of the UFLP is that customers always select the closest facility. In [16, 17] the parcel locker selection is enforced with random utility theory. More specifically, a (threshold) multinomial logit model is used [18]. Problem instances up to 400 nodes could still be solved exactly with the aid of valid inequalities. Contrary to this research, lockers are assumed to have infinite capacity. Which eliminates the requirement for demand data. Therefore, we shift our focus to CFLPs.

CFLP is a well-studied combinatorial problem with a vast amount of literature. As for almost all combinatorial problems, exact solutions are preferable since they provide optimal results. However, for larger problems, exact solutions become inaccessible due to excessive computational effort. Therefore, for larger problems we only have access to satisfactory solutions determined by (meta-)heuristics. Besides the classical CFLP, some research is devoted to special cases with added restrictions.

Exact approaches to solve the CFLP mostly rely on B&B. Alternative formulations [19], dominance rules [20, 21], valid inequalities [22, 23] and bounding procedures [20, 21] are used to increase the problem sizes for which B&B remains a valid option.

However, the majority of the work on Facility Location Problems apply Lagrangian relaxations [24]. This may be because [25] shows that this approach is dominant for large instances with respect to other relaxations. Lagrangian relaxations can either be used as a subroutine of B&B to obtain lower bounds [26–30]. Or in combination with heuristics to obtain satisfactory results for large problems [31–40]. Lagrangian relaxation can either be applied to the capacity constraint to obtain an UFLP [37]. Or to the demand satisfaction constraints to obtain multiple knapsack problems [26–36, 38–40]. Both the UFLP and the knapsack problem are much easier to solve compared to the CFLP. Generally, the problems where the demand satisfaction constraints are relaxed are larger. A subroutine of Lagrangian relaxation is the updating procedure of the duals. This can either be the general subgradient optimisation [3, 27–32, 34, 35, 38–40], the volume algorithm [33, 41] or the r-algorithm [36, 42]. Additionally, in [28] an improved version of the subgradient optimisation method is introduced. By applying the volume algorithm and a random algorithm [33] is able to solve CFLPs with a problem size of 1000x1000 in an hour with gaps <1%. The random algorithm LP-relaxes the binary variables and the fractional result is used as a chance to determine if a facility should be open. Subsequently, a transportation problem is solved to determine the assignment of customers.

[43, 44] lay the foundations for heuristics applicable to the CFLP that do not involve Lagrangian relaxation. They propose a local search heuristic that consists of two phases: the ADD phase where facilities are opened one-by-one based on the greatest cost-reduction. Followed by a DROP and SHIFT phase where facilities are closed or moved to another location based on the greatest cost-reduction. In [44, 45] the reverse approach is considered where initially all facilities are open and the facility that maximally reduces the total cost is closed.

In [37] the local search techniques of [44] are complemented with a Lagrangian relaxation to solve the CFLP by repetitively solving the UFLP. The UFLP is solved with the dual ascend and dual adjustment procedure described in [46].

In [47] different metaheuristic solutions to various location problems are compared with each other. It concludes that Tabu Search (TS) dominates Genetic Algorithm (GA)s, Particle Swarm Optimization (PSO) and Scatter Search for UFLPs, CFLPs and HLPs up to 2500 demand nodes. This is concluded based on a comparison of benchmark problems introduced in [48]. TS algorithms applied to larger instances often use a random initial solution [47, 49–51]. Otherwise, an initial greedy solution is applied [10, 52–54]. Furthermore, the neighbourhood structure is often a 1-interchange or 2-swap neighbourhood [10, 47, 49–54]. 1-interchange stands for opening or closing a facility that was closed or open respectively. Whereas 2-swap signifies that one open facility is moved elsewhere [47]. Furthermore, different memory structures can be used to either intensify [10, 51, 54] or diversify [10, 51, 52, 54] the solution. Similar conclusions are drawn by [55] who implemented TS, Simulated Annealing (SA) and a GA for CFLPs up to 250 customers and 100 facilities. They show that TS is superior to SA and GA when comparing computation time required to obtain a given solution quality.

The classical Facility Location Problem can be extended in many different ways. The first applicable extension is the closest assignment constraint. For UFLP, this constraint is naturally satisfied since shorter distances reduce costs and each facility can be assigned to an infinite number of customers. However, for the CFLP the closest assignment constraint should be enforced. In [56] different closest

assignment constraints are theoretically compared with each other. It is concluded that the closest assignment constraint proposed in [2] is the tightest formulation. Furthermore, the proposed formulation handles ties and does not require a fixed number of facilities.

Whenever single-source constraints are introduced a customer should be assigned to a single facility [9, 32, 39, 40, 57–59]. Therefore, the number of binary variables increases substantially. The generalized assignment problem that becomes a subproblem of the CFLP is by itself an NP-hard problem [54, 60]. This makes single-source CFLP an NP-hard problem even when only feasible solutions are required [54]. To solve the single-source CFLP different techniques are applied such as Variable Neighbourhood Search (VNS) [9, 54, 61], Langrangian heuristics [32], GA [40, 57] and scatter search [62, 63]. Firstly, Variable Neighbourhood Search (VNS) is a meta heuristic that is only applied to single-source CFLPs [9, 54, 61] and not to ordinary CFLPs. This lead to the development of new neighbourhoods to assign customers to facilities. Basic neighbourhoods are moving a customer to another facility and swapping two customers [9, 54, 61]. However, also larger neighbourhoods are developed to allow for more diversification [9, 54]. Secondly, [32] applies a basic Lagrangian relaxation proceeded by a Multiple Ant Colony System to solve the single-source CFLP. The Lagrangian relaxation determines which facilities should be open and the Multiple Ant Colony System is used to determine which customers should be assigned to them. Afterwards, basic local search techniques are used to further improve the solutions. Thirdly, different GAs are applied to single-source CFLPs by [40, 57]. The gene representation that is used is such that each customer indicates by which facility it is served. This automatically creates a set of open facilities [40, 57]. Finally, ordinary GAs can be extended to obtain a scatter search, that can also be used to solve a single-source CFLP [47, 62, 63]. Compared to GA, Scatter Search is a more directed solution combination method [47, 62, 63]. It maintains a smaller reference set (population) but invests more time in local improvement and recombination. Sometimes the local search process involves a TS [62].

Other extensions to the classical CFLP are multi-plant, multi-commodity and multi-period (dynamic) additions. Multi-plant means that multiple facilities can be opened in the same location [35, 36, 64]. Multi-commodity means that there are multiple commodities [34]. Finally, multi-period means that the problem considers multiple periods [29, 38]. All of these extensions do not significantly change the problem structure or applicable solution methods. Multi-plant options are simply modeled by duplicating a location numerous times [35, 36] or by replacing the variables with a continuous decision on capacity [64]. Multiple commodities are introduced by duplicating the original assignment variables and related constraints [34]. Lastly, multiple periods are incorporated by duplicating every variable for each period [29, 38]. After these modifications ordinary solution methods to classical CFLPs can be used.

When it comes to uncertain demand, either stochastic programming or robust optimization is used. The former assumes that the distribution of the uncertain parameter is known a priori. Whereas the latter assumes that we only know the set to which our observations belong. Within stochastic programming, the focus is often to ensure that constraints hold with a certain probability and to optimize an expected value. Contrary to robust optimization, that is more focused on worst case performance [65–67]. The uncertainty set can be arbitrary shaped but often either discrete scenarios are considered [30, 58, 61, 68] or the box uncertainty set [66], or the budgeted uncertainty set [69], or the ellipsoidal uncertainty set [30, 39, 57, 66], or Poisson demand [39, 40, 53], or fuzzy random demand [64]. [66] concludes that the box uncertainty set is often too conservative with respect to the ellipsoidal uncertainty set. However, problems with ellipsoidal uncertainty become non-linear and require conic programming [66, 67]. Problems with box uncertainty and Poisson distributed demand can be converted to a deterministic problem without increasing the problem size [39, 40, 66, 69]. Examples of stochastic programming applied to CFLP can be found in [30, 64, 70], robust optimization can be found in [39, 40, 57, 66, 68, 69]. In [58, 61, 65] a hybrid between the two called p-robust optimization is used. P-robust optimization means that whenever optimization is applied to a given number of scenarios, the overall final solution should not be worse than p% of the solution considering a single scenario. This is because in stochastic programming the expectation is considered which could mean that some very favourable scenarios compensate for some unfavourable ones. Whereas in robust optimization one protects itself to these unfavourable scenarios but this might be too conservative. Additionally, optimization under uncertainty is sometimes integrated in a dynamic environment [58, 61, 64, 69, 70]. In a dynamic environment, decisions regarding facility openings are made with uncertainty. Afterwards, the uncertainty is removed and customers are assigned to these facilities (recourse decisions). Similarly to deterministic versions, exact approaches as described in [30, 58, 66, 68–70] only work for

small problems. Therefore, similar heuristics are introduced to cope with larger problems. Lagrangian heuristics [39, 40], GAs [40, 57], TS [53], VNS [61] and PSO [64].

Besides the Facility Location Problem, our problem also shows similarities with the Set Covering Problem (SCP). The SCP states that a complete set (sometimes called rows) should be covered with a number of subsets (sometimes called columns). Each subset is associated with a cost and the sum of subset costs should be minimized. Relating to our problem the set that should be covered are all customers and the subsets are represented by the retailers. However, in our case the costs of a subset cannot be calculated for each subset individually. Similar to the Facility Location Problem, we recognise different solution approaches to the classical NP-hard SCP [71, 72]. Exact approaches by means of a B&B scheme can be found in [73, 74]. Furthermore, Lagrangian relaxations are applied in [73, 75, 76]. Finally, heuristics are also available. The heuristics can be divided into population based heuristics [77–84] and local search heuristics [75, 76, 85–87].

Almost all B&B exact methods use the LP-relaxation as a lowerbound during the branching process. This is because stronger lower bounds are hard to obtain [72, 74]. Other lower bounds can be obtained with Lagrangian relaxation and subgradient optimization [74]

Common to many local search heuristics is that subsets are selected based on a score relative to the subset cost and the number of rows covered [72, 85–87]. [88] shows that the basic greedy score defined as the subset costs divided by number of rows covered gives the best results. Furthermore, heuristics often apply methods to discard redundant columns [72, 78, 82, 85–87].

There are many variants of population based heuristics. Classical GA are applied by [79–81]. We recognise two gene representations for the SCP. Either the binary version where each subset is represented by its binary variable [79, 80] or the row version where each row indicates a column by which it is covered [81]. The advantage of the latter is that it always produces feasible solutions. However, the binary representation is superior in terms of optimization speed even when incorporating a method to make solutions feasible [79]. Other population heuristics mainly use the binary representation but apply different methods for combining and diversifying the solution. With regards to population based heuristics we conclude based on the work of [78, 89] that binary black hole optimization is superior to cuckoo search, bee colony optimization, firefly optimization and electromagnetism like algorithms. Binary black hole optimization is very similar to PSO [90]. Furthermore, [82] shows that a jumping PSO is superior to the ACO algorithm of [83], randomized local search of [87] and the indirect GA of [84].

3.1. Synthesis

Based on the encountered problems in literature we can classify our problem as a: single-source, ordered, discrete soft capacitated Facility Location Problem with uncertainty.

Discrete, Single source CFLPs were earlier described in [9, 32, 39, 40, 57–59]. None of these did simultaneously incorporate multi-plant options as was done in [35, 36, 64]. CFLPs with uncertain demand are frequently investigated such as in [30, 39, 40, 57, 61, 64–66, 68–70]. Of these papers, the ellipsoidal uncertainty set, similar to our stochastic constraint, is used by [30, 39, 57, 66]. However, only [57] discusses discrete single-source constraints with ellipsoidal uncertain demand simultaneously. Soft capacity restrictions can be found in [91, 92] which specifies that capacity restrictions are placed in the objective function. The ordered constraints, as in the context of this report, have not been earlier described to our knowledge. Furthermore, we did not find any papers that considered multiple streams of demand with different behaviour. Whereas this has many real-life applications not necessarily limited to the parcel delivery industry. Therefore, this research aims to close this gap. On top of that, all CFLP studies that we found focus on greenfield studies. Whereas in reality often there is already a network in place that should be revised. However, current models could be generalized easily, none of them specifically mentions how.

Similar as in [19, 20, 20, 21, 21–23, 73, 74] we rely on B&B to find exact solutions. However, because of the problem size and complexity we are forced to customize the algorithm. Therefore, we look into valid inequalities and alternative bounding procedures just as in [20–23].

With regards to heuristics, TS is extensively applied in location problems because of its effectiveness [47, 55]. Within TS we apply a greedy initial solution [10, 52–54] with a 1-interchange neighbourhood [10, 47, 49–54]. Finally, we only use basic memory structures [10, 51, 54] that prove to be capable of sufficient intensification and diversification to obtain satisfactory results.

4

Data

In this chapter we elaborate on the data and the scope of the project. Furthermore, we fit our project into the framework of PostNL. Finally, we elaborate and justify some of the assumptions that we make.

4.1. Parcel flow

PostNL delivers over 800.000 parcels a day. All of these parcels visit minimally one sorting center and maximally one retailer. Within this project we focus on the parcels that visit a retailer. In Figure 4.1 we show the parcel flow that interferes with the capacity at a retailer. A filled arrow indicates that the flow is in scope of the project. A green flow signifies that customers control the flow, whereas blue flows are flows that PostNL regulates.

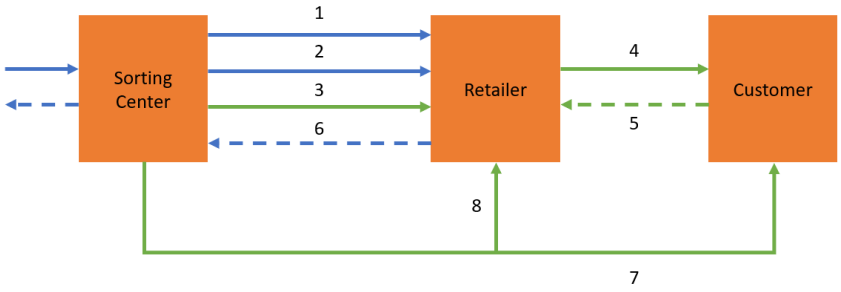


Figure 4.1: A schematic overview of the parcel flows that interfere with a retailer. Green flows indicate that the customer influences the flow, blue flows indicate that PostNL controls the flow, dashed arrows indicate the return stream of parcels that is out of scope of this report. [93]

In Figure 4.1, 8 flows can be distinguished. These flows signify the following:

1. Failed home deliveries
2. Signed parcels from Asia.
3. Customer indicated to deliver the parcel at a retailer
4. Pick-up of parcels within two weeks.
5. Incoming parcel flow.
6. Return flow to the sorting center.
7. Home deliveries
8. Customer rerouted the parcel to a retailer.

Figure 4.1 shows that this project focuses on the flow of parcels from a sorting center to the customer that pass a retailer. This flow of parcels is completely separated from the parcel flow from customer to sorting center. Furthermore, all outgoing flows from a retailer are assumed to remain proportional to the incoming flows.

4.2. Level of detail

In the Netherlands a unique address can by specified by a postal code, of 4 digits and 2 letters, and a house number. Where each extra character adds an extra level of detail. $PC4$ is a set of areas defined by the first four characters of a postal code, $PC5$ uses the first 5 characters and $PC6$ uses the entire postal code. Historically, a $PC5$ area corresponded with the coverage of one mailman. However, newly built neighbourhoods no longer adhere to this rule. Since all data available at PostNL is aggregated by postal codes we restrict ourselves to these different levels of detail as well.

In Table 4.1 the cardinality of each of the different aggregation levels is shown for The Netherlands and its capital Amsterdam. PostNL already has a (simple) tool to recommend locations for new retailers that considers the entire country on $PC4$ level. This tool is very much appreciated. However, for some areas, such as Amsterdam, $PC4$ is too crude and PostNL would like to investigate more local effects. Therefore, an optimization on $PC5$ level is preferred.

For some regions a solution on $PC5$ level does not offer much benefits. Therefore, to avoid making the problem unnecessarily large, we condsider a hybrid version where some regions are included at $PC4$ level and others at $PC5$. $PC6$ or household level is not preferred since their cardinality is too large for current available methods to solve in reasonable time and the extra benefit compared to $PC5$ level is expected to be small. Furthermore, these small areas are very susceptible to local changes which renders forecasts inaccurate.

We decided that $PC4$ areas with an area bigger than X and the number of inhabitants greater than Y are split up into $PC5$ locations. The area restriction is there because we assume that people in large $PC4$ locations do not have the same preferences based on their exact location. We add the number of inhabitants restriction to prevent that rural areas are divided.

Region or City	$\lvert Households \rvert$	$\lvert PC6 \rvert$	$\lvert PC5 \rvert$	$\lvert PC4 \rvert$
The Netherlands	8,630,593	455,140	33,210	4049
Amsterdam	442,748	16,711	1018	74

Table 4.1: Overview of the sizes of different detail levels in the Netherlands [93].

4.3. Parcel streams

PostNL recognizes two streams of parcels. A customer stream, where the customer is free to choose a retailer and a PostNL stream where PostNL selects a retailer for the customer. In Figure 4.2 these different parcel streams are indicated along with the assumed delivery location.

PostNL has the policy that parcels within the PostNL stream are delivered to the closest retailer with respect to the address of the recipient.

The customer stream of parcels is difficult to forecast since all customers have preferences that are unknown to PostNL. Therefore, we can only try to predict their behaviour based on experience. We assumed that there are two groups of people, those people who prefer a retailer close to their home. These could be people who have spare time during office hours, people who do not own a car, etc. And people who favor a specific location. These specific locations could be railway stations, office buildings, gas stations etc. Most likely these people work during office hours and are not frequently at home. We assume that over time and irrespective of opening a new facility, the balance between these groups remains constant. Furthermore, we assume that the group of people who favors a specific location continues to visit their preferred location with equal proportions. However, the volume of their demand can change over time.

The service policy for the PostNL stream and the assumed customer preferences have as a result that a new retailer at location j only attracts a fraction of demand of customer i if it is the closest retailer to customer i. Because only then, it attracts both the customers who favor a close retailer

and the PostNL stream. Furthermore, another implication is that a retailer that is currently a preferred location, is not allowed to close.

To exclude the situation that every currently open location is a preferred location we introduce a threshold. Only those locations where the demand of customers who favor a preferred location exceeds a certain fraction of the total demand at that location are considered preferred locations. Whenever, the fraction does not meet the threshold, all customers who have that location as their preferred location are modelled as customers who favor a close location. This assumption might be invalid for a small amount of customers, but we do not want that small groups of customers with uncommon preferences have a large impact on the new network.

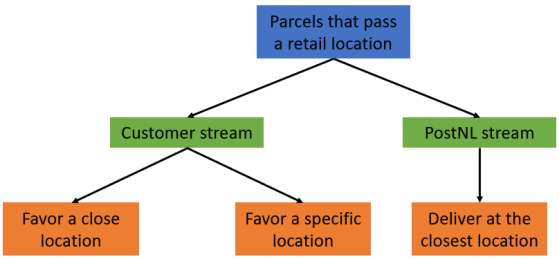


Figure 4.2: The different parcel streams and their delivery locations that are in scope of this project.

4.4. Distance and locations

We use euclidean distance as a distance measure. Because we mainly use distances to determine the closest location we assume that those are unaffected by the our distance measure. Furthermore, euclidean distance is easier to determine compared to a distance by road. To determine the location of an area, we use the center of addresses within an area.

4.5. Connectivity

As a service, PostNL desires to guarantee that all inhabitants in the Netherlands have access to retailer within close proximity of their home. However, close is a very relative word even for a country as small as the Netherlands. People living within the city have a different perception of close compared to people living in more remote areas. Therefore, we decided that everyone should have access to a retailer no further away than $p\%$ compared to the distance to the current closest retailer $\bar{D}_i$. On top of that, this maximum is bounded on the interval $[a, b]$. Mathematically, this is expressed as Equation 4.1. With the aid of D_i we can construct the connectivity matrix $\mathbf{a}$.

$$D_i = \min\left(b, \max\left(a, p\bar{D}_i\right)\right) \tag{4.1}$$

4.6. Potential facility locations

Potential facility locations are identified with a database of stores called Locatus. The number of Locatus locations determines the capacity penalty used in Equation 2.15.

4.7. Forecasts

PostNL forecasts the demand volume of the customer and the PostNL parcel stream with an Ordinary Least Squares (OLS) model. Each month the forecast is updated based on the newly acquired data. Because the forecast model is not 100% accurate we also consider the uncertainty. Unfortunately, because of some updates to the forecast model and the corona crisis only quarter 2019Q4 is available to evaluate the forecast error. A visualization of the forecast error is shown in Figure 4.3. We assume that the error is normally distributed.

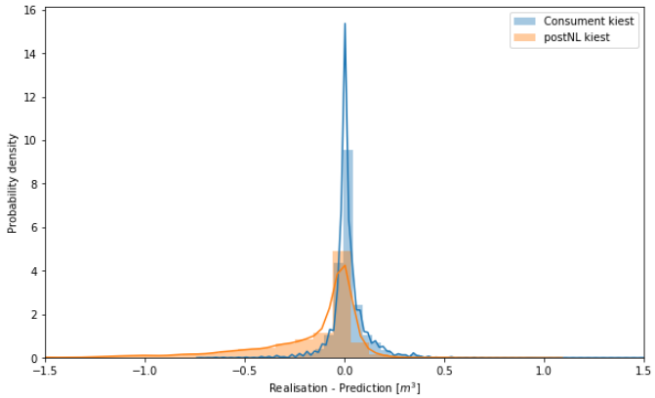


Figure 4.3: The prediction error in 2019Q4 for the two parcel streams.

5 Methodology

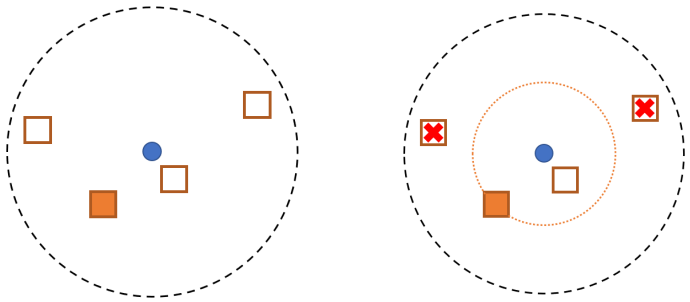
In this chapter we present our methods to solve the problem that is described in Chapter 2. First, we introduce methods to decompose the general problem into multiple smaller problems. Afterwards, we introduce a B&B algorithm and a TS heuristic to solve the decomposed problems.

5.1. Problem decomposition

Before starting the optimization procedure we investigate whether we can reduce the number of potential connections within the connectivity matrix. In Section 4.5 we indicated which customers are connected to which retailers. However, because of the problem structure we can reduce the number of potential retailers for each customer.

5.1.1. Reduce based on retailers that are always open

First, we can reduce the number of connections using the retailers that are always open. Some retailers are always open because they are either a preferred location for certain customers or they are the sole retailer that can satisfy the demand of some customers. All retailers further than the closest always open retailer for a specific customer will never satisfy the demand of that customer. Therefore, these connections can be removed. This process is illustrated in Figure 5.1.

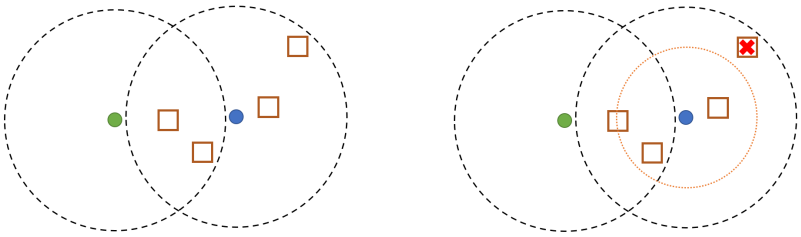


(a) Initial situation (b) New situation

Figure 5.1: The process of reducing the connectivity based on retailers that are always open. Always open retailers are indicated by filled squares, potentially open retailers are indicated by open squares, customers by circles and the maximum distance by the dashed black circle. Retailers with a cross will never satisfy demand of the blue customer.

5.1.2. Reduce based on superset of another customer

Moreover, the number of connections can be further reduced based on the following rule. If a customer is potentially served by a superset of the potential retailers of another customer (the subset), then all retailers further than the furthest retailer of this subset will never satisfy demand of the original customer. This is because one of the potential retailers will for sure be open to satisfy the demand of the other customer. This reduction process is illustrated in Figure 5.2.



(a) Initial situation (b) New situation

Figure 5.2: The process of reducing the connectivity based on subsets. Potentially open retailers are indicated by open squares, customers by circles and the maximum distance by the dashed black circles. Since the potential retailers of the green customer are a subset of those of the blue customer, all potential retailers of the blue customers further than the furthest in subset can be removed. Retailers with a cross will never satisfy demand of the blue customer.

5.1.3. Notation

Before we continue, we first introduce several new sets for ease of notation. Starting with M^o and M^c to identify the sets of open and closed retailers.

Next, set M_i defines the set of retailers that have a potential connection to customer $i \in N$.

Additionally, we introduce the set M_i^n to describe the neighbouring retailers of retailer $i \in M$. Two retailers are considered neighbours of each other whenever they can potentially satisfy the demand of at least one mutual customer.

Besides subsets of retailers, we also formulate customer subsets. First of all, N^d is the set of customers that do not have access to a retailer from the set M^o. Mathematically this is expressed as:

$$N^d = \{i \mid i \in N \wedge M_i \cap M^o = \varnothing\}$$

Additionally, we introduce three sets of customers of increasing size to define which customers visit a retailer under certain conditions.

N_j^-	The set of customers for whom retailer j is the absolute closest retailer.
N_j	The set of customers that visit retailer j.
N_j^+	The maximum sized set of customers that retailer j can satisfy.

Naturally, it follows that:

$$N_j^- \subseteq N_j \subseteq N_j^+ \quad \forall j \in M$$

Furthermore, we define the function $G_j(N_j)$ to express the costs at retailer j given that it is visited by the set of customers N_j. Basically, $G_j(N_j) = g\left(\sum_{i \in N_j} \mu_{ui}(\beta_{ij} + \gamma_i) + q_{\alpha_j} \left\|\boldsymbol{\Sigma}_{\mathbf{u}}^{1/2}(\boldsymbol{\beta}_{\mathbf{j}} + \gamma_i)\right\|_2\right)$ or in words, the costs associated to the total demand of all customers that visit retailer j equal $G_j(N_j)$. Since $g_j(c_j)$ is a monotonically increasing function, so is $G_j(N_j)$. Therefore, $G(S) \leq G(T)$ given that $S \subseteq T$. With the newly introduced sets and function, we express the minimum costs at retailer $j \in M$ as $G_j(N_j^-)$ and the maximum costs at retailer $j \in M$ as $G_j(N_j^+)$.

In all following proofs we will use a $\hat{}$ to express a given set in the optimal solution and a neutral letter for the current solution.

5.1.4. Close retailers by cost reduction

In this section we indicate how we identify retailers that are closed in an optimal solution. To determine which retailers are closed in an optimal solution we use the retailers that are always open to claim that there is another assignment of customers to these always open retailers that is feasible and cheaper. This is somewhat similar to the omega-reduction described in [20, 21].

To identify which retailers are closed in an optimal solution we first construct the directed graph $G(M, E)$ that indicates which closed retailers can reduce the costs at which open retailers. We put all

retailers on the nodes and an arc from node i to node j indicates that retailer i could potentially reduce the costs at retailer j. A requirement is that retailer $i \in M^c$ and $j \in M_i^n$. Meaning that retailer i is closed and that retailer i and j have at least one mutual potential customer.

For each single node $i \in M$ we create a subgraph $G_i(M, E)$. Subgraph $G_i(M, E)$ consists of all nodes and arcs from the initial graph $G(M, E)$ that can be reached from node i. We put all open retailers within $G_i(M, E)$ in the set M_i^o and all closed retailers within $G_i(M, E)$ in the set M_i^c. Therefore, M_i^o only contains retailers that could possibly reduce their costs whenever retailer i opens. Figure 5.3 shows an example of the directed graph $G(M, E)$ and its subgraph $G_A(M, E)$. By definition $M_A^o = \{X, Y, Z\}$ and $M_A^c = \{A, B, C, D\}$.

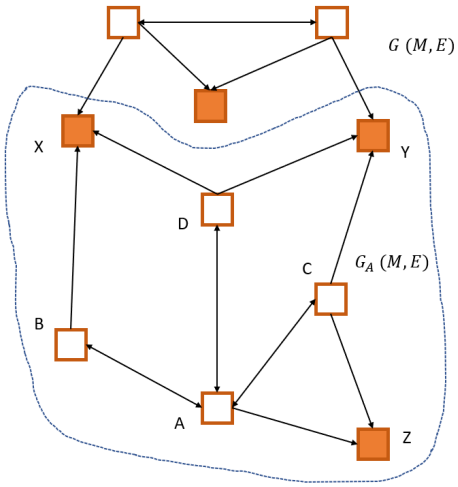


Figure 5.3: Directed graph $G(M, E)$ with retailers on the nodes and an arc from node i to node j indicates that retailer i could potentially reduce costs at retailer j. A requirement is that retailer i is closed and that retailers i and j potentially share a customer. Filled squares indicate always open retailers and open squares indicate currently closed retailers.

Next, we define a few more expressions. First, C^o are all combinations of open retailers in M_i^o. We express a unique combination in C^o with c^o. Second, $C^c(c^o)$ are all combinations of closed retailers in M_i^c such that the set $C^c(c^o)$ at least contains those closed retailers from M_i^c such that each open retailer in c^o is directly connected to one retailer in $C^c(c^o)$. Additionally, $C^c(c^o)$ contains the retailers on the path from and including retailer i to these direct neighbours of c^o. Consequently, for each combination c^o there are possibly multiple combinations $C^c(c^o)$ since the direct neighbours of c^o can vary and there are multiple paths from i to these direct neighbours. We need to include the paths because only whenever the retailers on these paths are opened the costs at the open retailers c^o possibly reduce because of opening retailer i. In other words, C^o contains combinations of retailers in M_i^o that can possibly reduce their cost. To possibly reduce the cost at a unique combination $c^o \in C^o$ there are multiple options of additional retailers that can be opened. All valid options that require that retailer i is open are the combinations in the set $C^c(c^o)$. In our example (see Figure 5.3), if $c^o = \{X, Z\}$ then $C^c(c^o) = \{\{A, B\}, \{A, D\}, \{A, B, D\}, \{A, C, D\}, \{A, B, C\}, \{A, B, C, D\}\}$.

With the new expressions we are ready to use Theorem 1 to determine which retailers are closed in an optimal solution.

Theorem 1 *In an optimal solution retailer A is closed if $N^d \cap \{N_j^+ \mid j \in M_A^c\} = \emptyset$ and*

$$\min_{c^c \in C^c(c^o)} \Big(\sum_{j \in c^c} G_j(N_j^-) \Big) + \sum_{i \in c^o} G_i(N_i^-) > \sum_{i \in c^o} G_i(N_i^+) \quad \forall c^o \in C^o$$

Where the first condition indicates that there are no disconnected customers that could potentially be satisfied by a retailer from the set M_A^c and the second condition indicates that there is a cheaper alternative set of retailers without retailer A. We proof Theorem 1 as follows: in an optimal solution where retailers in a set $c^c \in C^c(c^o)$ and retailers in the corresponding set c^o are concurrently open we

can express the total costs as:

$$\sum_{j\in c^c} G_j(N_j) + \sum_{j\in c^o} G_j(N_j) + \sum_{j\in M^o\setminus(c^c\cup c^o)} G_j(N_j)$$

Since $N_i^- \subseteq N_i \quad \forall i \in M$ and because $G_j(\cdot)$ is a monotonically increasing function it follows that

$$\sum_{j\in c^c} G_j(N_j) + \sum_{j\in c^o} G_j(N_j) + \sum_{j\in M^o\setminus(c^c\cup c^o)} G_j(N_j) \geq \sum_{j\in c^c} G_j(N_j^-) + \sum_{j\in c^o} G_j(N_j^-) + \sum_{j\in M^o\setminus(c^c\cup c^o)} G_j(N_j)$$

By definition of a minimum the following inequality holds

$$\sum_{j\in c^c} G_j(N_j^-) + \sum_{j\in c^o} G_j(N_j^-) + \sum_{j\in M^o\setminus(c^c\cup c^o)} G_j(N_j) \geq \min_{c^c\in C^c(c^o)} \Big(\sum_{j\in c^c} G_j(N_j^-) \Big) + \sum_{j\in c^o} G_j(N_j^-) + \sum_{j\in M^o\setminus(c^c\cup c^o)} G_j(N_j)$$

Furthermore, because of the condition that

$$\min_{c^c\in C^c(c^o)} \Big(\sum_{j\in c^c} G_j(N_j^-) \Big) + \sum_{j\in c^o} G_j(N_j^-) > \sum_{i\in c^o} G_i(N_i^+) \quad \forall c^o \in C^o$$

we show that:

$$\min_{c^c\in C^c(c^o)} \Big(\sum_{j\in c^c} G_j(N_j^-) \Big) + \sum_{j\in c^o} G_j(N_j^-) + \sum_{j\in M^o\setminus(c^c\cup c^o)} G_j(N_j) > \sum_{i\in c^o} G_i(N_i^+) + \sum_{j\in M^o\setminus(c^c\cup c^o)} G_j(N_j)$$

Because the condition concerns the minimum of $c^c \in C^c(c^o)$ the condition holds for all $c^c \in C^c(c^o)$. Therefore, there is no combination of retailers $c^c \in C^c(c^o)$ that possibly reduce the costs at retailers c^o. Furthermore, because of the condition that $N^d \cap \{N_j^+ \mid j \in M_A^c\} = \varnothing$ an alternative solution where retailers $i \in M_A^o$ are open is a feasible solution for the subgraph $G_A(M, E)$.

Since all of the above holds for all sets $c^o \in C^o$ and $c^c \in C^c(c^o)$. Where each set $c^c \in C^c(c^o)$ contains retailer A we conclude that retailer A can never reduce the costs at any retailer from the set M_A^o. On top of that, since M_A^o are the only retailers where retailer A could potentially reduce the costs. Retailer A can never reduce the costs at any retailer. Therefore, retailer A is closed in the optimal solution. This concludes our proof of Theorem 1 □.

Verifying Theorem 1

Using Theorem 1 is difficult since the number of combinations $|C^o|$ and $|C^c(c^o)|$ can become huge. Therefore, to reduce the number of combinations, we limit ourselves to those directed graphs $G_i(M, E)$ where if node i is removed each remaining component $G_i'(M, E)$ maximally contains one open retailer that is not a direct neighbour of node i in the graph $G_i(M, E)$. Each component $G_i'(M, E)$ can be considered individually since there are no paths between the different components.

The number of combinations reduces because if the inequality of Theorem 1 holds for the union of direct neighbours, it holds for all subsets of direct neighbours since

$$\sum_{i\in S} \Big(G_i(N_i^+) - G_i(N_i^-) \Big) < \sum_{i\in T} \Big(G_i(N_i^+) - G_i(N_i^-) \Big)$$

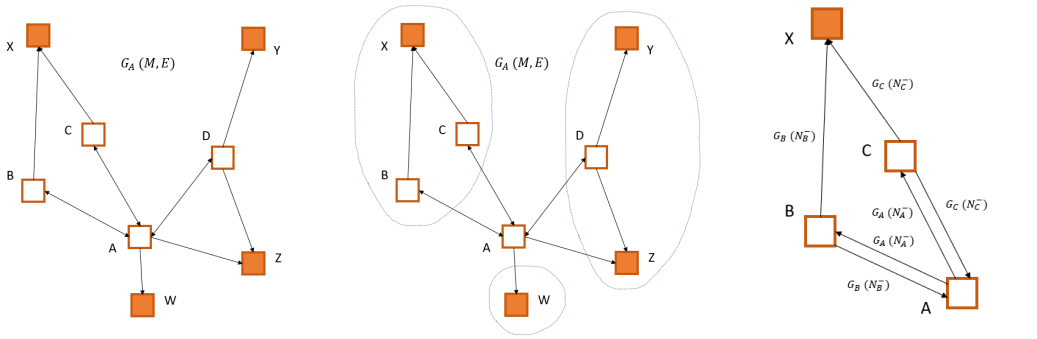
whenever $S \subset T$. Furthermore, $\min_{c^c\in C^c(c^o)} \Big(\sum_{j\in c^c} G_j(N_j^-) \Big) = G_i(N_i^-)$ since $\{i\}$ is the smallest set of retailers that neighbours all direct neighbours of retailer i. Therefore, we need one combination from C^o and one from $C^c(c^o)$ for the direct open neighbours of retailer i.

Additionally, whenever there is one additional open retailer in $G_i'(M, E)$ that is not a direct neighbour of retailer i there is one additional combination in C^o for component $G_i'(M, E)$. However, finding the minimum $\min_{c^c\in C^c(c^o)} \Big(\sum_{j\in c^c} G_j(N_j^-) \Big)$ remains easy. To find this minimum, we first include node i in component $G_i'(M, E)$. Furthermore, we put $G_j(N_j^-)$ on the outgoing edges of a node $j \in M$ in the directed subgraph $G_i'(M, E)$. Now, $\min_{c^c\in C^c(c^o)} \Big(\sum_{j\in c^c} G_j(N_j^-) \Big)$ equals the shortest path from retailer i to the additional open retailer that is not a direct neighbour of i. The shortest path problem can be

solved quickly ($O(E + M log M)$ [94]) with Dijkstra's Algorithm [95]. Whenever the shortest path length is longer than $\sum_{i \in c^o} \left(G_i(N_i^+) - G_i(N_i^-) \right)$ for each component $G'_i(M, E)$ the condition

$$\min_{c^c \in C^c(c^o)} \left(\sum_{j \in c^c} G_j(N_j^-) \right) + \sum_{i \in c^o} G_i(N_i^-) > \sum_{i \in c^o} G_i(N_i^+) \quad \forall c^o \in C^o$$

is satisfied for the general graph $G_i(M, E)$ as well. Therefore, in a directed graph $G_i(M, E)$ where there are no retailers that can satisfy the demand of any customers from N^d and where each component $G'_i(M, E)$ contains maximally one open retailer that is not a direct neighbour of i. We can quickly evaluate Theorem 1 since maximally $1 + |G'_i(M, E)|$ combinations should be checked. Where each check is executed in polynomial time.



(a) Subgraph $G_A(M, E)$ (b) Components $G'_A(M, E)$ (c) Shortest path problem

Figure 5.4: An example of a subgraph $G_A(M, E)$ for which we can quickly evaluate Theorem 1. Filled squares indicate open retailers, open squares indicate retailers with an unknown status. Dotted lines indicate the different components $G'_A(M, E)$.

Figure 5.4a shows a new example subgraph $G_A(M, E)$. Figure 5.4b shows the different components $G'_A(M, E)$. Because each component $G'_A(M, E)$ contains maximally one open retailer that is not a direct neighbour of A we can quickly evaluate Theorem 1 for subgraph $G_A(M, E)$. Based on what we previously indicated the combinations $c^o \in C^o$ that we evaluate are: $\{W, Z\}, \{W, Z, X\}, \{W, Z, Y\}$. The respective combinations $c^c \in C^c(c^o)$ that we should evaluate are $\{A\}$, the shortest path from A to X and the shortest path from A to Y. In Figure 5.4c we show the graph and edge costs that we use to find the shortest path from A to X.

5.1.5. Decompose problems

The above described processes are repeated until the connectivity matrix cannot be further reduced. The network associated with the resulting connectivity matrix can be decomposed into multiple subnetworks. Each subnetwork contains a combination of retailers and customers that solely interact with each other and not with retailers or customers of another subnetwork. Therefore, the decomposition can be performed without loss of optimality. A visualization of a decomposed network can be seen in Figure 5.5.

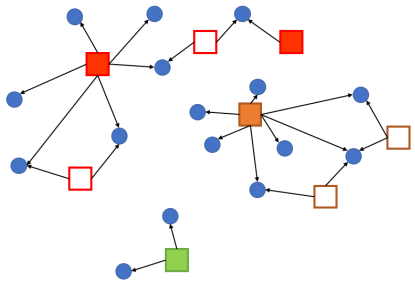


Figure 5.5: A decomposed network. Blue circles signify customers whereas squares signify retailers. Filled square indicate retailers that are always open whereas open squares indicate retailers that can be either open or closed.

Problem decomposition is effective for the PostNL case. We are able to close 762 retailers. Furthermore, we decompose our initial problem into 1307 subproblems of which 264 contain at least one retailer with an unfixed opening status. These 264 subproblems consider on average 20.7 retailers with an unfixed opening status. The degree of effectiveness is determined by the connectivity of the general problem.

5.2. Branch and Bound

For the PostNL case the above described reductions are sufficient to enable the usage of an exact method. We use Branch and Bound (B&B) with custom lower and upper bounds and a custom branching procedure. This is because our problem has a non-linear objective function and non-linear constraints. Furthermore, linear programming relaxations are inefficient because some of the constraints strongly rely on the integer structure. With a custom B&B procedure we can overcome these issues. In the following sections we discuss all facets of the B&B algorithm.

5.2.1. Solution representation

Each node in the branching tree is represented by a binary vector that encodes the locations with an open retailer. This automatically fixes the customer assignment to these open retailers based on the closest assignment constraints and the customer preference. Additionally, each solution is characterized by a lower and upper bound on the costs.

5.2.2. Root node

At the root node we solely fix the status of the retailers that are always open. From the root node onward, the number of opened retailers at each layer of the branching tree increases.

5.2.3. Upper bound

During calculation of the upper bound customers are assigned to the open retailers based on the closest assignment constraints. Therefore, the upper bound solution is a feasible assignment of customers to open retailers. For some problems there is already a feasible assignment at the root node. For others, more retailers should become fixed to obtain a feasible solution. Whenever there is no feasible assignment for all customers to any of the open retailers, we set the upper bound equal to positive infinity. Otherwise, the costs associated with the assignment of customers to the open retailers is used as an upper bound. For each problem it is guaranteed that there is a feasible solution. Mathematically the upper bound is expressed as:

$$UB = \begin{cases} \sum_{j \in M^o} G_j(N_j)) & \text{if } N^d = \varnothing \\ \infty, & \text{otherwise} \end{cases} \tag{5.1}$$

5.2.4. Lower bound

During the infeasible- and the feasible- phase we use different lower bounds. This is because the lower bound during the feasible phase is tighter compared to the lower bound during the infeasible phase.

Infeasible phase

During the infeasible phase the lower bound can be determined with a lower bound on the cost of the currently open retailers plus a lower bound on the additional cost for connecting each disconnected customer to a retailer.

We define the set L as the set that contains the smallest disjoint subsets of all retailers $M_i \quad \forall i \in N^d$ such that each set $M_i \quad \forall i \in N^d$ is a subset of a single element of L. Therefore, L can be expressed as:

$$L = \{(M_i \cup M_j \mid M_i \cap M_j \neq \varnothing, \quad \forall j \in N^d) \quad \forall i \in N^d\}$$

Each element in the set L is expressed as $\mathscr{L}$. With an example we clarify set L. Suppose there are three customers a, b and c that currently do not have access to an open retailer. Therefore, $N^d = \{a, b, c\}$. Furthermore, each of these customers have some potential retailers. Suppose these are $M_a, M_b, M_c = \{A, B\}, \{B, D\}, \{C, E\}$ respectively. Then L can be expressed as $L = \{\{A, B, D\}, \{C, E\}\}$.

Finally, we use $\hat{M}^o$ and $\hat{N}_i \quad \forall i \in \hat{M}^o$ to express the set of open retailers and the sets of customers that visit these open retailers in the optimal solution given that $M^o \subseteq \hat{M}^o$.

Now, we apply Theorem 2 to express a lower bound on the costs during the infeasible phase given the set of open retailers M^o as:

$$LB = \sum_{i \in M^o} G_i(N_i^-) + \sum_{\mathscr{L} \in L} \min_{l \in \mathscr{L}}(G_l(N_l^-)) \tag{5.2}$$

Theorem 2 *$\sum_{i \in M^o} G_i(N_i^-) + \sum_{\mathscr{L} \in L} \min_{l \in \mathscr{L}}(G_l(N_l^-))$ is a lower bound on problem P (Equations 2.4-2.11) given that $z_j = 1 \quad \forall j \in M^o$.*

We proof Theorem 2 based on the following arguments. First, since $\forall i \in M$ it holds that $N_i^- \subseteq \hat{N}_i$ and because $G_i(\cdot)$ is a monotonically increasing function it follows that $G_i(N_i^-) \leq G_i(\hat{N}_i) \; \forall i \in M$. Consequently, because $M^o \subseteq M$ it follows that

$$\sum_{i \in M^o} G_i(N_i^-) \leq \sum_{i \in M^o} G_i(\hat{N}_i)$$

In other words, a lower bound on the costs at the already open retailers is expressed as the minimal cost at the already open retailers.

Moreover, $N^d \neq \emptyset$ because we are in the infeasible phase. Consequently, because of the condition $z_j = 1 \quad \forall j \in M^o$ it follows that $M^o \subset \hat{M}^o$. Because, each customer in the set N^d should be connected to a retailer from its respective set M_i and because $\exists! \; (\mathscr{L} \in L) \supseteq M_i \quad \forall i \in N^d$ it holds that

$$(\hat{M}^o \setminus M^o) \cap \mathscr{L} \neq \emptyset \quad \forall \mathscr{L} \in L$$

In other words, since there exists one and only one set $\mathscr{L} \in L$ that is a superset of $M_i \quad \forall i \in N^d$ we know that at least one retailer from each set $\mathscr{L} \in L$ should be open with respect to the current set M^o in the optimal solution $\hat{M}^o$. Therefore, if we select the cheapest retailer in each set $\mathscr{L} \in L$ and use its minimal cost it follows that:

$$\sum_{\mathscr{L} \in L} \min_{l \in \mathscr{L}}(G_l(N_l^-)) \leq \sum_{i \in \hat{M}^o \setminus M^o} G_i(\hat{N}_i)$$

Consequently, the costs at both the currently open retailers (M^o) and the required additional retailers ($\hat{M}^o \setminus M^o$) are lower than or equal to the costs in the optimal solution. This concludes our proof because:

$$\sum_{i \in M^o} G_i(N_i^-) + \sum_{\mathscr{L} \in L} \min_{l \in \mathscr{L}}(G_l(N_l^-)) \leq \sum_{i \in \hat{M}^o} G_i(\hat{N}_i)$$

□

Feasible phase

The lower bound that is used during the infeasible phase is a lower bound for feasible solutions as well. However, since $N^d = \emptyset$ it loses some of its effectiveness. Therefore, we develop a tighter bound that is used during the feasible phase. The idea behind this lower bound is that the costs at the open retailers, M^o, cannot be lower than $G_i(N_i^-) \quad \forall i \in M^o$. However, to reduce the costs at a retailer from $G_i(N_i)$ to $G_i(N_i^-)$ additional retailers should be opened which incurs additional costs as well. For this lower bound we assume that all retailers $j \in M^c \cap M_i^n$ can reduce the costs at retailer i from $G_i(N_i)$ to $G_i(N_i^-)$ at the costs $G_j(N_j^-)$. Therefore, we express the lower bound given the current set M^o with problem LB:

$$LB\text{: min} \quad UB - \sum_{j\in M^c,\, i\in M^o} (G_i(N_i) - G_i(N_i^-))s_{ij} + \sum_{j\in M^c} G_j(N_j^-)z_j \tag{5.3}$$

$$\text{Subject to} \quad \sum_{i\in C_j} s_{ij} \leq |C_j| z_j \qquad \forall j \in M^c \tag{5.4}$$

$$\sum_{j\in M^c} s_{ij} \leq 1 \qquad \forall i \in M^o \tag{5.5}$$

$$\sum_{j\in M^c} z_j \leq n_{open} \tag{5.6}$$

$$s_{ij} = 0 \qquad \forall j \in M^c,\ \forall i \notin C_j \tag{5.7}$$

$$0 \leq s_{ij} \leq 1 \qquad \forall j \in M^c,\ \forall i \in M^o \tag{5.8}$$

$$z_j \in \{0,1\} \qquad \forall j \in M^c \tag{5.9}$$

The problem has two decision variables:

$$z_j = \begin{cases} 1, & \text{If a retailer at location } j \text{ is opened} \\ 0, & \text{Otherwise} \end{cases} \qquad \forall j \in M^c \tag{5.10}$$

$$s_{ij} = \begin{cases} 1, & \text{Retailer } i \text{ can reduce its costs} \\ 0, & \text{Otherwise} \end{cases} \qquad \forall j \in M^c,\ \forall i \in M^o \tag{5.11}$$

Objective function 5.3 minimizes the cost, given M^o, by maximizing the cost reduction at retailers $i \in M^o$ and minimizing the additional cost of opening new retailers $j \in M^c$. The cost reduction and additional costs of a retailer are determined in an optimistic way.

Constraints 5.4 ensure that opening retailer j can only reduce the costs at retailers, $i \in C_j$, with whom retailer j shares at least one potential customer. Therefore, $C_j = \{i \mid i \in M^o \cap M_j^n\}$. Constraints 5.5 ensure that each retailer $i \in M^o$ can only reduce its costs from $G_i(N_i)$ to $G_i(N_i^-)$ once.

The maximum number of retailers that can be opened, n_{open}, is determined with Equation 5.12. $\overline{UB}$ is the best known upper bound, $G_{i\in M^c}(N^-_{i\in M^c})^{(k)}$ is the k^{th} minimal cost of all retailers $i \in M^c$ and $\sum_{i\in M^o} G_i(N_i^-)$ signifies the minimal costs at all currently open retailers. Therefore, n_{open} is the maximal number of additional retailers that can be opened before the minimal costs at the open retailers plus the minimal additional costs exceed the global upper bound.

$$n_{open} = \max_n \left(n \ : \ \frac{\overline{UB} - \sum_{i\in M^o} G_i(N_i^-)}{\sum_{k=1}^{n} G_{i\in M^c}(N^-_{i\in M^c})^{(k)}} \geq 1 \right) \tag{5.12}$$

Therefore, Constraint 5.6 limits the number of retailers that can be opened.

Constraints 5.7 force that retailer $i \in M^o$ cannot reduce its costs by an opening of retailer $j \in M^c$ whenever they do not have any mutual potential customers. Finally, Constraints 5.8 and 5.9 specify the domains of z_j and s_{ij}.

Because of Theorem 3 we can use LB as a lower bound on the optimal solution given that $M^o \subseteq \hat{M}^o$.

Theorem 3 *$UB - \sum_{j\in M^c,\, i\in M^o}(G_i(N_i) - G_i(N_i^-))s_{ij} + \sum_{j\in M^c} G_j(N_j^-)z_j$ is a lower bound on problem P (Formulation 2.4-2.11) given that $z_j = 1 \quad \forall j \in M^o$ and Constraints 5.4- 5.9 are satisfied.*

We proof Theorem 3 based on the following arguments. We rewrite the objective function as

$$\min \sum_{i\in M^o} G_i(N_i)(1 - \sum_{j\in M^c} s_{ij}) + \sum_{i\in M^o} G_i(N_i^-) \sum_{j\in M^c} s_{ij} + \sum_{j\in M^c} G_j(N_j^-)z_j$$

This way of writing better illustrates that if $\sum_{j\in M^c} s_{ij} = 0$ the costs $G_i(N_i)$ are incurred for retailer $i \in M^o$ just as it is used in the upper bound. However, if $\sum_{j\in M^c} s_{ij} = 1$ the costs at retailer $i \in M^o$

reduce to $G_i(N_i^-)$. Because of constraints 5.4, 5.8 and 5.9, $\sum_{j \in M^c} s_{ij} = 1$ can only be true if $z_j = 1$.

Because $N_i^- \subseteq N_i$ and because $G_i(\cdot)$ is a monotonically increasing function it holds that $\sum_{i \in M^o} G_i(\hat{N}_i) \geq \sum_{i \in M^o} G_i(N_i^-)$.

Therefore,

$$\sum_{j \in M^c,\, i \in M^o} (G_i(N_i) - G_i(\hat{N}_i)) s_{ij} \leq \sum_{j \in M^c,\, i \in M^o} (G_i(N_i) - G_i(N_i^-)) s_{ij}$$

This means that the cost reduction at the already open retailers is smaller within the optimal solution compared to the lower bound solution. Moreover, by definition of an optimal solution and because of the condition that $M^o \subseteq \hat{M}^o$ and the previous statements it follows that

$$\sum_{i \in M^o} G_i(N_i) \geq \sum_{i \in \hat{M}^o} G_i(\hat{N}_i) \geq \sum_{i \in M^o} G_i(\hat{N}_i) \geq \sum_{i \in M^o} G_i(N_i^-)$$

Therefore,

$$\max_n \left(n \mid \frac{\overline{UB} - \sum_{i \in M^o} G_i(N_i^-)}{\sum_{k=1}^{n} G_{i \in M^c}(N_{i \in M^c}^-)^{(k)}} \geq 1 \right) \geq \max_n \left(n \mid \frac{\overline{UB} - \sum_{i \in \hat{M}^o} G_i(\hat{N}_i)}{\sum_{k=1}^{n} G_{i \in M^c}(N_{i \in M^c}^-)^{(k)}} \geq 1 \right)$$

Consequently, $n_{open} \geq \hat{n}_{open} = |\hat{M}^o \setminus M^o|$. So, within problem LB we allow for more additional retailers compared to the true number of additional retailers between the current and the optimal solution. Additionally, within problem LB we use assumption 1.

Assumption 1

$$d_{kj} < d_{ki} \quad \forall i \in M^o,\ \forall j \in M^c \cap M_i^n,\ \forall k \in N_i \setminus N_i^-$$

Assumption 1 invalidly assumes that each retailer from the set $M^c \cap M_i^n$ is closer to all customers from the set $N_i \setminus N_i^-$ for all retailers $i \in M^o$. Therefore, only a single retailer from the set $M^c \cap M_i^n$ is required to reduce the costs at retailer i from $G_i(N_i)$ to $G_i(N_i^-)$. Since assumption 1 is invalid for the general case the number of additional retailers required to reduce the costs from $G_i(N_i)$ to $G_i(\hat{N}_i)$ for any retailer is larger than or equal to the number of additional retailers required to reduce the costs from $G_i(N_i)$ to $G_i(N_i^-)$ under the assumptions of problem LB. On top of that, problem LB allows for more additional retailers because $n_{open} \geq \hat{n}_{open}$.

Moreover, since $G_j(N_j^-) \leq G_j(\hat{N}_j) \quad \forall j \in M^c$ the costs for each additional retailer within problem LB is lower than or equal to the true costs of each additional retailer.

Therefore, the additional costs for a cost reduction from $G_i(N_i)$ to $G_i(N_i^-)$ within problem LB for all retailers $i \in M^o$ are assumed smaller than or equal to the actual required additional costs in the optimal solution.

This concludes the proof since the cost reduction is assumed larger and the additional costs for cost reduction are assumed smaller within problem LB compared to the optimal solution. Therefore,

$$UB - \sum_{j \in M^c,\, i \in M^o} (G_i(N_i) - G_i(N_i^-)) s_{ij} + \sum_{j \in M^c} G_j(N_j^-) z_j \leq \sum_{i \in \hat{M}^o} G_i(\hat{N}_i)$$

□

Solving problem LB

Problem LB is a new difficult combinatorial problem. However, we can simplify problem LB with polynomial time preprocessing techniques that reduce the problem size. The preproccessing techniques apply Theorems 4-8. For the PostNL case these Theorems significantly reduce the problem size.

First of all, we use Theorem 4 to determine which open retailers to include, reducing the cardinality of M^o.

Theorem 4 $\sum_{j \in M^c} s_{Aj} = 0$ *if* $G_A(N_A) = G_A(N_A^-)$

Theorem 4 states that retailers where the current costs are equal to the minimal cost do not require a cost reduction. A proof of Theorem 4 is as follows. Suppose we have an optimal solution where $\sum_{j \in M^c} s_{Aj} = 1$ and $G_A(N_A) = G_A(N_A^-)$. Because of Constraints 5.4, one retailer from the set $M^c \cap M_A^n$ should be open. Suppose this is retailer B. Consequently, $z_B = 1$. Therefore, we can express the objective value as:

$$UB - \sum_{j \in M^c,\, i \in (M^o \setminus A)} (G_i(N_i) - G_i(N_i^-))s_{ij} + \sum_{j \in (M^c \setminus B)} G_j(N_j^-)z_j - (G_A(N_A) - G_A(N_A^-)) + G_B(N_B^-)$$

Because of the condition that $G_A(N_A) = G_A(N_A^-)$, the above is equal to:

$$UB - \sum_{j \in M^c,\, i \in (M^o \setminus A)} (G_i(N_i) - G_i(N_i^-))s_{ij} + \sum_{j \in (M^c \setminus B)} G_j(N_j^-)z_j + G_B(N_B^-)$$

In an alternative solution where, $\sum_{j \in M^c} s_{Aj} = 0$ we do not require that any retailer from the set M_A^n should be open. Therefore, $z_B = 0$ or $z_B = 1$ are both feasible. Therefore, the objective value is equal to

$$UB - \sum_{j \in M^c,\, i \in (M^o \setminus A)} (G_i(N_i) - G_i(N_i^-))s_{ij} + \sum_{j \in (M^c \setminus B)} G_j(N_j^-)z_j + G_B(N_B^-)z_B$$

Since z_B lies within the domain $\{0, 1\}$ it follows that:

$$UB - \sum_{j \in M^c,\, i \in (M^o \setminus A)} (G_i(N_i) - G_i(N_i^-))s_{ij} + \sum_{j \in (M^c \setminus B)} G_j(N_j^-)z_j + G_B(N_B^-)z_B \leq$$
$$UB - \sum_{j \in M^c,\, i \in (M^o \setminus A)} (G_i(N_i) - G_i(N_i^-))s_{ij} + \sum_{j \in (M^c \setminus B)} G_j(N_j^-)z_j + G_B(N_B^-)$$

This shows that there is a feasible alternative solution with a lower objective value than or equal objective value to the optimal objective value whenever $\sum_{j \in M^c} s_{Aj} = 1$. Therefore, an optimal solution exists with $\sum_{j \in M^c} s_{Aj} = 0$, given that $G_A(N_A) = G_A(N_A^-)$. This concludes our proof of Theorem 4 □.

Secondly, we use Theorem 5 to determine which closed retailers we include.

Theorem 5 $z_A = 0$ *if* $G_A(N_A^-) > G_B(N_B^-)$ *and* $C_A \subseteq C_B$

Closed retailer A is not part of the optimal solution if the minimum cost of retailer A is larger than the minimum cost of retailer B and retailer A permits a cost reduction at a subset of the open retailers where retailer B permits a cost reduction.

To proof Theorem 5 we suppose to have an optimal solution where $z_A = 1$, this permits that we obtain a cost reduction for at least all retailers in C_A. Therefore, the objective value can be expressed as:

$$UB - \sum_{j \in M^c,\, i \in (M^o \setminus C_A)} (G_i(N_i) - G_i(N_i^-))s_{ij} + \sum_{j \in (M^c \setminus A)} G_j(N_j^-)z_j - \sum_{i \in C_A} (G_i(N_i) - G_i(N_i^-)) + G_A(N_A^-)$$

An alternative solution where $z_A = 0$ and $z_B = 1$ permits that we obtain a cost reduction for at least all retailers in C_B. Moreover because $C_B \supseteq C_A$ the same solution also permits for a cost reduction at all retailers in C_A. Therefore, the objective value of an alternative solution with $z_A = 0$ and $z_B = 1$ can be expressed as:

$$UB - \sum_{j \in M^c,\, i \in (M^o \setminus C_B)} (G_i(N_i) - G_i(N_i^-))s_{ij} + \sum_{j \in (M^c \setminus \{A,B\})} G_j(N_j^-)z_j - \sum_{i \in C_A} (G_i(N_i) - G_i(N_i^-)) + G_B(N_B^-)$$

Whenever $z_A = 1$ is feasible, $z_A = 0$ and $z_B = 1$ is feasible as well. Furthermore, because of the condition $G_A(N_A^-) > G_B(N_B^-)$ it holds that

$$UB - \sum_{j \in M^c,\, i \in (M^o \setminus C_A)} (G_i(N_i) - G_i(N_i^-))s_{ij} + \sum_{j \in (M^c \setminus A)} G_j(N_j^-)z_j - \sum_{i \in C_A} (G_i(N_i) - G_i(N_i^-)) + G_A(N_A^-) >$$
$$UB - \sum_{j \in M^c,\, i \in (M^o \setminus C_B)} (G_i(N_i) - G_i(N_i^-))s_{ij} + \sum_{j \in (M^c \setminus \{A,B\})} G_j(N_j^-)z_j - \sum_{i \in C_A} (G_i(N_i) - G_i(N_i^-)) + G_B(N_B^-)$$

Therefore, we show that there is a feasible solution with $z_A = 0$ for which the objective value is strictly lower than the optimal objective value given that $z_A = 1$. Therefore, a solution with $z_A = 1$, given that $G_A(N_A^-) > G_B(N_B^-)$ and $C_A \subseteq C_B$, cannot be optimal. This concludes our proof □.

An extension of Theorem 5 is Theorem 6.

Theorem 6 $z_A + z_B \leq 1$ *if* $C_A = C_B$

If two closed retailers permit for a cost reduction at the same set of open retailers maximally one of the closed retailers is part of the optimal solution.

Suppose we have an optimal solution where $z_A = 1$ and $z_B = 1$, this permits that we obtain a cost reduction for at least all retailers in $C_A \cup C_B$. Because of the condition $C_A = C_B$, $C_A \cup C_B = C_A = C_B$. Therefore, the objective function is expressed as:

$$UB - \sum_{j \in M^c,\, i \in (M^o \setminus C_A)} (G_i(N_i) - G_i(N_i^-))s_{ij} + \sum_{j \in (M^c \setminus \{A,B\})} G_j(N_j^-)z_j - \sum_{i \in C_A} (G_i(N_i) - G_i(N_i^-)) + G_A(N_A^-) + G_B(N_B^-)$$

However, in an alternative solution where $z_A + z_B = 1$ together with the condition $C_A = C_B$ and Theorem 5 the objective value can be expressed as:

$$UB - \sum_{j \in M^c,\, i \in (M^o \setminus C_A)} (G_i(N_i) - G_i(N_i^-))s_{ij} + \sum_{j \in (M^c \setminus \{A,B\})} G_j(N_j^-)z_j - \sum_{i \in C_A} (G_i(N_i) - G_i(N_i^-)) + \min(G_A(N_A^-), G_B(N_B^-))$$

Because a solution with $z_A = 1$ and $z_B = 1$ is feasible, so is a solution with $z_A + z_B = 1$. Furthermore, since

$$UB - \sum_{j \in M^c,\, i \in (M^o \setminus C_A)} (G_i(N_i) - G_i(N_i^-))s_{ij} + \sum_{j \in (M^c \setminus \{A,B\})} G_j(N_j^-)z_j - \sum_{i \in C_A} (G_i(N_i) - G_i(N_i^-)) +$$
$$\min(G_A(N_A^-), G_B(N_B^-)) <$$
$$UB - \sum_{j \in M^c,\, i \in (M^o \setminus C_A)} (G_i(N_i) - G_i(N_i^-))s_{ij} + \sum_{j \in (M^c \setminus \{A,B\})} G_j(N_j^-)z_j - \sum_{i \in C_A} (G_i(N_i) - G_i(N_i^-)) +$$
$$G_A(N_A^-) + G_B(N_B^-)$$

We show that there is a feasible alternative solution with a strictly lower objective value than the objective value obtained whenever $z_A = z_B = 1$. Therefore, a solution with $z_A = z_B = 1$, given that $C_A = C_B$, cannot be optimal. This concludes our proof □.

Further simplifications can be obtained with Theorem 7 to reduce the number of included closed retailers:

Theorem 7 $z_A = 0$ *if* $G_A(N_A^-) - \sum_{i \in C_A} (G_i(N_i) - G_i(N_i^-)) \geq 0$

If the minimum costs of a closed retailer are larger than or equal to the maximal cost reduction that it could produce, then that retailer is not part of the optimal solution.

Suppose we have an optimal solution with $z_A = 1$. This allows that all $s_{iA} = 1 \quad \forall i \in C_A$. Therefore, the objective function can be written as:

$$UB - \sum_{j \in M^c,\, i \in (M^o \setminus C_A)} (G_i(N_i) - G_i(N_i^-))s_{ij} + \sum_{j \in (M^c \setminus A)} G_j(N_j^-)z_j + G_A(N_A^-) - \sum_{i \in C_A} (G_i(N_i) - G_i(N_i^-))$$

In an alternative feasible solution where $z_A = 0$ and $s_{ij} = 0 \quad \forall i \in C_A,\ \forall j \in M^c$, the objective function can be expressed as:

$$UB - \sum_{j \in M^c,\ i \in (M^o \setminus C_A)} (G_i(N_i) - G_i(N_i^-))s_{ij} + \sum_{j \in (M^c \setminus A)} G_j(N_j^-)z_j$$

Since $G_A(N_A^-) - \sum_{i \in C_A}(G_i(N_i) - G_i(N_i^-)) \geq 0$ it is obvious that:

$$UB - \sum_{j \in M^c,\ i \in (M^o \setminus C_A)} (G_i(N_i) - G_i(N_i^-))s_{ij} + \sum_{j \in (M^c \setminus A)} G_j(N_j^-)z_j + G_A(N_A^-) - \sum_{i \in C_A} (G_i(N_i) - G_i(N_i^-)) \geq$$
$$UB - \sum_{j \in M^c,\ i \in (M^o \setminus C_A)} (G_i(N_i) - G_i(N_i^-))s_{ij} + \sum_{j \in (M^c \setminus A)} G_j(N_j^-)z_j$$

This means that there is an alternative feasible solution where the objective value with $z_A = 0$ is smaller than or equal to the optimal objective value given that $z_A = 1$. Therefore, a solution with $z_A = 1$ cannot be optimal if $G_A(N_A^-) - \sum_{i \in C_A}(G_i(N_i) - G_i(N_i^-)) \geq 0$. □

Finally, we use Theorem 8 to determine which retailers are open in an optimal solution. First, we introduce the set $M^s = \{i | \exists! C_j \supseteq i \quad \forall i, j\}$ to define the retailers that can only reduce their cost by an opening of a single retailer. Second, we introduce k_j as the shorthand notation for the ranked position of maximal improvement of the upper bound that retailer $j \in M^c$ could account for. The maximal improvement of the upper bound that retailer j could account for is equal to $G_j(N_j^-) - \sum_{i \in C_j}(G_i(N_i) - G_i(N_i^-))$. So the retailer j that accounts for the largest improvement has $k_j = 1$.

Theorem 8 *$z_A = 1$ if $C_A \subseteq M^s$ and $k_A \leq n_{open}$*

Retailer A is open if it only permits for a cost reduction at open retailers that cannot reduce their costs otherwise and the improvement of the upper bound that retailer A accounts for is within the top n_{open}.

To proof Theorem 8 we suppose to have an optimal solution where $z_A = 0$ but $C_A \subseteq M^s$ and $k_A \leq n_{open}$. Furthermore, to avoid a trivial solution suppose that $|M^c| > n_{open}$. We introduce the set M^o_{LB} that contains all retailers in problem LB where $z_j = 1 \quad \forall j \in M^c$. Obviously, $A \notin M^o_{LB}$. However, suppose that $z_B = 1$. Consequently, retailer $B \in M^o_{LB}$. Therefore, the objective value of the optimal solution with $z_A = 0$ and $z_B = 1$ can be expressed as:

$$UB + \sum_{j \in M^o_{LB} \setminus B} G_j(N_j^-) - \sum_{j \in M^c,\ i \in (M^o \setminus C_B)} (G_i(N_i) - G_i(N_i^-))s_{ij} + G_B(N_B^-) - \sum_{i \in C_B} (G_i(N_i) - G_i(N_i^-))$$

Because of the condition that $k_A \leq n_{open}$ there is at least one retailer $j \in M^o_{LB}$ where $k_j > n_{open} \geq k_A$. Suppose this is retailer $B \in M^o_{LB}$. Therefore, it holds that:

$$G_B(N_B^-) - \sum_{i \in C_B} (G_i(N_i) - G_i(N_i^-)) > G_A(N_A^-) - \sum_{i \in C_A} (G_i(N_i) - G_i(N_i^-))$$

Suppose that in an alternative feasible solution we open retailer A and close retailer B, $z_A = 1$ and $z_B = 0$. Because of the condition that $C_A \subseteq M^s$ all $s_{iA} = 1 \quad \forall i \in C_A$. Therefore, the objective value of the alternative feasible solution can be expressed as:

$$UB + \sum_{j \in M^o_{LB} \setminus B} G_j(N_j^-) - \sum_{j \in M^c,\ i \in (M^o \setminus C_B)} (G_i(N_i) - G_i(N_i^-))s_{ij} + G_A(N_A^-) - \sum_{i \in C_A} (G_i(N_i) - G_i(N_i^-))$$

Based on the previous inequality it follows that:

$$UB + \sum_{j \in M^o_{LB} \setminus B} G_j(N_j^-) - \sum_{j \in M^c,\ i \in (M^o \setminus C_B)} (G_i(N_i) - G_i(N_i^-))s_{ij} + G_B(N_B^-) - \sum_{i \in C_B} (G_i(N_i) - G_i(N_i^-)) >$$
$$UB + \sum_{j \in M^o_{LB} \setminus B} G_j(N_j^-) - \sum_{j \in M^c,\ i \in (M^o \setminus C_B)} (G_i(N_i) - G_i(N_i^-))s_{ij} + G_A(N_A^-) - \sum_{i \in C_A} (G_i(N_i) - G_i(N_i^-))$$

Therefore, we show that there is a feasible solution where $z_A = 1$ with a strictly lower objective value than a supposed optimal solution where $z_A = 0$. Therefore, a solution with $z_A = 0$ cannot be optimal if $C_A \subseteq M^s$ and $k_A \leq n_{open}$. This concludes our proof of Theorem 8 □.

Theorems 5-8 reduce the cardinality of the set M^c within problem LB. Furthermore, the cardinality of the set M^o may be reduced as well with Theorem 4 and because we can remove retailer $i \in M^o$ from problem LB if $\nexists C_j \supseteq i \quad \forall j$. Or in words, if there is no closed retailer in problem LB that allows for a cost reduction at retailer $i \in M^o$.

The effectiveness of applying the introduced Theorems 4-8 ranges from no effect to a reduction of problem LB to a polynomial time problem. In Tables 5.2a and 5.2b we show problem instances where $M^o = \{X, Y, Z\}$ and $M^c = \{A, B, C\}$. A 1 indicates that there is a mutual customer between two retailers. Furthermore, we show the potential cost reduction and minimal cost of open and closed retailers respectively. Finally, $n_{open} = 2$ in both problem instances. In Table 5.2a we show an example of a problem instance where no reduction is available. Whereas the problem instance in Table 5.2b can be solved in polynomial time by applying Theorems 5 and 8.

	X	Y	Z	$G_j(N_j^-)$
A	1	1	0	1
B	0	1	1	2
C	1	0	1	3
$G_i(N_i) - G_i(N_i^-)$	1	2	3	$n_{open} = 2$

(a) A problem instance where no reduction is available

	X	Y	Z	$G_j(N_j^-)$
A	1	1	0	1
B	0	1	0	2
C	0	0	1	2
$G_i(N_i) - G_i(N_i^-)$	1	2	3	$n_{open} = 2$

(b) A problem instance that can be solved in polynomial time

Table 5.1: Problem instances where $M^o = \{X, Y, Z\}$ and $M^c = \{A, B, C\}$. A 1 indicates that there is a mutual customer between two retailers. Furthermore, we show the potential cost reduction and minimal cost of open and closed retailers respectively. Finally, $n_{open} = 2$ in both problem instances.

If Theorems 4-8 are applied in the order of introduction the computational effort is generally the smallest. Furthermore, Theorem 8 should always be applied last for maximum effectiveness.

With Theorems 4-8 we can generally reduce the number of variables in problem LB significantly for the PostNL case. Approximately 55% of the problems LB can be solved in polynomial time by applying Theorems 4-8. For the remaining 45% the problem size significantly reduces.

Additionally, to accelerate the solution process of problem LB, each B&B tree stores the solution to problem LB at the root node. Parts of this solution can be used by other nodes as an initial solution for problem LB to reduce the computational effort.

5.2.5. Branching rule

As a branching rule we remove one retailer from the set M^c by either opening it or closing it definitively. With a definitive closing of a retailer we aim to decompose our general problem into multiple smaller problems.

To identify which retailers should be closed definitively to decompose the general problem into multiple smaller problems we first construct a network $G(M, A)$. Each node in the network represents a retailer. Two retailers $i, j \quad \forall i, j \in M$ are connected with an undirected arc if $j \in M_i^n$. The goal is to find the minimal set of retailers in the set $M^c \subseteq M$ such that the network $G(M, A)$ disconnects. We focus on the minimal set of retailers because this requires minimal fixing effort. Because we are uninterested in retailers from the set M^o we remove them from the network $G(M, A)$ to create the equivalent network $G'(M^c, A)$. An undirected arc is drawn between retailers $i, j \quad \forall i, j \in M^c$ if $j \in M_i^n \vee (k \in M_i^n \wedge k \in M_j^n \quad k \in M^o)$. Or in words, whenever two retailers are neighbours of each other (they share a potential customer) or both retailers are neighbours of the same open retailer we draw an arc between them. Furthermore, to keep the decomposition effective, we remove all retailers from $G'(M^c, A)$ with only a single outgoing and incoming arc. Repeating this procedure removes tails from the network. We exclude tails because it is unlikely that separating a tail significantly reduces the number of solutions.

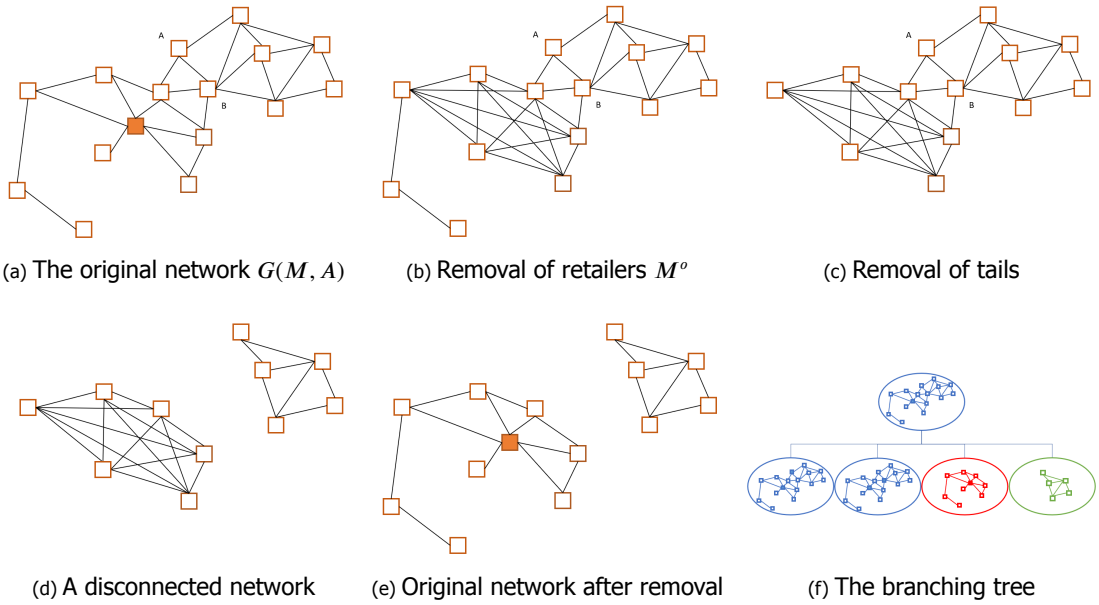
We solve multiple max-flow problems to identify the minimal sized set of nodes, called the cutset, to remove from the network in order to disconnect the network. We use Algorithm 11 from [96, 97] to determine which max-flow problems to solve. Each max-flow problem itself is solved by Edmonds and Karp's algorithm [98]. The orders of these two algorithms are respectively $O(M - \delta + 1 + \frac{\delta(\delta-1)}{2})$ and $O(ME^2)$ where δ is the minimal degree of all nodes in the network [96–99]. We select Edmonds and Karp's algorithm because it is intuitive and easy to implement.

Upon identifying the cutset, we are able to branch from our node in the B&B tree. First, we construct one branch for each retailer in the cutset where we open each retailer from the cutset individually. Additionally we create one branch where we close the retailers from the cutset definitively. By definition this creates at least two subproblems. We call this branch a fork because it roots two new trees from our initial tree. An additional restriction is that each new branch should be unique otherwise it can be skipped. A new tree is unique whenever any of the sets M, M^o or N is unique. Finally, we only create opening branches for retailers from the set M^b, where M^b is the set of retailers that upon opening will not cause the minimal costs to exceed the global upper bound.

$$M^b = \{i | i \in M^c \wedge \sum_{j \in M^o} G_j(N_j^-) + G_i(N_i^-) \leq \overline{UB}\}$$

On top of that, since all retailers $\bar{M}^b = \{i | i \in M^c \wedge \sum_{j \in M^o} G_j(N_j^-) + G_i(N_i^-) > \overline{UB}\}$ never open, we can remove them. As more retailers are removed, $G_j(N_j^-) \quad \forall j \in M$ increases. Therefore, we repeatedly remove all retailers from the set $\bar{M}^b$ until $\bar{M}^b = \emptyset$.

We clarify our branching rule with an example. Suppose that the network in Figure 5.6a is $G(M, A)$. Then by removal of the retailers M^o we obtain the network $G'(M^c, A)$ as shown in Figure 5.6b. Subsequently, removing tails from the network results in the network shown in Figure 5.6c. Now we identify $\{A, B\}$ as the cutset that disconnects the network as shown in Figure 5.6d. After we include the nodes that we previously removed to find the cutset we obtain the disconnected networks shown in Figure 5.6e. In Figure 5.6f we show the branching tree. The initial network is shown in blue on top of the tree. Opening both retailers A and B individually leads to the left two branches. These are also colored blue as they consider the same network as the source node. Furthermore, the two remaining subnetworks after removal are shown in red and green as they signify new trees.



(a) The original network $G(M, A)$ (b) Removal of retailers M^o (c) Removal of tails

(d) A disconnected network (e) Original network after removal (f) The branching tree

Figure 5.6: Finding a cutset to branch a source node. Filled squares represent open retailers and open squares represent retailers with an undefined opening status.

Each time a retailer is opened, the number of connections in the connectivity matrix reduces based on the rules defined in Section 5.1. Therefore, sometimes the cutset could also have size zero. Naturally, this will only root the new trees and will not fix any retailers open during a new branching routine. To

save some time we only apply the reduction rules of Sections 5.1.1 and 5.1.2. Finally, in case the network $G'(M^c, A)$ consists of a single tail (this is unlikely for the PostNL case). We do not root any new trees but instead define the cutset as M^c. Therefore, all retailers in $M^b \subseteq M^c$ are opened individually.

5.2.6. Node selection

To select the branching node we apply a best first search strategy since this quickly increases the lowest lower bound. This leads to quicker pruning by bound of other branches. In case of a tie, the lowest upper bound is used as well.

Since one problem can be composed of multiple subproblems, node selection is a tedious process. We explain this process with a fictitious branching tree in Figure 5.7. Underneath each node there are either other nodes (circles) or 'forks' (triangles). Forks indicate that the problem is an addition of its subproblems. The subproblems are the newly rooted trees described in the previous section.

To select a node, we first consider the main tree that is composed of all blue elements. Nodes that are dashed outlined cannot be branched any further and are therefore not considered. From all blue elements we select the best one. Whenever this is an ordinary node, the branching rules described in Section 5.2.5 can be applied directly. Whenever the best element is a fork, we need to determine which of its subproblems is most likely to decrease the incumbent solution. This likeliness for a single subproblem is defined as the difference between the lower and upper bound of the subproblem. Once we turned either left or right, the same process repeats itself until we arrive at a single node.

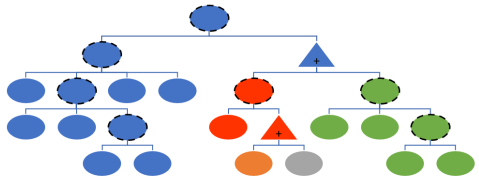


Figure 5.7: A fictitious branching tree. Similar colors indicate that the elements treat the same problem. Circles are nodes whereas triangles are an addition of two nodes. Dashed outlined nodes cannot be branched anymore because all branches are already created.

5.2.7. Pruning

One of the key concepts why ordinary B&B is effective for general problems is "pruning" certain branches prematurely [100]. Within our custom tailored B&B algorithm we apply two pruning strategies.

First, we prune by infeasibility. Whenever a newly rooted tree contains disconnected customers but no potential retailers to connect these customers, the tree is infeasible. Mathematically, a problem is feasible whenever $M^i \cap M \neq \emptyset \quad \forall i \in N$. Whenever a tree is infeasible, the fork from which it rooted is infeasible as well. Therefore, we can prune the fork.

Second, we prune by bound. A branch can be pruned by bound whenever the lower bound is greater than or equal to its tree's upper bound. For ordinary nodes the lower bound can be determined as described in Section 5.2.4. The lower bound of a fork is the sum of the lower bounds of its underlying trees. The lower bound of a tree is equal to the lowest lower bound among all of its active elements. Similarly, the upper bound of a node can be obtained with Equation 5.1. The upper bound of a fork is the sum of the upper bounds of its underlying trees. The upper bound of a tree is equal to the lowest upper bound among all elements that were ever active. Because of the relationship between nodes, forks and trees, each time a new node is constructed we work our way upwards to update the lower and upper bounds of the parent elements until we either reach the original tree or one ancestor element does not update.

5.2.8. Initial solution

To speed up the branching process we supply an initial solution. However, because of the way that our branching trees grow an initial solution is not as powerful as it is in a classical B&B algorithm. Contrary to classical B&B, our B&B algorithm is solely able to fix variables $z_j \quad \forall j \in M$ from 0 to 1 and not the other way around. Consequently, at the root node we started with $z_j = 1$ for as few retailers as possible to include all possible solutions. Therefore, an initial solution where additional retailers

are open cannot be used as a starting point for the B&B algorithm without excluding some solutions. However, we can use the objective value of the initial solution as an upper bound to enable quicker pruning of certain branches. Furthermore, having a lower upper bound increases the lower bound as well. This is because n_{open} in Equation 5.12 decreases whenever $\overline{UB} - \sum_{i \in M^o} G_i(N_i^-)$ decreases, where M^o is unaffected by the initial solution. Therefore, the effect of the initial solution is twofold: lower upper bounds and higher lower bounds. Both effects lead to quicker pruning by bound. However, the effect of an initial solution within our B&B algorithm for our case study is marginal as most time is invested in verifying that the incumbent solution is optimal. Therefore, we do not want to invest a big computational effort into constructing an initial solution. Therefore, we decided to use the objective value of the first feasible solution at every newly rooted tree.

Constructing the initial feasible solution

We use the current network of PostNL as an initial starting solution. However, we discovered that some customers do not have access to a retailer. Meaning that $N^d \neq \emptyset$. Therefore, we choose to open the retailer from the set M^c that connects most disconnected customers. This process is repeated until all disconnected customers are connected. The selection process is visualized in Figure 5.8.

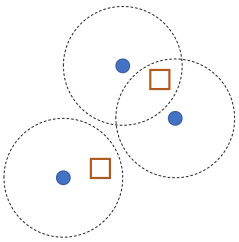


Figure 5.8: We satisfy the distance requirement for the most customers (blue circles) simultaneously by opening one closed retailer (orange square). In this example this means opening the upper right retailer.

5.3. Tabu Search

Because problem P (formulation 2.4-2.11) is NP-hard [14, 54] and the number of locations $|N|$ and $|M|$ is very large, the B&B algorithm is relatively slow. Furthermore, approximation algorithms such as the one defined in [101] cannot be applied because we do not have a Poisson distributed demand and customer allocation follows strict rules. Therefore, next to the exact solution we develop a heuristic solution. We apply TS because of the superiority over other heuristics for similar large-scale problems [47, 55]. Extensively applied Lagrangian heuristics are not selected since these methods are not as flexible as TS when it comes to slightly changing problem formulations. Furthermore, [102] indicates that Lagrangian heuristics are inferior to TS when capacity and single source constraints are included. The TS heuristic is used to find candidate solutions for the binary decision variables $\boldsymbol{z}$. Once $\boldsymbol{z}$ is known, the customers can be assigned to the open retailers easily.

5.3.1. Initial solution

We use the same initial solution for the TS heuristic as we used for the B&B algorithm (see Section 5.2.8). Contrary to the B&B algorithm the TS heuristic is able to build new solutions from the initial solution.

5.3.2. Local search

We improve the obtained feasible initial solution with basic local search techniques. Local search executes moves within a neighbourhood. Moves are executed based on a first improvement strategy. Each neighbourhood is explored in a randomized order. First improvement is used because it is assumed that different areas do not interfere with each other. Therefore, evaluation time can be reduced by first improvement.

The first neighbourhood is the DROP neighbourhood. DROP means that an open retailer is closed. Only retailers that solely satisfy demand of customers that are assigned based on closest assignment constraints are eligible to be closed. The DROP neighbourhood is an $O(N)$ size neighbourhood that can be searched as follows: First, identify the customers that are served by the to be closed retailer.

Next, determine which retailer these customers will select after closing. For each of these new retailers update the required capacity and the costs. Finally, compare the additional costs associated with the rerouted customers and the costs of the closed retailer. This means that the evaluation of the DROP neighbourhood is $O(N^2M)$.

The second neighbourhood is the ADD neighbourhood. ADD means that a closed retailer is opened. All closed retailers are eligible to be opened. Therefore, similarly as the DROP neighbourhood the ADD neighbourhood is of size $O(N)$ but the evaluation of the neighbourhood is $O(N^2M)$. However, for the network of PostNL searching the ADD neighbourhood takes longer than the DROP neighbourhood since a cost effective network contains more closed than open retailers. The ADD neighbourhood can be searched similar to the DROP neighbourhood. First, identify the customers that would be attracted upon opening. For each of these customers subtract their demand from their current retailer and add it to the opened retailer. For all retailers where the demand changed, update the required capacity and the costs. Finally, compare the cost savings by rerouting customers with the opening costs of the new retailer.

Each iteration of local search starts with a search in the DROP neighbourhood followed by a search in the ADD neighbourhood if there is no improvement in the DROP neighbourhood. This is because the DROP neighbourhood is generally smaller than the ADD neighbourhood for the PostNL case. Therefore, computationally cheap improvements are discovered quickly.

5.3.3. Tabu tenure

To escape local minima we allow for worsening moves during the local search as well. To prevent that the algorithm starts to cycle we make use of a Tabu List. Status reversals of retailers involved in a move during the local search are placed on the Tabu List. This means that moves in the DROP or the ADD neighbourhood add one item to the Tabu List. The number of iterations, $t(\tau)$, that a status reversal remains Tabu is a function of the change in objective value τ that is obtained by changing the status of a retailer. Function $t(\tau)$ is a monotonically decreasing function. We use a sigmoid function (Equation 5.13, Figure 5.9) to achieve this behaviour. Parameters ϕ and ω are used as scale and shift parameters respectively.

$$t(\tau) = T_{min} + T_{max} - \frac{T_{max}}{1 + e^{-\phi(\tau+\omega)}} \tag{5.13}$$

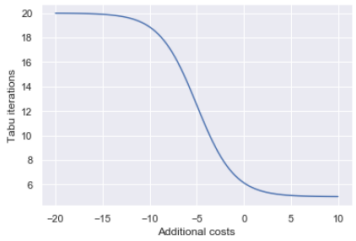


Figure 5.9: A sigmoid function where $T_{min} = 3$, $T_{max} = 10$, $\phi = 1/2$ and $\omega = 5$

Aspiration criteria

Aspiration criteria are used to overrule the Tabu List. We use the most basic aspiration criterion that says that whenever a Tabu move results in a better objective value compared to the incumbent objective value it is always executed. Thereby we give permission to overrule the Tabu List.

5.3.4. Termination criteria

As a termination criterion we use a maximum number of iterations without any improvement in the objective function.

5.4. Integration of the branch and bound algorithm with first descend

Preliminary results show that the TS heuristic is very competitive with the B&B algorithm for the PostNL case because it almost always finds the optimal solution far earlier than the B&B algorithm. However, there are no guarantees on the solution quality. Nevertheless, we want to exploit this observation to accelerate the B&B algorithm. Furthermore, we find that for the PostNL case only a few additional locations have to be opened. To make use of these observations in the B&B algorithm we use a different initial solution and additional branching rules.

The updated initial solution is the result of updating the conventional initial solution (Section 5.2.8) with the local search procedure without Tabu Tenure of Section 5.3.2. Without Tabu Tenure, the local search is equivalent to a first descent search.

To enable usage in the B&B algorithm we label each retailer that is open in the minimal situation as 'fixed', M^f, and all open retailers as 'open', M^o. The set M^o is determined with the updated initial solution. Obviously, $M^f \subseteq M^o$. We replace our conventional expression of the lower bound LB with a function $LB(\cdot)$ where the dot represents a set of retailers. Therefore, we can now calculate two lower bounds: $LB(M^f)$ and $LB(M^o)$. The sole difference is the retailers that we consider open. Therefore, $LB(M^f) \leq LB(M^o) \leq UB$ for all nodes. $LB(M^f)$ is the lower bound that is used to prune branches whenever possible and can thus be used at all places where we used LB before. Therefore, we can prune a branch whenever $LB(M^f) \geq \overline{UB}$. $LB(M^o)$ is solely used later to exclude a specific branching technique.

Selecting a branching node remains the same within our new B&B algorithm where we now use $LB(M^f)$ instead of LB.

After selecting a branch node, we determine which branching rule we use. Whenever $M^o = M^f$ we apply our ordinary branching rule introduced in Section 5.2.5. Otherwise, we use a different branch rule that we call 'close and fix'. First, we create branches where we definitively close one retailer from the set $M^o \setminus M^f$. Second, if $LB(M^o) < \overline{UB}$, we create a branch where we fix all retailers in M^o such that $M^o = M^f$. Figure 5.10 shows an example of a branching tree without pruning where we solely apply 'close and fix', while supposing that $LB(M^o) < \overline{UB}$, to a toy example.

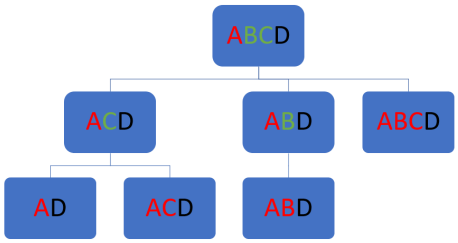


Figure 5.10: An example with 4 retailers for the close and fix branching rule. Red letters represent the set M^f, green and red letters represent the set M^o. Closing a retailer removes one retailer from the set $M^o \setminus M^f$. Fixing retailers open results in $M^o = M^f$.

What we achieve with this updated initial solution and additional branching rule is that we realise quicker that a specific retailer in the set $M^o \setminus M^f$ is crucial in the optimal solution and without it we cannot find any other solutions where $LB(M^f) < \overline{UB}$. For completeness, we also consider the possibility that $M^o \subset \hat{M}^o$. However, this branch is not pruned quicker compared to the original situation.

The initial solution is used at every newly rooted tree. Therefore, each tree benefits from the improved initial solution which increases the overall speed. Because preliminary experiments show that the integration of B&B with first descend is superior to the original B&B algorithm in terms of computational time, we discard the original B&B algorithm and continue with the B&B with first descend algorithm. The B&B with first descend algorithm is the algorithm that we refer to when speaking of the B&B algorithm. Furthermore, we also tried to include the whole TS heuristic as an initial solution. However, this does not accelerate the computations. Therefore, we discard this option.

6

Experiments

In this Chapter we introduce the experiments that we use to evaluate the quality of the model in combination with the solution techniques. We first introduce the benchmark problems and performance indicators that we use for evaluation. Afterwards, we evaluate the effectiveness of the problem decomposition techniques described in Section 5.1. Thereafter, we evaluate the efficiency of Theorems 3-8 to reduce the problem size of problem LB (Formulation 5.3 - 5.9). Subsequently, we perform several different experiments. First, to compare the B&B solution with the TS solution. Afterwards, we compare the solutions obtained with current model of PostNL with the newly obtained solutions. Later, we specifically investigate the added value of the chance constraint compared to PostNL's assumption. Subsequently, we evaluate the sensitivity to division between PC4 and PC5 areas. Additionally, we investigate the sensitivity to the cost parameters. Moreover, we investigate the intended effect of the quadratic capacity penalty. Finally, we perform one final analysis where we apply our model and solution technique not only to the benchmark problem but to the complete PostNL network for multiple periods. All results are obtained with a machine with 8 GB RAM and an Intel i7-7700HQ 2.8 GHz CPU. We use Python 3.7 as our programming language. Furthermore, we only use open source packages.

6.1. Benchmark problems and performance indicators

In this section we introduce the benchmark problems and performance indicators.

6.1.1. Benchmark problem

We choose to use the nationwide problem as our benchmark problem because it is difficult to identify a subset of the Netherlands that reflects the Netherlands as a whole within this problem context. As a reference time period we use June 2020. This is because this is the first month for which all required data is available at PostNL. Within our benchmark problem we choose to optimize for 2022Q4. Because this is a time period with large customer demand volumes that can be predicted with sufficient accuracy. Large demand volumes are desired as we expect that different methods become better distinguishable.

6.1.2. Performance indicators

As performance indicators we use the objective value, the number of open retailers, the size of the new retailers and the computational time. PostNL would like to open as few retailers as possible since each opening results in additional costs and effort. Additionally, we use statistics on the size of the open retailers to compare different methods. This is because it is hard for PostNL to acquire retailers of very small or very large volumes. Lastly, shorter computational times are desired but not very important to PostNL as long as the algorithm can finish overnight. However, from an academic perspective it is relevant to see differences in computation time since it tells something about the scale ability and the complexity of the problem.

6.2. Problem decomposition results for the PostNL case

To evaluate the effectiveness of the problem decomposition techniques introduced in Section 2. We evaluate how many clusters can be created for the benchmark problem, how many retailers can be closed because of Theorem 1 and how many clusters are trivial. The benchmark problem started with 6216 retailers that could possibly open, with Theorem 1 this is reduced to 5454 retailers. Furthermore, problem decomposition created 1307 clusters of which 264 are non-trivial. A trivial cluster is a cluster where the status of all retailers is either fixed open or fixed closed. Consequently, a non-trivial cluster is a cluster that contains retailers that could potentially open. In Table 6.1 we show statistics on the number of retailers within a non-trivial cluster that could potentially open.

Metric	
mean	20.7
std	43.1
min	1
25% quantile	3
median	14.5
75% quantile	22
max	621

Table 6.1: Statistics on the number of retailers within a non-trivial cluster that could potentially open.

Besides the PostNL case data we generate 50 additional random datasets. Where we draw random values from a kernel density estimate distribution for each input data column individually. This results in 5992 (26) retailers of which we can close 206 (13) retailers because of Theorem 1. The general problem can be decomposed into 558 (11) subproblems of which 142 (2) are non-trivial. The size of each non-trivial cluster is 23 (0.6) retailers. All values are averages with their standard error in between brackets. The PostNL data and the random data show different results because the random data is no longer logical. Barriers such as rivers and roads are ignored. Furthermore, high demand areas do not correspond to large current capacities.

6.3. Efficiency of Theorems 3-8 for the PostNL case

In Section 5.2.4 we introduced the NP-hard problem LB (Formulation 5.3-5.9). Moreover, we introduced Theorems 3-8 to reduce the number of variables in problem LB. In Table 6.2 we show the effectiveness of these theorems. We sample 5% of the problems LB for a subset of clusters for which we determine how many variables are removed. Each column in Table 5.2.4 should be read individually.

	# initial variables	# variabls after Theorems 3-8	# removed variables	Relative problem reduction
mean	115	5	110	0.95
std	146	8	141	0.09
min	2	0	2	0.50
25%	24	0	22	0.93
50%	60	0	56	1.00
75%	156	8	149	1.00
max	1095	70	1075	1.00

Table 6.2: Statistics on the efficiency of Theorems 3-8 to reduce the size of problem LB (Formulation 5.3-5.9). Each column should be read individually.

Theorems 3-8 are very efficient for the PostNL case since 55% of the problems LB within the PostNL case are solved to optimality in polynomial time by the pre-processing techniques. The other problem instances that cannot be solved in polynomial time significantly reduce in size as well.

6.4. Comparison Tabu Search and Branch and Bound

We compare the TS heuristic and the B&B algorithm based on the average optimality gap and the required computational time. Furthermore, we indicate the number of subproblems where the optimality gap is zero.

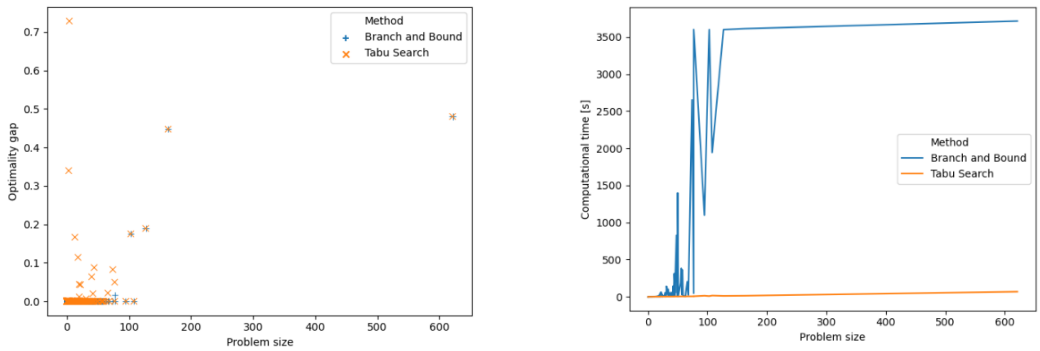
For the B&B algorithm we set a time limit of 3600 seconds. Therefore, not all subproblems are solved to optimality. In Table 6.3 we show the results obtained with the B&B algorithm and th TS heuristic regarding the optimality gap and the computational time for all subproblemns and for the non-trivial subproblems specifically.

	All problems		Non-trivial problems	
	Branch and Bound	Tabu Search	Branch and Bound	Tabu Search
# clusters	1307	1307	264	264
# solved to opt	1302	1289	259	246
objective	9412	9443	5619	5650
opt. gap	4.58%	4.89%	7.67%	8.18%
comp. time [s]	30266	557	30213	542

Table 6.3: Comparison of the B&B exact method with the TS heuristic. For the overall problem and the non-trivial problems specifically.

In Figure 6.1a we show the effect of the problem size on the optimality gap for the B&B algorithm and the TS heuristic. This shows that clusters larger than 100 retailers are difficult to solve to optimality with the B&B algorithm. Furthermore, there are a few smaller clusters for which the TS heuristic is unable to find the optimal solution.

The results on computational time per subproblem are shown in Figure 6.1b. It can be seen that the TS heuristic scales almost linearly with problem size. Whereas this relation is weaker for the B&B algorithm. This is because also the number of connections and the cost information determine the computational time.



(a) The problem size vs optimality gap

(b) The problem size vs computational time

Figure 6.1: Results on the problem size versus the optimality gap and the computational time for the B&B algorithm and the TS heuristic. The B&B algorithm is applied with a time limit of 3600 seconds.

Table 6.4 shows the number of open retailers and the required additional capacity at the open retailers for the B&B algorithm and the TS heuristic. Furthermore, Table 6.4 shows the number of currently open retailers that can be closed.

	B&B	TS
count	655	659
mean	0.92	0.91
std	0.88	0.87
min	0.00	0.00
25%	0.30	0.30
50%	0.67	0.67
75%	1.23	1.24
max	5.90	5.90
close	688	693

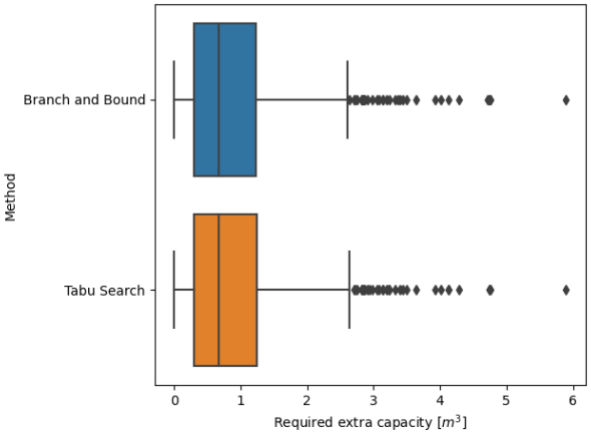


Table 6.4: A table and its boxplot with statistics on the required extra capacity at retailers given that the required extra capacity is larger than 0. Furthermore, the number of open retailers and the number of retailers that can be closed is indicated.

6.5. Comparison of PostNL's algorithm with the newly developed algorithms

Comparing PostNL's algorithm with the newly developed algorithms is more difficult compared to comparing the TS heuristic with the B&B algorithm. This is because PostNL's algorithm does not perform an optimization at all nor does it satisfy all constraints from the business. Furthermore, the underlying assumptions on customer preference are different. Therefore, this comparison is primarily used to show the added value of the new model with respect to the current model.

It is expected that PostNL's algorithm suggests to open much more smaller sized retailers compared to the models introduced within this report. Because PostNL's algorithm runs within a cloud environment with more computational power compared to the machine that is used for the other experiments, we cannot make a fair comparison on computational time. Additionally, we are currently unable to tell how much time of the PostNL's experts is saved by the new solution.

Table 6.5 shows the required additional capacity at the open retailers according to the B&B algorithm, the TS heuristic and PostNL's algorithm. For B&B and TS algorithms these results are different from the results in Table 6.4 because we aggregated the results to $PC4$ areas since this corresponds to PostNL's algorithm. Furthermore, Table 6.5 shows how many of the currently open retailers can be closed. As expected, the number of retailers that should be opened is far larger when applying PostNL's algorithm. Moreover, the size of each new retailer is much smaller. On top of that, the new models also found many retailers that could be closed.

	B&B	TS	PostNL
count	629	632	787
mean	0.95	0.95	0.38
std	0.92	0.92	0.44
min	0.00	0.00	0.00
25%	0.31	0.31	0.05
50%	0.69	0.69	0.23
75%	1.28	1.28	0.57
max	5.90	5.90	3.18
close	688	693	0

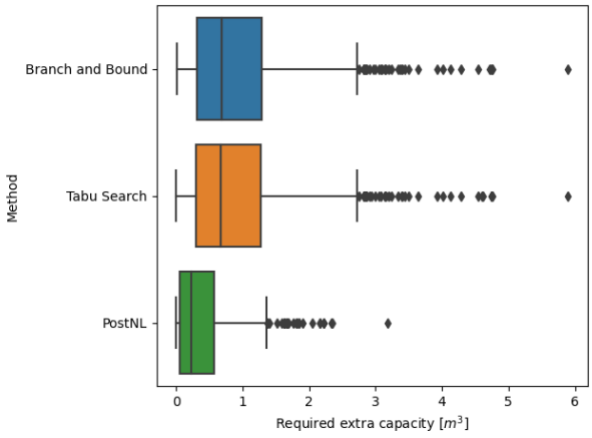


Table 6.5: A table and its boxplot with statistics on the required extra capacity at retailers given that the required extra capacity is larger than 0. Furthermore, the number of open retailers and the number of retailers that can be closed is indicated.

Because the TS heuristic and the B&B algorithm provide almost equal results and because the TS heuristic is much faster, the remainder of our experiments is performed with the TS heuristic. We expect that the conclusions that we draw based on these experiments hold for the B&B algorithm as well.

6.6. Added value of stochastic constraint

This experiment focuses purely on the added value of the stochastic constraint. We obtain results with our TS heuristic twice. Once where $\alpha_j = 0.95 \quad \forall j \in M$ and once where $\alpha_j = 0.5 \quad \forall j \in M$. Whenever $\alpha_j = 0.5 \quad \forall j \in M$, $q_{\alpha_j} = 0$ we purely rely on the expected customer demand and ignore the uncertainty. With $\alpha_j = 0.95$ and $\alpha_j = 0.5 \quad \forall j \in M$ we generate two different solutions for the required capacity $\mathbf{c}_j^1$ and $\mathbf{c}_j^2$ respectively for all retailers $j \in M$. Additionally, we determine a third solution $\mathbf{c}_j^{2'} = 1\frac{1}{9}\mathbf{c}_j^2$ for the capacity at each retailer $j \in M$. To obtain solution $\mathbf{c}_j^{2'}$ we assume that the aggregated demand at a retailer is equal to 90 % of the required capacity. This adds some robustness. Summarizing, there are three solutions for the required capacities at retailers $j \in M$. These are $\mathbf{c}_j^1$, $\mathbf{c}_j^2$ and $\mathbf{c}_j^{2'}$. We refer to these solutions with Solution 1, Solution 2 and Solution 2' respectively. For each of these solutions we calculate the probability that the capacity is sufficient to satisfy customer demand.

	Solution 1	Solution 2	Solution 2'
count	2332	2332	2332
mean	0.95	0.50	0.89
std	0.00	0.00	0.12
min	0.95	0.50	0.50
25%	0.95	0.50	0.81
50%	0.95	0.50	0.95
75%	0.95	0.50	1.00
max	0.95	0.50	1.00
obj	9443	8222	9699

Table 6.6: Statistics on the probability that customer demand can be satisfied by the opened retailers and the associated objective value. The solutions respectively represent $\alpha_j = 0.95$, $\alpha_j = 0.5$ and a fixed margin of $11\frac{1}{9}\%$ with respect to the $\alpha_j = 0.5$ solution. α_j is a parameter used in Constraints 2.6.

In Table 6.6 we show statistics on the probability that a retailer has sufficient capacity to satisfy their customer's demand and the objective value for Solutions 1, 2 and 2'. Confirming with Constraints 2.6, the probability that customer demand can be satisfied by the retailer is 0.95 and 0.5 for Solutions 1 and 2 respectively. Logically, the objective value of Solution 2 is lower than the objective of Solutions 1 and 2'. However, the probability that customer demand cannot be satisfied is much lower as well. Therefore, it is more interesting to compare Solution 1 with Solution 2'. It can be seen that Solution 2' is more expensive in terms of objective value and is less likely to satisfy customer demand. Therefore, we conclude that Solution 1 is superior to Solution 2'.

6.7. Aggregation of customer and retail areas

In Chapter 4 we explained that areas in the Netherlands are defined by their postal codes. We use the decision rule that $PC4$ areas that are larger than X and that have more inhabitants than Y are divided into $PC5$ areas. We want to determine the effects of this division. Obviously, reducing X or Y produces more areas at $PC5$ level. First of all, this increases computational time since more locations should be considered. Secondly, the closest assignment constraints change which could produce a different solution.

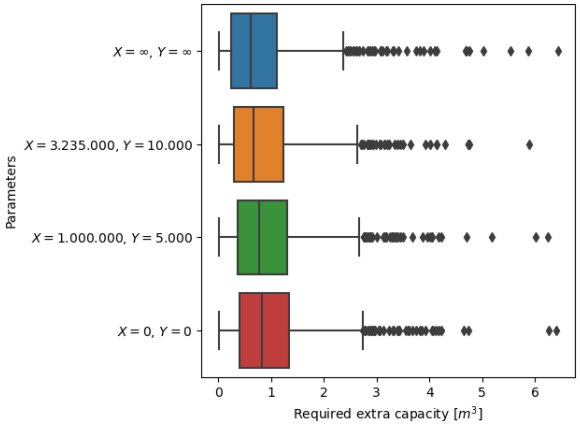


Figure 6.2: A boxplot of the required extra capacity at retailers given that the required extra capacity is larger than 0 for different levels of aggregation.

	# Open	# Close	# PC4	# PC5	Comp. time [s]
$X = \infty$, $Y = \infty$	522	757	4026	0	185
$X = 3.235.000$, $Y = 10.000$	659	693	3763	4885	557
$X = 1.000.000$, $Y = 5.000$	968	538	2693	19941	2055
$X = 0$, $Y = 0$	1242	559	0	32244	2219

Table 6.7: Results for different aggregation levels of customer and retail areas. The columns signify: the number of locations that should be opened and closed, the number of areas at $PC4$ and $PC5$ level and the computational time respectively.

In Figure 6.2 and Table 6.7 we show results on the required extra capacity at open retailers and some statistics for four different aggregation levels. First of all, in Table 6.7 it can be seen that the smaller X and Y the more $PC5$ areas and the less $PC4$ areas there are. Consequently, the computational time increases with the number of $PC5$ areas. Furthermore, we see that the number of open locations increases with the number of $PC5$ areas. First of all, this is because there is less aggregation and thus more locations in general. Secondly, this is because closest assignment constraints are much stronger on $PC5$ aggregation level than on $PC4$ aggregation level. Therefore, more new retailers are required. Moreover, the required extra capacity at the open retailers also slightly increases with the number of $PC5$ areas. This is because more retailers means that fewer customers are satisfied by each individual retailer. Consequently, the aggregated normal distribution of demand remains wider compared to a situation where many independent customers visit a single retailer.

6.8. Sensitivity of cost parameters

We investigate the model's sensitivity to the the fixed (f_j^n) and variable (f_j) costs of a new facility. In Table 6.8 we show that our model responds intuitively to these parameters. High fixed costs and low variable costs lead to fewer larger sized retailers whereas lower fixed costs and higher variable costs lead to more smaller sized retailers.

	#Open	#Close	Comp. time [s]
$f_n = 2,\ f = 2.56$	701	684	552
$f_n = 5,\ f = 1.6$	659	693	557
$f_n = 8,\ f = 0.64$	652	735	547

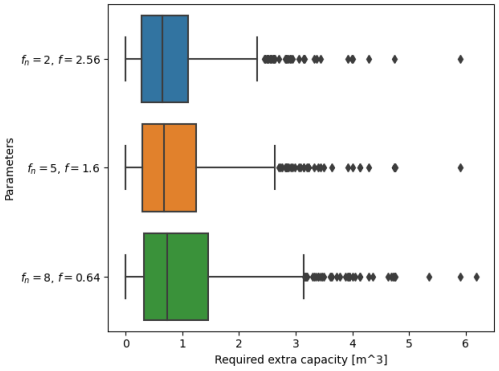


Table 6.8: A table and its boxplot with statistics on the required extra capacity at retailers given that the required extra capacity is larger than 0. Furthermore, the number of open retailers and the number of retailers that can be closed along with the computational time is indicated.

6.9. Linear vs Quadratic capacity penalty

In Chapter 2 we indicated that we choose to use a quadratic capacity penalty because it is preferred to have multiple small capacity deficits compared to a few major deficits. In this section we evaluate the effect of the quadratic capacity penalty compared to a linear capacity penalty. In Table 6.9 we show a table with the the number of additional retailers, the number of closed retailers and the computational time. Additionally, we show a boxplot of the required extra capacity at retailers given that the extra capacity is larger than zero.

	#Open	#Close	Comp. time [s]
Quadratic	659	693	558
Linear	646	761	526

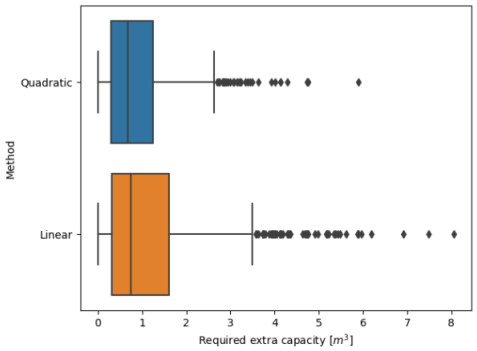


Table 6.9: A table with the number of open retailers and the number of retailers that can be closed along with the computational time. And a boxplot of the required extra capacity at retailers given that the required extra capacity is larger than 0.

Table 6.9 confirms the intended effect of the quadratic capacity penalty that more locations of smaller size are required. However, the difference in size is not very large and the mean and median are still well below the average retail size of $2.5m^3$. Therefore, it could be considered to apply the linear capacity penalty. A linear capacity penalty enables a tighter lower bound in case the B&B algorithm is preferred because it can account for some of the ignored demand. For example, suppose that during calculation of the lower bound we ignore d demand. Furthermore, suppose there are three retailers $(x,\ y,\ z)$ that can be used to account for the ignored demand d. Therefore, the d demand should be divided most economically over capacities c_x, c_y and c_z. This requires solving system 6.1 for the quadratic capacity penalty. This is difficult compared to solving the linear system 6.2. Therefore, a linear capacity penalty enables that we can include the capacity penalty on demand d.

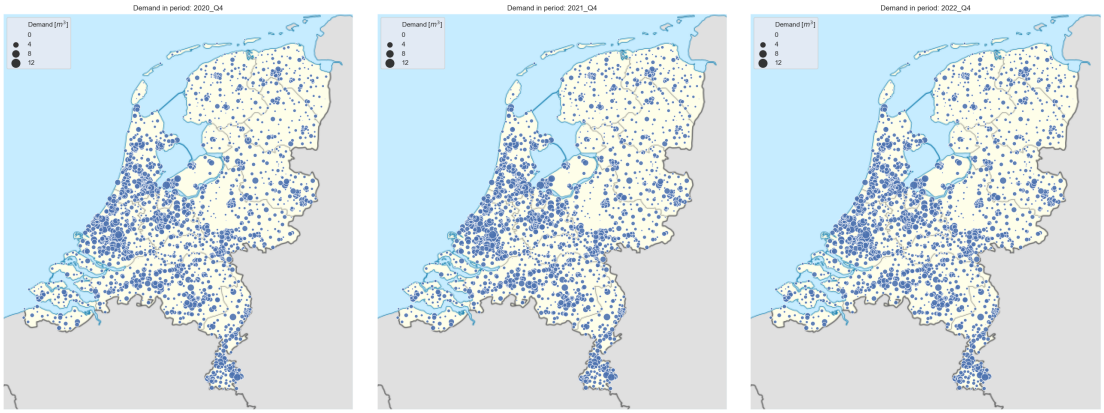
$$\min_{x,y,z} ac_x^2 + bc_y^2 + ec_z^2 \qquad c_x + c_y + c_z = d \tag{6.1}$$

$$\min_{x,y,z} ac_x + bc_y + ec_z \qquad c_x + c_y + c_z = d \tag{6.2}$$

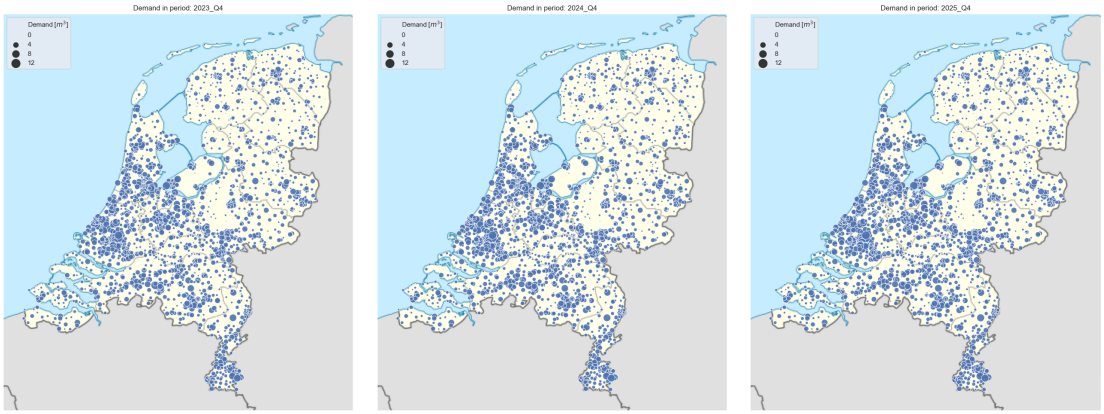
6.10. Final solution

After applying our model to the benchmark problem, we also apply our model to multiple other time periods to see the development of PostNL's parcel delivery network. We present results for every fourth quarter within the period 2020-2025. We optimize in a sequential order to keep the model tractable. This means that all retailers open in one period are also open in the next period. Because we expect the parcel volume to grow during the entire period it is unlikely that opening and closing a retailer alternates.

In Figure 6.3 we show the demand at each retailer for every fourth quarter within the period 2020-2025. From Figure 6.3 it is hard to see the development of the network. Therefore, in Figure 6.4 we show how much extra capacity is required with regards to the network in June 2020. To obtain the networks shown in Figure 6.3. Therefore, Figure 6.4 can be used to quickly identify regions of growth. Within the Netherlands it can be seen that Almere, Utrecht, the Hague and Tilburg are the major growers the upcoming five years with regards to parcel volume at retailers.

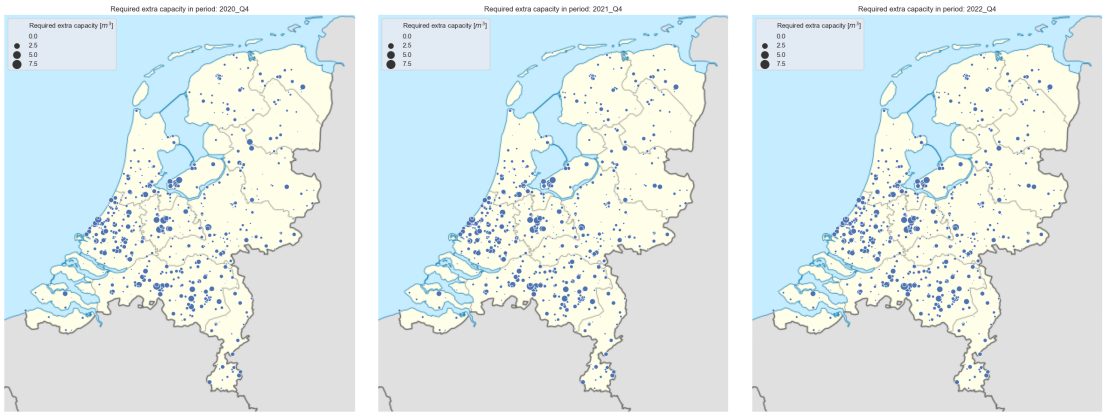


(a) Demand in 2020Q4 (b) Demand in 2021Q4 (c) Demand in 2022Q4



(d) Demand in 2023Q4 (e) Demand in 2024Q4 (f) Demand in 2025Q4

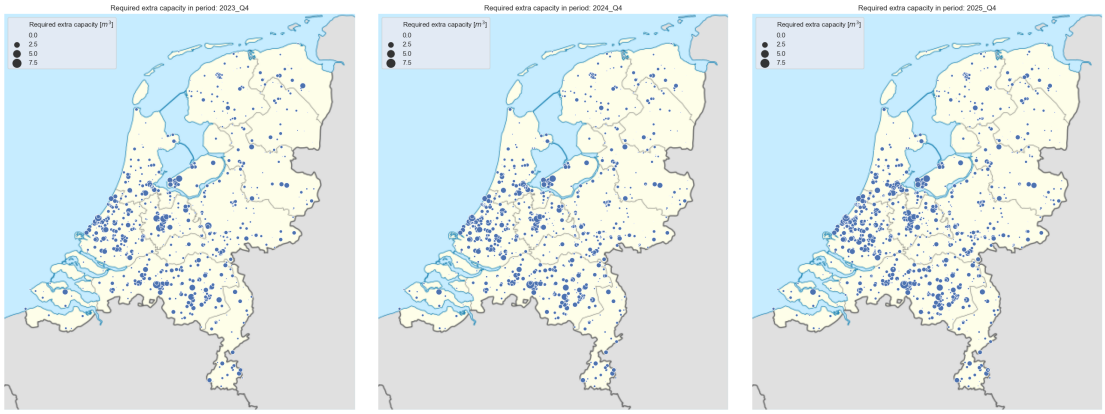
Figure 6.3: The development of PostNL's parcel delivery network. Each dot represents the required total capacity in an area during the specified period.



(a) Required extra capacity in 2020Q4

(b) Required extra capacity in 2021Q4

(c) Required extra capacity in 2022Q4



(d) Required extra capacity in 2023Q4

(e) Required extra capacity in 2024Q4

(f) Required extra capacity in 2025Q4

Figure 6.4: The development of the required extra capacity in PostNL's parcel delivery network. Each dot represents the required extra capacity in an area during the specified period.

6.10.1. Sequential versus individual optimization

In this section we compare the sequential solution with an individual solution for a specific quarter. We do this to validate whether it is acceptable to use a sequential solution instead of an integrated solution. With a sequential solution we mean that we first optimize for the first period, fix those decisions and move to the second period etc. With an individual solution we mean that we specifically optimize for one individual period. Obviously, the optimal individual solution for a specific period is the best possible solution for that period. Therefore, whenever the sequential solution is similar to the individual solution for each period we conclude that an integrated solution cannot offer much improvement. In Table 6.10 we show the absolute and relative difference between the sequential and the individual solution for all fourth quarters during the years 2021-2025. 2020Q4 is not considered since the sequential and the individual solution are equal for that period. Obviously, the absolute difference for the objective value is positive. Moreover, the positive absolute difference for the number of open locations is a direct consequence of the sequential solution where we fix the opening status of a retailer upon opening for all consecutive periods. Consequently, the absolute difference in demand is slightly positive as well since the standard error at the open retailers is slightly larger because there are more open retailers. The required extra capacity has a negative absolute difference since the sequential solution could allocate a small amount of demand in one period to a retailer without foreseeing that at a later period this could mean that large amounts of demand should be allocated to that retailer or an additional retailer should be opened. Naturally, all absolute differences increase over time since errors accumulate.

	# open retailers				Demand [m^3]				Objective value				Required extra capacity [m^3]			
Period	Seq.	Ind.	Abs.	Rel. %	Seq.	Ind.	Abs.	Rel. %	Seq.	Ind.	Abs.	Rel. %	Seq.	Ind.	Abs.	Rel. %
2021_Q4	2359	2335	24	1.0	6317	6314	3	0.0	9315	9271	44	0.5	573	578	-4	-0.8
2022_Q4	2367	2333	34	1.5	6403	6398	5	0.1	9509	9445	64	0.7	593	603	-9	-1.6
2023_Q4	2392	2342	50	2.1	6601	6594	7	0.1	9959	9795	164	1.7	652	671	-19	-2.9
2024_Q4	2425	2345	80	3.4	6879	6869	11	0.2	10645	10415	230	2.2	745	781	-36	-4.7
2025_Q4	2460	2361	99	4.2	7238	7226	12	0.2	11475	11170	305	2.7	890	929	-39	-4.2

Table 6.10: Absolute and relative difference in results between a sequential solution for multiple periods (Seq.) and an individual solution (Ind.). Difference is shown in absolute numbers (Abs.) and relative with respect to the individual solution (Rel. %). The absolute difference is calculated with $Seq. - Ind.$. The relative difference is calculated with $(Seq. - Ind.)/Ind.$

7

Conclusion and future research

In this Chapter we summarize the report and present our conclusions. Furthermore, we indicate future research directions.

7.1. Conclusions

The goal of this project is to construct a sustainable parcel delivery network for PostNL operating in the Netherlands for the upcoming five years. Restrictions that characterize a sustainable network are that each customer is free to select a retailer of his or her preference. Furthermore, each customer should have access to a retailer within a specific distance depending on their address. Additionally, a parcel from the PostNL stream should be delivered at the retailer closest to the recipient of the parcel. Moreover, each retailer should have sufficient capacity with a specific certainty to satisfy customer demand. On top of that, the objective is that the transition from the current network to the new network is smooth and that maintaining the new network is cost efficient. We formulate a mathematical model P 2.4-2.11 that translates the restrictions and objective to mathematics.

Because model P is NP-hard and consists of many variables we develop a method to decompose the general problem into multiple subproblems to reduce the computational effort. The decomposition method works without loss of optimality. Decomposition methods decompose PostNL's network into 1307 subproblems of which 264 are non-trivial. Furthermore, from the initial 6216 binary opening variables 762 could be fixed. The median number of binary opening variables in a non-trivial cluster is 14.5. Therefore, we obtain a significant problem simplification with our decomposition method.

To solve the decomposed problems individually we developed two solution techniques: an exact B&B algorithm and a TS heuristic. The exact B&B algorithm is capable of solving 1302 of the 1307 subproblems to optimality within an hour. The average optimality gap is 4.58 %. The optimality gap signifies the relative gap between the lower bound and the upper bound of the problem. Both of these bounds are calculated with a custom developed method. To obtain the lower bound it is required to solve problem LB 5.3-5.9.

Since problem LB is NP-hard and contains many binary decision variables we use extensive pre-processing to reduce the problem size. On average, for the PostNL case, problem LB consists of 115 variables, after pre-processing this can be reduced to 5 variables on average. Therefore, the pre-processing is on average able to reduce the problem size by 95 %. Moreover, 55 % of the problems LB within the PostNL case could be solved to optimality in polynomial time by the pre-processing techniques.

The TS heuristic is capable of solving 1289 of the 1307 subproblems to optimality. This results in an optimality gap with respect to the lower bound, obtained with the exact method, of 4.89 %. However, the required computational time of the TS heuristic is 557 seconds compared to 30266 seconds for the exact B&B algorithm.

With respect to PostNL's current model, both our methods open fewer but larger retail locations. This is a desired outcome according to PostNL's experts. Furthermore, our new model has some other new features that are currently unavailable in PostNL's current model.

By including the maximum capacity constraint as a stochastic constraint instead of relying on a fixed margin for additional certainty, we obtained solutions with a higher probability of having sufficient capacity to satisfy customer demand against lower total costs. Therefore, we have shown the relevance of stochasticity for our specific problem.

All in all, we conclude that we have developed an accurate decision support tool that enables PostNL to maintain a sustainable parcel delivery network for the upcoming five years in the Netherlands.

7.2. Future work

In this section we indicate several research directions that can be used to extend the work of this report.

First, multiple periods could be included simultaneously to construct an optimal parcel delivery network throughout the transition from the current network to an ideal network. Second, the model scope could be extended from the Netherlands to the Benelux to cover all countries in which PostNL operates. However, this is mainly a data and aggregation issue not related to the model or solution techniques.

Moreover, it could be worthwhile to seek for new decomposition rules that are able to further decompose some of the subproblems. Especially focusing on those subproblems with high connectivity. Furthermore, the lower bound in the B&B algorithm could be tightened to enable quicker pruning. One direction would be to investigate whether a lower bound could be developed that does not ignore a proportion of customer demand. This could be easier if a linear capacity penalty is satisfactory instead of the quadratic capacity penalty that we currently use. On top of that, it could be investigated whether it is possible to apply more directed branching rules. Currently, branches are created for all possibilities whereas an investigation into which branches should be created could pay-off computational time wise.

Furthermore, it could be interesting to see whether more advanced Tabu tenures and neighbourhoods lead to better solutions in shorter time. Currently, a relatively basic Tabu tenure to add, remove and overrule moves on the Tabu list is used concurrently with two basic neighbourhoods. Even broader, different heuristics could also be evaluated.

More generally, it is advised to optimize the overall implementation of the developed methods. We expect that a proportion of the computational effort is consumed by inefficient implementation. Moreover, we expect that the required memory can also be reduced, this would reduce computational time and required storage space.

Other directions that are less related to the model and solution methods itself but more to the underlying assumptions and data are also recognised. It is advised to investigate customer preference for specific retailers. Why do customers select a specific retailer? and more importantly, what is required to persuade them to select another retailer? This brings the modelled future network closer to reality. On top of that, it is advised to investigate what enables a location in Locatus to become a retailer? and what is required to integrate this location in the parcel delivery network? This provides PostNL more guidance into which areas are likely to contain possible new retailers and at what costs these can be acquired.

Bibliography

[1] P. de Weerd, *E-commerce blijft groeien met dubbele cijfers,* logistiek (2019).

[2] L. CáNovas, S. GarcíA, M. Labbé, and A. MaríN, *A strengthened formulation for the simple plant location problem with order,* Operations Research Letters **35**, 141 (2007).

[3] J. G. Klincewicz, *Heuristics for the p-hub location problem,* European Journal of Operational Research **53**, 25 (1991).

[4] J. Ebery, M. Krishnamoorthy, A. Ernst, and N. Boland, *The capacitated multiple allocation hub location problem: Formulations and algorithms,* European journal of operational research **120**, 614 (2000).

[5] M. Wasner and G. Zäpfel, *An integrated multi-depot hub-location vehicle routing model for network planning of parcel service,* International journal of production economics **90**, 403 (2004).

[6] O. Ben-Ayed, *Parcel distribution network design problem,* Operational Research **13**, 211 (2013).

[7] A. Bruns, A. Klose, and P. Stähly, *Restructuring of swiss parcel delivery services,* OR-Spektrum **22**, 285 (2000).

[8] H. A. Eiselt and V. Marianov, *Foundations of location analysis*, Vol. 155 (Springer Science & Business Media, 2011).

[9] R. K. Ahuja, J. B. Orlin, S. Pallottino, M. P. Scaparra, and M. G. Scutellà, *A multi-exchange heuristic for the single-source capacitated facility location problem,* Management Science **50**, 749 (2004).

[10] M. Sun, *A tabu search heuristic procedure for the capacitated facility location problem,* Journal of Heuristics **18**, 91 (2012).

[11] N. Ghaffarinasab and B. Y. Kara, *Benders decomposition algorithms for two variants of the single allocation hub location problem,* Networks and Spatial Economics **19**, 83 (2019).

[12] Z. Kartal, S. Hasgul, and A. T. Ernst, *Single allocation p-hub median location and routing problem with simultaneous pick-up and delivery,* Transportation Research Part E: Logistics and Transportation Review **108**, 141 (2017).

[13] F. Yano, T. Shohdohji, and Y. Toyoda, *Modification of hybridized particle swarm optimization algorithms applying to facility location problems,* in *Proceedings of the 9th Asia Pasific Industrial Engineering & Management Systems Conference* (2008) pp. 2278–2287.

[14] K. Jakob and P. M. Pruzan, *The simple plant location problem: survey and synthesis,* Eur J Oper Res **12**, 36 (1983).

[15] Y. Deutsch and B. Golany, *A parcel locker network as a solution to the logistics last mile problem,* International Journal of Production Research **56**, 251 (2018).

[16] Y. H. Lin, D. He, Y. Wang, and L. H. Lee, *Last-mile delivery: Optimal locker location under multinomial logit choice model,* arXiv preprint arXiv:2002.10153 (2020).

[17] Y. H. Lin, Y. Wang, and L. H. Lee, *Parcel locker location problem under threshold luce model,* arXiv preprint arXiv:2002.10810 (2020).

[18] K. Haase and S. Müller, *A comparison of linear reformulations for multinomial logit choice probabilities in facility location models,* European Journal of Operational Research **232**, 689 (2014).

[19] G. Sá, *Branch-and-bound and approximate solutions to the capacitated plant-location problem,* Operations Research **17**, 1005 (1969).

[20] U. Akinc and B. M. Khumawala, *An efficient branch and bound algorithm for the capacitated warehouse location problem,* Management Science **23**, 585 (1977).

[21] C. Canel, B. M. Khumawala, J. Law, and A. Loh, *An algorithm for the capacitated, multi-commodity multi-period facility location problem,* Computers & Operations Research **28**, 411 (2001).

[22] J. M. Leung and T. L. Magnanti, *Valid inequalities and facets of the capacitated plant location problem,* Mathematical Programming **44**, 271 (1989).

[23] K. Aardal, *Capacitated facility location: separation algorithms and computational experience,* Mathematical Programming **81**, 149 (1998).

[24] C. R. Reeves, *Modern heuristic techniques for combinatorial problems* (John Wiley & Sons, Inc., 1993).

[25] G. Cornuéjols, R. Sridharan, and J.-M. Thizy, *A comparison of heuristics and relaxations for the capacitated plant location problem,* European journal of operational research **50**, 280 (1991).

[26] A. Geoffrion and R. M. Bride, *Lagrangean relaxation applied to capacitated facility location problems,* AIIE transactions **10**, 40 (1978).

[27] R. M. Nauss, *An improved algorithm for the capacitated facility location problem,* Journal of the Operational Research Society **29**, 1195 (1978).

[28] L. A. Lorena and E. L. Senne, *Improving traditional subgradient scheme for lagrangean relaxation: an application to location problems,* International Journal of Mathematical Algorithms **1**, 133 (1999).

[29] S.-K. Lim and Y.-D. Kim, *An integrated approach to dynamic plant location and capacity planning,* Journal of the Operational Research society **50**, 1205 (1999).

[30] L. V. Snyder, M. S. Daskin, and C.-P. Teo, *The stochastic location model with risk pooling,* European Journal of Operational Research **179**, 1221 (2007).

[31] N. Christofides and J. E. Beasley, *Extensions to a lagrangean relaxation approach for the capacitated warehouse location problem,* European Journal of Operational Research **12**, 19 (1983).

[32] C.-H. Chen and C.-J. Ting, *Combining lagrangian heuristic and ant colony system to solve the single source capacitated facility location problem,* Transportation research part E: logistics and transportation review **44**, 1099 (2008).

[33] F. Barahona and F. A. Chudak, *Near-optimal solutions to large-scale facility location problems,* Discrete Optimization **2**, 35 (2005).

[34] H. Pirkul and V. Jayaraman, *A multi-commodity, multi-plant, capacitated facility location problem: formulation and efficient heuristic solution,* Computers & Operations Research **25**, 869 (1998).

[35] G. Ghiani, F. Guerriero, and R. Musmanno, *The capacitated plant location problem with multiple facilities in the same site,* Computers & Operations Research **29**, 1903 (2002).

[36] L.-Y. Wu, X.-S. Zhang, and J.-L. Zhang, *Capacitated facility location problem with general setup cost,* Computers & Operations Research **33**, 1226 (2006).

[37] J. G. Klincewicz and H. Luss, *A lagrangian relaxation heuristic for capacitated facility location with single-source constraints,* Journal of the Operational Research Society **37**, 495 (1986).

[38] A. Shulman, *An algorithm for solving dynamic capacitated plant location problems with discrete expansion sizes,* Operations research **39**, 423 (1991).

[39] C. Lin, *Stochastic single-source capacitated facility location model with service level requirements,* International Journal of Production Economics **117**, 439 (2009).

[40] H. Salemi, *A hybrid algorithm for stochastic single-source capacitated facility location problem with service level requirements,* International Journal of Industrial Engineering Computations **7**, 295 (2016).

[41] F. Barahona and R. Anbil, *The volume algorithm: producing primal solutions with a subgradient method,* Mathematical Programming **87**, 385 (2000).

[42] N. Z. Shor, *Minimization methods for non-differentiable functions*, Vol. 3 (Springer Science & Business Media, 2012).

[43] A. A. Kuehn and M. J. Hamburger, *A heuristic program for locating warehouses,* Management science **9**, 643 (1963).

[44] S. K. Jacobsen, *Heuristics for the capacitated plant location model,* European Journal of Operational Research **12**, 253 (1983).

[45] E. Feldman, F. Lehrer, and T. Ray, *Warehouse location under continuous economies of scale,* Management Science **12**, 670 (1966).

[46] D. Erlenkotter, *A dual-based procedure for uncapacitated facility location,* Operations Research **26**, 992 (1978).

[47] S. Basu, M. Sharma, and P. S. Ghosh, *Metaheuristic applications on discrete facility location problems: a survey,* Opsearch **52**, 530 (2015).

[48] J. E. Beasley, *Or-library: distributing test problems by electronic mail,* Journal of the operational research society **41**, 1069 (1990).

[49] L. Michel and P. Van Hentenryck, *A simple tabu search for warehouse location,* European Journal of Operational Research **157**, 576 (2004).

[50] M. G. Resende and R. F. Werneck, *A hybrid multistart heuristic for the uncapacitated facility location problem,* European Journal of Operational Research **174**, 54 (2006).

[51] M. Sun, *Solving the uncapacitated facility location problem using tabu search,* Computers & Operations Research **33**, 2563 (2006).

[52] D. Ghosh, *Neighborhood search heuristics for the uncapacitated facility location problem,* European Journal of Operational Research **150**, 150 (2003).

[53] Q. Wang, R. Batta, and C. M. Rump, *Algorithms for a facility location problem with stochastic customer demand and immobile servers,* Annals of operations Research **111**, 17 (2002).

[54] H. Delmaire, J. A. Díaz, E. Fernández, and M. Ortega, *Reactive grasp and tabu search based heuristics for the single source capacitated plant location problem,* INFOR: Information Systems and Operational Research **37**, 194 (1999).

[55] M. A. Arostegui Jr, S. N. Kadipasaoglu, and B. M. Khumawala, *An empirical comparison of tabu search, simulated annealing, and genetic algorithms for facilities location problems,* International Journal of Production Economics **103**, 742 (2006).

[56] I. Espejo, A. Marín, and A. M. Rodríguez-Chía, *Closest assignment constraints in discrete location problems,* European Journal of Operational Research **219**, 49 (2012).

[57] A. Diabat, T. Aouam, and L. Ozsen, *An evolutionary programming approach for solving the capacitated facility location problem with risk pooling,* International Journal of Applied Decision Sciences **2**, 389 (2009).

[58] L. V. Snyder and M. S. Daskin, *Stochastic p-robust location problems,* Iie Transactions **38**, 971 (2006).

[59] K. Fleszar and K. S. Hindi, *An effective vns for the capacitated p-median problem,* European Journal of Operational Research **191**, 612 (2008).

[60] J. A. Dıaz and E. Fernández, *A tabu search heuristic for the generalized assignment problem,* European Journal of Operational Research **132**, 22 (2001).

[61] R. Rahmaniani, M. Saidi-Mehrabad, and H. Ashouri, *Robust capacitated facility location problem optimization model and solution algorithms,* Journal of Uncertain Systems **7**, 22 (2013).

[62] I. A. Contreras and J. A. Díaz, *Scatter search for the single source capacitated facility location problem,* Annals of Operations Research **157**, 73 (2008).

[63] B. B. Keskin and H. Üster, *A scatter search-based heuristic to locate capacitated transshipment points,* Computers & Operations Research **34**, 3112 (2007).

[64] S. Wang and J. Watada, *A hybrid modified pso approach to var-based facility location problems with variable capacity in fuzzy random uncertainty,* Information Sciences **192**, 3 (2012).

[65] L. V. Snyder, *Facility location under uncertainty: a review,* IIE transactions **38**, 547 (2006).

[66] O. Baron, J. Milner, and H. Naseraldin, *Facility location: A robust optimization approach,* Production and Operations Management **20**, 772 (2011).

[67] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, *Robust optimization*, Vol. 28 (Princeton University Press, 2009).

[68] V. Ghezavati, M. Saidi-Mehrabad, and S. Sadjadi, *A robust approach to location-allocation problem under uncertainty,* Journal of Uncertain Systems **3**, 131 (2009).

[69] V. Gabrel, M. Lacroix, C. Murat, and N. Remli, *Robust location transportation problems under uncertain demands,* Discrete Applied Mathematics **164**, 100 (2014).

[70] G. Laporte, F. V. Louveaux, and L. van Hamme, *Exact solution to a location problem with stochastic demands,* Transportation Science **28**, 95 (1994).

[71] D. S. Johnson, *The np-completeness column: An ongoing guide,* Journal of algorithms **8**, 285 (1987).

[72] A. Caprara, P. Toth, and M. Fischetti, *Algorithms for the set covering problem,* Annals of Operations Research **98**, 353 (2000).

[73] E. Balas and M. C. Carrera, *A dynamic subgradient-based branch-and-bound procedure for set covering,* Operations Research **44**, 875 (1996).

[74] J. E. Beasley, *An algorithm for set covering problem,* European Journal of Operational Research **31**, 85 (1987).

[75] M. Caserta, *Tabu search-based metaheuristic algorithm for large-scale set covering problems,* in *Metaheuristics* (Springer, 2007) pp. 43–63.

[76] J. E. Beasley, *A lagrangian heuristic for set-covering problems,* Naval Research Logistics (NRL) **37**, 151 (1990).

[77] J. García, B. Crawford, R. Soto, and P. García, *A multi dynamic binary black hole algorithm applied to set covering problem,* in *International Conference on Harmony Search Algorithm* (Springer, 2017) pp. 42–51.

[78] R. Soto, B. Crawford, R. Olivares, J. Barraza, I. Figueroa, F. Johnson, F. Paredes, and E. Olguín, *Solving the non-unicost set covering problem by using cuckoo search and black hole optimization,* Natural Computing **16**, 213 (2017).

[79] J. E. Beasley and P. C. Chu, *A genetic algorithm for the set covering problem,* European journal of operational research **94**, 392 (1996).

[80] K. Al-Sultan, M. Hussain, and J. Nizami, *A genetic algorithm for the set covering problem,* Journal of the Operational Research Society **47**, 702 (1996).

[81] M. Dabibi, B. Moghaddam, and M. Kazemi, *Locating distribution/service centers based on multi objective decision making using set covering and proximity to stock market,* International Journal of Industrial Engineering Computations **7**, 635 (2016).

[82] S. Balaji and N. Revathi, *A new approach for solving set covering problem using jumping particle swarm optimization method,* Natural Computing **15**, 503 (2016).

[83] Z.-G. Ren, Z.-R. Feng, L.-J. Ke, and Z.-J. Zhang, *New ideas for applying ant colony optimization to the set covering problem,* Computers & Industrial Engineering **58**, 774 (2010).

[84] U. Aickelin, *An indirect genetic algorithm for set covering problems,* Journal of the Operational Research Society **53**, 1118 (2002).

[85] L. W. Jacobs and M. J. Brusco, *Note: A local-search heuristic for large set-covering problems,* Naval Research Logistics (NRL) **42**, 1129 (1995).

[86] M. J. Brusco, L. W. Jacobs, and G. M. Thompson, *A morphing procedure to supplement a simulated annealing heuristic for cost-andcoverage-correlated set-covering problems,* Annals of Operations Research **86**, 611 (1999).

[87] G. Lan, G. W. DePuy, and G. E. Whitehouse, *An effective and simple heuristic for the set covering problem,* European journal of operational research **176**, 1387 (2007).

[88] F. J. Vasko, Y. Lu, and K. Zyma, *What is the best greedy-like heuristic for the weighted set covering problem?* Operations Research Letters **44**, 366 (2016).

[89] R. Soto, B. Crawford, C. Galleguillos, J. Barraza, S. Lizama, A. Muñoz, J. Vilches, S. Misra, and F. Paredes, *Comparing cuckoo search, bee colony, firefly optimization, and electromagnetism-like algorithms for solving the set covering problem,* in *International Conference on Computational Science and Its Applications* (Springer, 2015) pp. 187–202.

[90] A. P. Piotrowski, J. J. Napiorkowski, and P. M. Rowinski, *How novel is the "novel" black hole optimization approach?* Information Sciences **267**, 191 (2014).

[91] X. Chen and B. Chen, *Approximation algorithms for soft-capacitated facility location in capacitated network design,* Algorithmica **53**, 263 (2009).

[92] M. Mahdian, Y. Ye, and J. Zhang, *A 2-approximation algorithm for the soft-capacitated facility location problem,* in *Approximation, Randomization, and Combinatorial Optimization.. Algorithms and Techniques* (Springer, 2003) pp. 129–140.

[93] P. Moll, *Construction and application of a memetic algorithm assigning catchment areas to retailers for consumer parcel flow*, Master's thesis, Master's thesis, URL http://hdl. handle. net/2105/38120 (2017).

[94] M. L. Fredman and R. E. Tarjan, *Fibonacci heaps and their uses in improved network optimization algorithms,* Journal of the ACM (JACM) **34**, 596 (1987).

[95] E. W. Dijkstra *et al.*, *A note on two problems in connexion with graphs,* Numerische mathematik **1**, 269 (1959).

[96] A.-H. Esfahanian, *Connectivity algorithms,* in *Topics in structural graph theory* (Cambridge University Press, 2013) pp. 268–281.

[97] A. H. Esfahanian and S. Louis Hakimi, *On computing the connectivities of graphs and digraphs,* Networks **14**, 355 (1984).

[98] J. Edmonds and R. M. Karp, *Theoretical improvements in algorithmic efficiency for network flow problems,* Journal of the ACM (JACM) **19**, 248 (1972).

[99] M. Mützell and M. Josefsson, *Max flow algorithms: Ford-fulkerson, edmond-karp, goldberg-tarjancomparison in regards to practical running time on different types of randomized flow networks.* (2015).

[100] F. S. Hillier, *Introduction to operations research* (Tata McGraw-Hill Education, 2012).

[101] A. F. Gabor and J. C. van Ommeren, *An approximation algorithm for a facility location problem with stochastic demands and inventories,* Operations research letters **34**, 257 (2006).

[102] S. Salhi, *Defining tabu list size and aspiration criterion within tabu search methods,* Computers & Operations Research **29**, 67 (2002).